

```
<%@ Page Title="About" Language="C#" MasterPageFile="~/Site.Master"  
AutoEventWireup="true" CodeBehind="About.aspx.cs" Inherits="Web_App_Master.About" %>
```

```
<asp:Content ID="BodyContent" ContentPlaceHolderID="MainContent" runat="server">  
    <h2><%: Title %>.</h2>  
    <h3>Your application description page.</h3>  
    <p>Use this area to provide additional information.</p>  
</asp:Content>
```

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
```

```
namespace Web_App_Master
```

```
{
```

```
    public partial class About : Page
```

```
    {
```

```
        protected void Page_Load(object sender, EventArgs e)
```

```
        {
```

```
        }
```

```
    }
```

```
}
```

```
<%@ Page Title="Contact" Language="C#" MasterPageFile="~/Site.Master"
AutoEventWireup="true" CodeBehind="Contact.aspx.cs" Inherits="Web_App_Master.Contact" %>
```

```
<asp:Content ID="BodyContent" ContentPlaceHolderID="MainContent" runat="server">
```

```
<h2><%= Title %>.</h2>
```

```
<h3>Your contact page.</h3>
```

```
<address>
```

```
    One Microsoft Way<br />
```

```
    Redmond, WA 98052-6399<br />
```

```
    <abbr title="Phone">P:</abbr>
```

```
    425.555.0100
```

```
</address>
```

```
<address>
```

```
    <strong>Support:</strong> <a
```

```
href="mailto:Support@example.com">Support@example.com</a><br />
```

```
    <strong>Marketing:</strong> <a
```

```
href="mailto:Marketing@example.com">Marketing@example.com</a>
```

```
</address>
```

```
</asp:Content>
```

```
using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Web;  
using System.Web.UI;  
using System.Web.UI.WebControls;
```

```
namespace Web_App_Master
```

```
{
```

```
    public partial class Contact : Page
```

```
    {
```

```
        protected void Page_Load(object sender, EventArgs e)
```

```
        {
```

```
        }
```

```
    }
```

```
}
```

```

<%@ Page Title="Starrag" Language="C#" MasterPageFile="~/Site.Master"
AutoEventWireup="true" CodeBehind="Default.aspx.cs" Inherits="Web_App_Master._Default" %>

<%@ Register Src="~/Account/Templates/av_list_template.ascx" TagPrefix="uc1"
TagName="av_list_template" %>

<asp:Content ID="BodyContent" ContentPlaceHolderID="MainContent" runat="server">

<!-- <script>
    $(function(){
        var checks = $("input:radio");
        var lv = $("#lv1");
        checks.on('change', function(){
            if ($(this).is(":checked")) {
                lv.removeClass(function (index, css) {
                    return (css.match(/(^\s)list-type-\S+/g) || []).join(' ');
                }).addClass($(this).val());
            }
        });
    });
</script> <div class="row">
<div class="col-md-12">

    <div class="awp_box rounded bg-metro-dark shadow">
    <span class="awp-bar ">
        <span class="awp-bar-btn bg-green fg-white shadow"><i onclick="javascript:alert('hello')"
title="Add To Task List" class="mif-previous av-hand-cursor"></i></span>
        <span class="awp-bar-btn bg-green fg-white shadow"><i onclick="javascript:alert('hello')"
title="Add To Task List" class="mif-next av-hand-cursor"></i></span>
    </span>
    <div class="awp_box_title bg-metro-dark">
        <span class="fg-white shadow-metro-black">Title</span>
    </div>
    <div class="awp_box_content bg-metro-light" style="text-align:left !important;">
        <asp:Button Text="Test" runat="server" ID ="tester" OnClick="tester_Click" />
    </div>
</div>
</div>

```

```

</div>
<div class="col-md-12">
  <span class="awp-save-btn bg-red fg-white shadow"><i onclick="javascript:alert('hello')"
title="Add To Task List" style="font-size:1em;" class="glyphicon glyphicon-floppy-disk av-hand-
cursor"></i></span>
  <div class="awp_box rounded bg-metro-dark shadow">
    <div class="awp_box_title bg-metro-dark">
      <span class="fg-white shadow-metro-black">Title</span>
    </div>
    <div class="awp_box_content bg-metro-light" style="text-align:left !important;">
      <div class="example" data-text="HAHAHA">
    <div class="set-border padding10 no-padding-top no-padding-bottom" style="vertical-
align:middle;">
      <label class="input-control radio small-check">
        <input type="radio" name="r1" value="default" checked="">
        <span class="check"></span>
        <span class="caption">Default</span>
      </label>
      <label class="input-control radio small-check">
        <input type="radio" name="r1" value="list-type-icons">
        <span class="check"></span>
        <span class="caption">Icons</span>
      </label>
      <label class="input-control radio small-check">
        <input type="radio" name="r1" value="list-type-tiles">
        <span class="check"></span>
        <span class="caption">Tiles</span>
      </label>
      <label class="input-control radio small-check">
        <input type="radio" name="r1" value="list-type-listing">
        <span class="check"></span>
        <span class="caption">Listing</span>
      </label>
    </div>
    <br>
    <div class="listview set-border padding10 default" data-role="listview" id="lv1" data-on-list-
click="alert('You select ' + list.find('.list-title').text())">
      <div class="list">
        
        <span class="list-title">Video</span>
      </div>
      <div class="list">

```

```
        
        <span class="list-title">Documents</span>
    </div>
    <div class="list">
        
        <span class="list-title">Downloads</span>
    </div>
    <div class="list active">
        
        <span class="list-title">Images</span>
    </div>
    <div class="list">
        
        <span class="list-title">Music</span>
    </div>
</div>
</div>
</div>
</div>
-->

</asp:Content>
```

```

using Helpers;
using Microsoft.AspNet.Identity;
using Microsoft.AspNet.Identity.EntityFramework;
using Microsoft.AspNet.Identity.Owin;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading;
using System.Web;
using System.Web.Services;
using System.Web.UI;
using System.Web.UI.WebControls;
using Web_App_Master.Account;
using Web_App_Master.Models;
using static Notification.NotificationSystem;

namespace Web_App_Master
{
    public partial class _Default : Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {

        }

        protected void testbtn_Click(object sender, EventArgs e)
        {
            var a = AssetController.GetAsset("0007");
            if(a.IsOut)
            {
                //Asset Is Out
                a.IsOut = false;
                AssetController.UpdateAsset(a);
                a = AssetController.GetAsset("0007");
            }
            else
            {
                //Asset Is Not Out
                a.IsOut = true;
                AssetController.UpdateAsset(a);
                a = AssetController.GetAsset("0007");
            }
        }
    }
}

```



```
}
```

```
protected void tester_Click(object sender, EventArgs e)
{
    try
    {
        EmailNotice notice = new EmailNotice();

        notice.Scheduled = DateTime.Now.AddDays(30);
        notice.NoticeType = NoticeType.Checkout;
        notice.NoticeAction = Global.CheckoutAction;
        var assets = new List<Asset>() { Global.Library.Assets[0], Global.Library.Assets[1],
Global.Library.Assets[2], Global.Library.Assets[3], Global.Library.Assets[4] };
        foreach (var ass in assets)
        {
            notice.Assets.Add(ass.AssetNumber);
        }
        notice.NoticeControlNumber = assets[0].OrderNumber;
        notice.Body = Global.Library.Settings.CheckOutMessage;
        notice.Subject = "Asset Return Reminder";
        var engineer = (from d in Global.Library.Settings.ServiceEngineers where d.Name ==
assets[0].ServiceEngineer select d).FirstOrDefault();
        var statics = (from d in Global.Library.Settings.StaticEmails select d).ToList();

        notice.Emails.Add(engineer);
        notice.Emails.AddRange(statics);
        notice.EmailAddress = engineer;
        Global.NoticeSystem.Add(notice);
        var a = Global.NoticeSystem.SerializeToString(Global.NoticeSystem);
    }
    catch { }
}
```

```
protected void reateRoleBtn_Click(object sender, EventArgs e)
{
    try
    {
        var roleStore = new RoleStore<IdentityRole>(new ApplicationDbContext());
        var roleManager = new RoleManager<IdentityRole>(roleStore);
        var a = roleManager.Create(new IdentityRole("superadmin"));
        var b = roleManager.Create(new IdentityRole("Admins"));
        var c = roleManager.Create(new IdentityRole("Users"));
    }
}
```

```
var d = roleManager.Create(new IdentityRole("Customers"));
```

```
Page.SiteMaster().ShowError("Roles Created");
```

```
}
```

```
catch
```

```
{
```

```
    Page.SiteMaster().ShowError("Could Not Create Role.");
```

```
}
```

```
}
```

```
}
```

```
}
```

```

using Helpers;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Optimization;
using System.Web.Routing;
using System.Web.Security;
using System.Web.SessionState;
using System.Xml;
using Web_App_Master.Account;
using Notification;
using static Notification.NotificationSystem;
using System.IO;

namespace Web_App_Master
{

    public class Global : HttpApplication
    {
        public System.Timers.Timer Cleanup;
        public static NotificationSystem NoticeSystem { get { return Library.NotificationSystem; } set {
Library.NotificationSystem = value; } }

        public static DataStore Library;
        public static UPSaccount _UPSAccount
        {
            get
            {
                return Global.Library.Settings.UpsAccount;
            }
            set
            {
                Global.Library.Settings.UpsAccount = value;
            }
        }
    }
    void Application_Start(object sender, EventArgs e)
    {
        Cleanup = new System.Timers.Timer();
        Cleanup.Interval = (60000 * 30);
        Cleanup.Elapsed += Cleanup_Elapsed;
        Cleanup.Enabled = true;
    }
}

```

```

LoadLibrary();
LoadSettings();
LoadNotificationSystem();
// Code that runs on application startup
RouteConfig.RegisterRoutes(RouteTable.Routes);
BundleConfig.RegisterBundles(BundleTable.Bundles);
}

private void Cleanup_Elapsed(object sender, System.Timers.ElapsedEventArgs e)
{
    var files = Directory.GetFiles(Server.MapPath("/Zip/"));
    foreach(var file in files)
    {
        File.Delete(file);
    }
}

void Session_Start(object sender, EventArgs e)
{
    Session["SessionUserData"] = new Data.UserData();

    var list = new List<MenuNotice>();
    MenuNotice n = new MenuNotice();
    n.Name = User.Identity.Name;
    n.Text = "Session Started";
    list.Add(n);
    Session["Notifications"] = list;
}

public static void LoadLibrary()
{
    Library = new DataStore();
    try
    {
        Pull.Library();
    }
    catch { }
}

private static void LoadSettings()
{
    Library.Settings = new Settings();
    try
    {

```

```

        Pull.Globals();
        if(Library.Settings==null)
        {
            Library.Settings = new Settings();
        }
    }
    catch { }
}
private static void LoadNotificationSystem()
{
    //Loads and Starts Notification System
    NoticeSystem = new NotificationSystem();
    try
    {
        Pull.NotificationSystem();
        if (NoticeSystem == null)
        {
            NoticeSystem = new NotificationSystem("AWP_Notification_System");
        }
    }
    catch { }
    if (Global.Library.Settings.TESTMODE)
    {
        NoticeSystem.Interval = 60000;
        NoticeSystem.OnTimerTick += NoticeSystem_OnTimerTick;
    }

    NoticeSystem.OnNoticeAdded += NoticeSystem_OnNoticeAdded;
    NoticeSystem.OnNoticeChanged += NoticeSystem_OnNoticeChanged;
    NoticeSystem.OnScheduledTime += NoticeSystem_OnNoticeScheduled;
    NoticeSystem.OnNoticeRemoved += NoticeSystem_OnNoticeRemoved;

}

private static void NoticeSystem_OnTimerTick(object sender,
NotificationSystem.TimerTickEvent e)
{

}

private static void NoticeSystem_OnNoticeRemoved(object sender,

```

```

NotificationSystem.NotificationEvent e)
{

}

private static void NoticeSystem_OnNoticeScheduled(object sender,
NotificationSystem.NotificationEvent e)
{
    try
    {
        if (!Global.Library.Settings.TESTMODE)
        {
            EmailHelper.SendNotificationSystemNotice(e.Notice);
        }
    }
    catch { }
}

private static void NoticeSystem_OnNoticeChanged(object sender,
NotificationSystem.NotificationEvent e)
{

}

private static void NoticeSystem_OnNoticeAdded(object sender,
NotificationSystem.NotificationEvent e)
{

}

public static Notice CheckoutAction(Notice n)
{

    return n;
}

public static Notice CalibrationAction(Notice n)
{

    return n;
}
}
}

```

```
<%@ Page Title="" Language="C#" MasterPageFile="~/Site.Master" AutoEventWireup="true"  
CodeBehind="items.aspx.cs" Inherits="Web_App_Master.items" %>  
<asp:Content ID="Content1" ContentPlaceHolderID="MainContent" runat="server">  
    <div></div>  
</asp:Content>
```

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
```

```
namespace Web_App_Master
```

```
{
```

```
    public partial class items : System.Web.UI.Page
```

```
    {
```

```
        protected void Page_Load(object sender, EventArgs e)
```

```
        {
```

```
            Response.Redirect("/Account/AssetView.aspx");
```

```
        }
```

```
    }
```

```
}
```



```

<%@ Page Title="" Language="C#" Async="true" MasterPageFile="~/Site.Master"
AutoEventWireup="true" CodeBehind="Map.aspx.cs" Inherits="Web_App_Master.Map" %>
<asp:Content ID="Content1" ContentPlaceHolderID="MainContent" runat="server">
<script src="Scripts/Map.js"></script>
<script type="text/javascript">

```

```

$(document).ready(function () {

```

```

function calcDivHeight() {
    var newHeight = 0;
    var myViewHeight = jQuery(".main-content").height();
    newHeight = myViewHeight - 100;
    $("#MapDiv").height(newHeight);

```

```

}

```

```

// run on page load
calcDivHeight();

```

```

// run on window resize event
$(window).resize(function () {
    calcDivHeight();
});

```

```

});

```

```

</script>

```

```

<div class="bg" style="opacity:0.2;"></div>

```

```

<div class="row">

```

```

<div class="col-md-12">

```

```

<div class="awp_box rounded bg-sg-title shadow">

```

```

<span class="awp-bar " style="padding-bottom:-10px!important;">

```

```

<span class="awp-bar-btn bg-green fg-white shadow">

```

```

<asp:button runat="server" ClientIDMode="Static" ID="hiddenModeSwitch"

```

```

OnClick="hiddenModeSwitch_Click" style="display:none;"></asp:button>

```

```

<label class="awp-switch" title="Toggle Text/Barcode Label" style="padding-
left:2px!important;">

```

```

<input id="TestModeSwitch" type="checkbox" checked="checked"

```

```

onchange="ChangeMapView()">

```

```

<span class="check"></span>

```

```

</label>

```

```
        </span>
    </span>
    <div class="awp_box_title bg-sg-title">
        <span class="fg-white shadow-metro-black"><asp:Literal runat="server"
ID="PdfTitle" Text="Asset Map" ></asp:Literal></span>
    </div>
    <div class="awp_box_content bg-sg-box">
        <div id="MapHiddenDiv" style="display:none"><asp:Literal ID="MapHiddenData"
runat="server"></asp:Literal> </div>
        <div id="MapDiv" style="min-height:100px">
            <div id="map" style="width:100%;height:100%"></div>
        </div>
    </div>
</div>
</div>
</div>

<script
src='https://maps.googleapis.com/maps/api/js?callback=myMap&key=<%=ConfigurationManager.
AppSettings["GoogleAPIKey"] %>'></script>

</asp:Content>
```

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace Web_App_Master
{
    public partial class Map : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            if (!IsPostBack)
            {
                Session["MapView"] = "default";
                Session["MapBool"] = true;
            }
        }

        protected void hiddenModeSwitch_Click(object sender, EventArgs e)
        {
            if ((bool)Session["MapBool"])
            {
                Session["MapBool"] = false;
            }
            else
            {
                Session["MapBool"] = true;
            }
        }
    }
}

```

```

using System;
using System.Collections.Generic;
using System.Security.Claims;
using System.Security.Principal;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using Microsoft.AspNet.Identity;
using Helpers;
using System.Xml;
using System.Linq;
using Web_App_Master.Account;
using System.IO;
using static Notification.NotificationSystem;

namespace Web_App_Master
{
    public partial class SiteMaster : MasterPage
    {
        public event EventHandler<UpdateRequestEvent> OnPanelUpdate;
        protected virtual void UpdateSubscribers(UpdateRequestEvent e)
        {
            try
            {
                EventHandler<UpdateRequestEvent> handler = OnPanelUpdate;
                if (handler != null)
                {
                    handler(this, e);
                }
            }
            catch (Exception ex) { UpdateSubscribers(new UpdateRequestEvent(ex)); }
        }

        public void ShowError(string message,string title="")
        {
            ErrorLabel.Text = title;
            ErrorMessage.Text = message;
            var script = @"$(document).ready(function (){ HideLoader();ToggleError();});";
            ScriptManager.RegisterStartupScript(this, this.GetType(), "testScript", script, true);
        }
    }
}

```

```

public void UpdateAllPanels()
{
    try
    {
        BindHistory();
        AssetHistoryUpdatePanel.Update();
    }
    catch { }
    try
    {
        BindCalibration();
        CalibrationUpdatePanel.Update();
    }
    catch { }
    try
    {
        BindCheckin();
        CheckInUpdatePanel.Update();
    }
    catch { }
    try {
        BindCheckout();
        CheckoutUpdatePanel.Update();
    }
    catch { }
    try
    {
        BindUserNotice();
        UserNoticeUpdatePanel.Update();
    }
    catch { }
    try
    {
        UpdateSubscribers(new UpdateRequestEvent(this));
    }
    catch { }
}

private void BindCalibration()
{
    try

```

```

{
    var asset = Session["CurrentAsset"] as Asset;

    CalibrationRepeater.DataSource = asset.CalibrationHistory.Calibrations;
    CalibrationRepeater.DataBind();
}
catch { }
}

```

```

public List<Asset> LocalAssets { get; set; }
public bool Loggedin { get; set; }
public XmlDocument LocalXmlLibrary { get; set; }
public Asset _Asset { get; set; }
private const string AntiXsrfTokenKey = "__AntiXsrfToken";
private const string AntiXsrfUserNameKey = "__AntiXsrfUserName";
private string _antiXsrfTokenValue;
public Repeater _NoticeRepeater;
public UpdatePanel _NoticeUpdate;
public void BindCheckout()
{
    var checkout = Session["CheckOut"] as List<Asset>;
    if (checkout == null) checkout = new List<Asset>();
    CheckoutRepeater.DataSource = checkout;
    CheckoutRepeater.DataBind();
}
public void BindCheckin()
{
    var checkin = Session["CheckIn"] as List<Asset>;
    if (checkin == null) checkin = new List<Asset>();
    CheckInRepeater.DataSource = checkin;
    CheckInRepeater.DataBind();
}
public void BindUserNotice()
{
    int oldCount = 0;
    try { oldCount = UserNoticerRepeater.Items.Count; }
    catch { }
    var ud = Session["SessionUserData"] as Data.UserData;
    if (ud == null) return;
    int count = ud.Log.Count;
    UserNoticerRepeater.DataSource = ud.Log;
    UserNoticerRepeater.DataBind();
}

```

```

        if (oldCount!=count)
        {
            // NotificationIcon.Shake();
        }
    }
    protected void Page_Init(object sender, EventArgs e)
    {

        try
        {
            //BindCheckInOutBoxes();
        }
        catch { }
        _Asset = new Asset();
        Loggedin = false;
        // The code below helps to protect against XSRF attacks
        var requestCookie = Request.Cookies[AntiXsrfTokenKey];
        Guid requestCookieGuidValue;
        if (requestCookie != null && Guid.TryParse(requestCookie.Value, out
requestCookieGuidValue))
        {
            // Use the Anti-XSRF token from the cookie
            _antiXsrfTokenValue = requestCookie.Value;
            Page.ViewStateUserKey = _antiXsrfTokenValue;
        }
        else
        {
            // Generate a new Anti-XSRF token and save to the cookie
            _antiXsrfTokenValue = Guid.NewGuid().ToString("N");
            Page.ViewStateUserKey = _antiXsrfTokenValue;

            var responseCookie = new HttpCookie(AntiXsrfTokenKey)
            {
                HttpOnly = true,
                Value = _antiXsrfTokenValue
            };
            if (FormsAuthentication.RequireSSL && Request.IsSecureConnection)
            {
                responseCookie.Secure = true;
            }
            Response.Cookies.Set(responseCookie);
        }
    }

```

```

        Page.PreLoad += master_Page_PreLoad;
    }

    protected void master_Page_PreLoad(object sender, EventArgs e)
    {
        if (!IsPostBack)
        {
            // Set Anti-XSRF token
            ViewState[AntiXsrfTokenKey] = Page.ViewStateUserKey;
            ViewState[AntiXsrfUserNameKey] = Context.User.Identity.Name ?? String.Empty;
        }
        else
        {
            // Validate the Anti-XSRF token
            if ((string)ViewState[AntiXsrfTokenKey] != _antiXsrfTokenValue
                || (string)ViewState[AntiXsrfUserNameKey] != (Context.User.Identity.Name ??
String.Empty))
            {
                throw new InvalidOperationException("Validation of Anti-XSRF token failed.");
            }
        }
    }

    protected List<Asset> binddummy()
    {
        return new List<Asset>() { new Asset() { AssetName="None", AssetNumber="0000" } };
    }

    public bool IsAutoChecked = false;
    protected void Page_Load(object sender, EventArgs e)
    {
        if (!IsPostBack)
        {
            AssetHistoryRepeater.DataSource = binddummy();
            AssetHistoryRepeater.DataBind();
        }
        this.PreRender += SiteMaster_PreRender;
        AssetHistoryRepeater.HeaderTemplate =
Page.LoadTemplate("/Account/Templates/av_history_header_template.ascx");
        AssetHistoryRepeater.ItemTemplate =
Page.LoadTemplate("/Account/Templates/av_history_template.ascx");
        AssetHistoryRepeater.FooterTemplate =
Page.LoadTemplate("/Account/Templates/av_history_footer_template.ascx");
    }

```



```

var ud = Session["PersistingUserData"] as Data.UserData;
if (ud != null)
{
    try
    {
        if (ud.IsAutoChecked)
        {
            var cb = flexer.FindControl("BarcodeCheckBox") as CheckBox;
            cb.Checked = true;
        }
        else
        {
            var cb = flexer.FindControl("BarcodeCheckBox") as CheckBox;
            cb.Checked = false;
        }
    }
    catch { }
}
}

```

```

private void SiteMaster_PreRender(object sender, EventArgs e)
{
    try
    {
        if (Page.User.Identity.IsAuthenticated)
        {
            NotificationsHolder.Visible = true;
        }
    }
    catch { }
    UpdateAllPanels();
}

```

```

protected void Unnamed_LoggingOut(object sender, LoginCancelEventArgs e)
{
    Context.GetOwinContext().Authentication.SignOut(DefaultAuthenticationTypes.ApplicationCookie)
;
}
[System.Web.Services.WebMethod]
public static string search()
{

```

```
    return "worked master";  
}
```

```
public System.Collections.IEnumerable NoticeRepeater_GetData()  
{  
    return null;  
}
```

```
protected void NotifyRefreshBtn_Click(object sender, EventArgs e)  
{  
}
```

```
protected void button33_Click(object sender, EventArgs e)  
{  
    BindCheckout();  
    BindCheckin();  
}
```

```
protected void UpdateAllCarts_Click(object sender, EventArgs e)  
{  
    UpdateAllPanels();  
}
```

```
protected void AssetSaveBtn_Click(object sender, EventArgs e)  
{  
    var asset = Session["CurrentAsset"] as Asset;  
    //HOLY SHIT YOU NEED TO CHECK INPUT!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!  
    if (av_AssetNumber.Value.Length<=Global.Library.Settings.AssetNumberLength)  
        if ((from a in AssetController.GetAllAssets() where a.AssetNumber ==  
av_AssetNumber.Value select a).ToList().Count == 0)  
            { //Unique Number  
                asset.AssetNumber=av_AssetNumber.Value;  
            }  
    asset.AssetName=av_AssetName.Value;
```

```
    DateTime tmp;  
    var result = DateTime.TryParse(av_DateRecieved.Value, out tmp);  
    if (result)  
        asset.DateRecieved = tmp;  
    result = DateTime.TryParse(av_DateShipped.Value, out tmp);  
    if (result)
```

```
asset.DateShipped = tmp;
```

```
asset.Description = av_Description.Value;  
asset.OrderNumber = av_ServiceOrder.Value;  
asset.ServiceEngineer = av_ServiceEngineer.Value;  
asset.ShipTo = av_ShipTo.Value;  
asset.PersonShipping = av_PersonShipping.Value;  
try  
{  
    asset.weight = Convert.ToDecimal( av_Weight.Value);  
}  
catch { }
```

```
AssetController.UpdateAsset(asset);  
}
```

```
protected void UploadAssetImg_Click(object sender, EventArgs e)  
{  
    try  
    {  
        var filename = AssetImageUploader.FileName;  
        var asset = Session["CurrentAsset"] as Asset;
```

```
        var name = Server.MapPath("/Account/Images/" + Server.HtmlEncode(filename));  
        asset.Images += Server.HtmlEncode(filename) + ",";  
        AssetImageUploader.PostedFile.SaveAs(name);
```

```
        Save.Asset(asset);
```

```
    }  
    catch { }  
}
```

```
e) protected void AssetHistoryRepeater_ItemDataBound(object sender, RepeaterItemEventArgs  
  
{  
  
}
```

```
protected void HistoryBinderBtn_Click(object sender, EventArgs e)
```

```

{
    BindHistory();
}
public void BindHistory()
{

    var asset = Session["CurrentAsset"] as Asset;
    if (asset == null) { binddummy(); return; }
    if (asset.History.History.Count==0){ binddummy(); return; }
    try
    {
        AssetHistoryRepeater.DataSource = asset.History.History;
        AssetHistoryRepeater.DataBind();
        AssetHistoryUpdatePanel.Update();
    }
    catch { }
}

protected void clearCheckIn_Click(object sender, EventArgs e)
{
    Session["CheckIn"] = new List<Asset>();
}

protected void clearCheckout_Click(object sender, EventArgs e)
{
    Session["CheckOut"] = new List<Asset>();
}

protected void UploadAssetCertificateBtn_Click(object sender, EventArgs e)
{
    if (CalPeriod.Value == "" || CalCompany.Value == "")
    {
        ShowError("Please fill in Company and Period data...");
    }

    var asset = Session["Asset"] as Asset;
    var ext = Path.GetExtension(CertUpload.FileName);
    if (ext == "")
        ext = ".pdf";
    var file = System.Guid.NewGuid().ToString();
    var dest = Server.MapPath("/Account/Certs/" + asset.AssetNumber+file+ext);
    using (FileStream fs = new FileStream(dest, FileMode.Create))

```

```

{
    CertUpload.FileContent.CopyTo(fs);
}

try
{
    CalibrationData cd = new CalibrationData();
    cd.AssetNumber = asset.AssetNumber;
    cd.CalibrationCompany= CalCompany.Value.Sanitize();
    try {
        cd.SchedulePeriod= CalPeriod.Value;
    } catch { }
    cd.ImagePath = "/Account/Certs/" + asset.AssetNumber + file + ext;

    asset.CalibrationHistory.Calibrations.Add(cd);
    Save.Asset(asset);

    //ADD NOTIFICATION
    try
    {
        EmailNotice notice = new EmailNotice();

        notice.Scheduled = DateTime.Now.AddDays(30);
        notice.NoticeType = NoticeType.Calibration;
        notice.NoticeAction = Global.CalibrationAction;
        notice.NoticeControlNumber = asset.AssetNumber;
        notice.Body = Global.Library.Settings.CalibrationMessage;
        var statics = (from d in Global.Library.Settings.StaticEmails select d).ToList();
        EmailAddress person = new EmailAddress();
        if (statics.Count!=0)
        {
            person = statics.FirstOrDefault();
        }
        notice.Emails.AddRange(statics);
        notice.EmailAddress = person;
        notice.Scheduled =
DateTime.Now.AddMonths(Convert.ToInt32(CalPeriod.Value)).AddDays(-14);
        Global.NoticeSystem.Add(notice);
        Save.NotificationSystem();
    }
    catch { }
}

```

```

        //register startup script to hide cal uploader
        BindCalibration();
        Page.ClientScript.RegisterStartupScript(this.GetType(), "CallMyFunction",
"HideCalUploader()", true);
    }
    catch
    {

    }
}

```

```

protected void SaveCalOptionsBtn_Click(object sender, EventArgs e)
{
    if(CalPeriod.Value==" " || CalCompany.Value=="")
    {
        ShowError("Please fill in Company and Period data...");
    }
    var asset = Session["Asset"] as Asset;
    asset.CalibrationCompany = CalCompany.Value.Sanitize();
    DateTime tmp = DateTime.Now;

    asset.CalibrationPeriod = CalPeriod.Value;
    Save.Asset(asset);

}

```

```

protected void CalibrationBinderBtn_Click(object sender, EventArgs e)
{
    BindCalibration();
}

```

```

protected void DeleteCalBtn_Command(object sender, CommandEventArgs e)
{
    var asset = Session["Asset"] as Asset;
    CalibrationData rem = new CalibrationData();
    foreach (var cal in asset.CalibrationHistory.Calibrations)
    {
        if (cal.Guid == e.CommandName )
        {
            rem = cal;

```

```

        }
    }
    try
    {
        asset.CalibrationHistory.Calibrations.Remove(rem);
    }
    catch { }
    Save.Asset(asset);
    BindCalibration();
}
}
public class UpdateRequestEvent : EventArgs
{
    public UpdateRequestEvent(object obj = null)
    {
        if (obj != null)
        {
            Data = obj;
        }
    }
    public object Data { get; set; }
}
}

```

```

using System;
using System.Collections.Generic;
using System.Security.Claims;
using System.Security.Principal;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using Microsoft.AspNet.Identity;
using Helpers;
using System.Xml;
using System.Linq;
using Web_App_Master.Account;
using System.IO;
using static Notification.NotificationSystem;

namespace Web_App_Master
{
    public partial class SiteMaster : MasterPage
    {
        public event EventHandler<UpdateRequestEvent> OnPanelUpdate;
        protected virtual void UpdateSubscribers(UpdateRequestEvent e)
        {
            try
            {
                EventHandler<UpdateRequestEvent> handler = OnPanelUpdate;
                if (handler != null)
                {
                    handler(this, e);
                }
            }
            catch (Exception ex) { UpdateSubscribers(new UpdateRequestEvent(ex)); }
        }

        public void ShowError(string message,string title="")
        {
            ErrorLabel.Text = title;
            ErrorMessage.Text = message;
            var script = @"$(document).ready(function (){ HideLoader();ToggleError();});";
            ScriptManager.RegisterStartupScript(this, this.GetType(), "testScript", script, true);
        }

        public void AddMenuNotice(MenuNotice notice)
    }
}

```



```

{
    var list = Session["Notifications"] as List<MenuNotice>;

    list.Add(notice);

}
public void UpdateAllPanels()
{
    try
    {
        BindHistory();
        AssetHistoryUpdatePanel.Update();
    }
    catch { }
    try
    {
        BindCalibration();
        CalibrationUpdatePanel.Update();
    }
    catch { }
    try
    {
        //if (Context.IsAdmin())
        //    BindCheckin();
    }
    catch { }
    try {
        // if (Context.IsAdmin())
        //    BindCheckout();

    }
    catch { }
    try
    {
        BindUserNotice();
        var UserNoticeUpdatePanel = RoleLoginViewer.FindControl("UserNoticeUpdatePanel")
as UpdatePanel;
        UserNoticeUpdatePanel.Update();
    }
    catch { }
    try
    {

```

```

        UpdateSubscribers(new UpdateRequestEvent(this));
    }
    catch { }

}

private void BindCalibration()
{
    try
    {
        var asset = Session["CurrentAsset"] as Asset;

        CalibrationRepeater.DataSource = asset.CalibrationHistory.Calibrations;
        CalibrationRepeater.DataBind();
    }
    catch { }
}

public List<Asset> LocalAssets { get; set; }
public bool Loggedin { get; set; }
public XmlDocument LocalXmlLibrary { get; set; }
public Asset _Asset { get; set; }
private const string AntiXsrfTokenKey = "__AntiXsrfToken";
private const string AntiXsrfUserNameKey = "__AntiXsrfUserName";
private string _antiXsrfTokenValue;
public Repeater _NoticeRepeater;
public UpdatePanel _NoticeUpdate;
public void BindCheckout()
{
    var checkout = Session["CheckOut"] as List<Asset>;
    if (checkout == null) checkout = new List<Asset>();

    //var CheckoutRepeater = RoleLoginViewer.FindControl("CheckoutRepeater",
"Admins", "CheckoutUpdatePanel") as Repeater;
    //CheckoutRepeater.DataSource = checkout;
    //CheckoutRepeater.DataBind();
    //var CheckoutUpdatePanel = RoleLoginViewer.FindControl("CheckoutUpdatePanel",
"Admins") as UpdatePanel;
    //CheckoutUpdatePanel.Update();
    //var r = CheckoutUpdatePanel.RecursiveFindControl("CheckoutRepeater") as Repeater;
    //r.DataSource = checkout;

```

```

        //r.DataBind();
        //CheckoutUpdatePanel.Update();
    }
    public void BindCheckin()
    {
        var checkin = Session["CheckIn"] as List<Asset>;
        if (checkin == null) checkin = new List<Asset>();

        var CheckinLiteral= RoleLoginViewer.FindControl("CheckinLiteral", "Admins") as
Literal;
        CheckinLiteral.Text = "test";

        // CheckInRepeater.DataSource = checkin;
        // CheckInRepeater.DataBind();
        //var CheckInUpdatePanel = RoleLoginViewer.FindControl("CheckInUpdatePanel",
"Admins") as UpdatePanel;
        //CheckInUpdatePanel.Update();
    }
    public void BindUserNotice()
    {
        var UserNoticerRepeater = RoleLoginViewer.FindControl("UserNoticerRepeater") as
Repeater;
        var loginview = RoleLoginViewer.Controls[0] as LoginView;
        UserNoticerRepeater = loginview.FindControl("UserNoticerRepeater") as Repeater;
        int oldCount = 0;
        try { oldCount = UserNoticerRepeater.Items.Count; }
        catch { }
        var ud = Session["SessionUserData"] as Data.UserData;
        if (ud == null) return;
        int count = ud.Log.Count;
        UserNoticerRepeater.DataSource = ud.Log;
        UserNoticerRepeater.DataBind();
        if (oldCount!=count)
        {
            // NotificationIcon.Shake();
        }
    }
    protected void Page_Init(object sender, EventArgs e)
    {
        try

```

```

{
    //BindCheckInOutBoxes();
}
catch { }
_Asset = new Asset();
Loggedin = false;
// The code below helps to protect against XSRF attacks
var requestCookie = Request.Cookies[AntiXsrfTokenKey];
Guid requestCookieGuidValue;
if (requestCookie != null && Guid.TryParse(requestCookie.Value, out
requestCookieGuidValue))
{
    // Use the Anti-XSRF token from the cookie
    _antiXsrfTokenValue = requestCookie.Value;
    Page.ViewStateUserKey = _antiXsrfTokenValue;
}
else
{
    // Generate a new Anti-XSRF token and save to the cookie
    _antiXsrfTokenValue = Guid.NewGuid().ToString("N");
    Page.ViewStateUserKey = _antiXsrfTokenValue;

    var responseCookie = new HttpCookie(AntiXsrfTokenKey)
    {
        HttpOnly = true,
        Value = _antiXsrfTokenValue
    };
    if (FormsAuthentication.RequireSSL && Request.IsSecureConnection)
    {
        responseCookie.Secure = true;
    }
    Response.Cookies.Set(responseCookie);
}

Page.PreLoad += master_Page_PreLoad;
}

protected void master_Page_PreLoad(object sender, EventArgs e)
{
    if (!IsPostBack)
    {
        // Set Anti-XSRF token

```

```

        ViewState[AntiXsrfTokenKey] = Page.ViewStateUserKey;
        ViewState[AntiXsrfUserNameKey] = Context.User.Identity.Name ?? String.Empty;
    }
    else
    {
        // Validate the Anti-XSRF token
        if ((string)ViewState[AntiXsrfTokenKey] != _antiXsrfTokenValue
            || (string)ViewState[AntiXsrfUserNameKey] != (Context.User.Identity.Name ??
String.Empty))
        {
            throw new InvalidOperationException("Validation of Anti-XSRF token failed.");
        }
    }
}
protected List<Asset> binddummy()
{
    return new List<Asset>() { new Asset() { AssetName="None", AssetNumber="0000" } };
}
public bool IsAutoChecked = false;
public void BindMenuItems()
{
    if (Context.IsAdmin())
    {

    }
}
protected void Page_Load(object sender, EventArgs e)
{
    if (Context.IsAdmin())
    {
        try
        {
            var t = Web_App_Master.Pull.Transactions();
            Session["Transactions"] = t;
            var script = @"$(document).ready(function (){ PollTransactions();});";
            ScriptManager.RegisterStartupScript(this, this.GetType(), "pollScript", script, true);
        }
    }
}

```

```

    }
    catch (Exception)
    {
        Session["Transactions"] = new List<PendingTransaction>();
    }
}
if (!IsPostBack)
{
    AssetHistoryRepeater.DataSource = binddummy();
    AssetHistoryRepeater.DataBind();
    BindMenuItems();

}
this.PreRender += SiteMaster_PreRender;
AssetHistoryRepeater.HeaderTemplate =
Page.LoadTemplate("/Account/Templates/av_history_header_template.ascx");
AssetHistoryRepeater.ItemTemplate =
Page.LoadTemplate("/Account/Templates/av_history_template.ascx");
AssetHistoryRepeater.FooterTemplate =
Page.LoadTemplate("/Account/Templates/av_history_footer_template.ascx");
var ud = Session["PersistingUserData"] as Data.UserData;
if (ud != null)
{
    try
    {
        if (ud.IsAutoChecked)
        {
            var cb = RoleLoginViewer.FindControl("BarcodeCheckBox") as CheckBox;

            cb.Checked = true;
        }
        else
        {
            var cb = RoleLoginViewer.FindControl("BarcodeCheckBox") as CheckBox;
            cb.Checked = false;
        }
    }
    catch { }
}
}

```

```

private void SiteMaster_PreRender(object sender, EventArgs e)
{
    try
    {
        // UpdateAllPanels();
        if (!IsPostBack)
        {
            // BindMenuItems();
        }
    }
    catch { }
}

protected void Unnamed_LoggingOut(object sender, LoginCancelEventArgs e)
{
    Context.GetOwinContext().Authentication.SignOut(DefaultAuthenticationTypes.ApplicationCookie)
;
}
[System.Web.Services.WebMethod]
public static string search()
{
    return "worked master";
}

public System.Collections.IEnumerable NoticeRepeater_GetData()
{
    return null;
}

protected void NotifyRefreshBtn_Click(object sender, EventArgs e)
{
}

protected void button33_Click(object sender, EventArgs e)
{
    BindCheckout();
    BindCheckin();
}

protected void UpdateAllCarts_Click(object sender, EventArgs e)

```

```

{
    UpdateAllPanels();
}

```

```

protected void AssetSaveBtn_Click(object sender, EventArgs e)
{
    var asset = Session["CurrentAsset"] as Asset;
    //HOLY SHIT YOU NEED TO CHECK INPUT!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
    if (av_AssetNumber.Value.Length<=Global.Library.Settings.AssetNumberLength)
        if ((from a in AssetController.GetAllAssets() where a.AssetNumber ==
av_AssetNumber.Value select a).ToList().Count == 0)
            { //Unique Number
                asset.AssetNumber=av_AssetNumber.Value;
            }
    asset.AssetName=av_AssetName.Value;

    DateTime tmp;
    var result = DateTime.TryParse(av_DateRecieved.Value, out tmp);
    if (result)
        asset.DateRecieved = tmp;
    result = DateTime.TryParse(av_DateShipped.Value, out tmp);
    if (result)
        asset.DateShipped = tmp;

    asset.Description = av_Description.Value;
    asset.OrderNumber = av_ServiceOrder.Value;
    asset.ServiceEngineer = av_ServiceEngineer.Value;
    asset.ShipTo = av_ShipTo.Value;
    asset.PersonShipping = av_PersonShipping.Value;
    try
    {
        asset.weight = Convert.ToDecimal( av_Weight.Value);
    }
    catch { }

    AssetController.UpdateAsset(asset);
}

```

```

protected void UploadAssetImg_Click(object sender, EventArgs e)
{
    try
    {

```



```
var filename = AssetImageUploader.FileName;
```

```
var asset = Session["CurrentAsset"] as Asset;
```

```
var name = Server.MapPath("/Account/Images/" + Server.HtmlEncode(filename));
```

```
asset.Images += Server.HtmlEncode(filename) + ",";
```

```
AssetImageUploader.PostedFile.SaveAs(name);
```

```
Push.Asset(asset);
```

```
}
```

```
catch { }
```

```
}
```

e)

```
protected void AssetHistoryRepeater_ItemDataBound(object sender, RepeaterItemEventArgs
```

```
{
```

```
}
```

```
protected void HistoryBinderBtn_Click(object sender, EventArgs e)
```

```
{
```

```
BindHistory();
```

```
}
```

```
public void BindHistory()
```

```
{
```

```
var asset = Session["CurrentAsset"] as Asset;
```

```
if (asset == null) { binddummy(); return; }
```

```
if (asset.History.Count==0){ binddummy(); return; }
```

```
try
```

```
{
```

```
AssetHistoryRepeater.DataSource = asset.History;
```

```
AssetHistoryRepeater.DataBind();
```

```
AssetHistoryUpdatePanel.Update();
```

```
}
```

```
catch { }
```

```
}
```

```
protected void clearCheckIn_Click(object sender, EventArgs e)
```

```
{
```

```

    Session["CheckIn"] = new List<Asset>();
}

protected void clearCheckout_Click(object sender, EventArgs e)
{
    Session["CheckOut"] = new List<Asset>();
}

protected void UploadAssetCertificateBtn_Click(object sender, EventArgs e)
{
    if (CalPeriod.Value == "" || CalCompany.Value == "")
    {
        ShowError("Please fill in Company and Period data...");
    }

    var asset = Session["Asset"] as Asset;
    var ext = Path.GetExtension(CertUpload.FileName);
    if (ext == "")
        ext = ".pdf";
    var file = System.Guid.NewGuid().ToString();
    var dest = Server.MapPath("/Account/Certs/" + asset.AssetNumber + file + ext);
    using (FileStream fs = new FileStream(dest, FileMode.Create))
    {
        CertUpload.FileContent.CopyTo(fs);
    }

    try
    {
        CalibrationData cd = new CalibrationData();
        cd.AssetNumber = asset.AssetNumber;
        cd.CalibrationCompany = CalCompany.Value.Sanitize();
        try {
            cd.SchedulePeriod = CalPeriod.Value;
        } catch { }
        cd.ImagePath = "/Account/Certs/" + asset.AssetNumber + file + ext;

        asset.CalibrationHistory.Calibrations.Add(cd);
        Push.Asset(asset);

        //ADD NOTIFICATION
        try
        {

```

```

EmailNotice notice = new EmailNotice();

notice.Scheduled = DateTime.Now.AddDays(30);
notice.NoticeType = NoticeType.Calibration;
notice.NoticeAction = Global.CalibrationAction;
notice.NoticeControlNumber = asset.AssetNumber;
notice.Body = Global.Library.Settings.CalibrationMessage;
var statics = (from d in Global.Library.Settings.StaticEmails select d).ToList();
EmailAddress person = new EmailAddress();
if (statics.Count!=0)
{
    person = statics.FirstOrDefault();
}
notice.Emails.AddRange(statics);
notice.EmailAddress = person;
notice.Scheduled =
DateTime.Now.AddMonths(Convert.ToInt32(CalPeriod.Value)).AddDays(-14);
    Global.NoticeSystem.Add(notice);
    Push.NotificationSystem();
}
catch { }

//register startup script to hide cal uploader
BindCalibration();
Page.ClientScript.RegisterStartupScript(this.GetType(), "CallMyFunction",
"HideCalUploader()", true);
}
catch
{

}
}

protected void SaveCalOptionsBtn_Click(object sender, EventArgs e)
{
    if(CalPeriod.Value==" " || CalCompany.Value=="")
    {
        ShowError("Please fill in Company and Period data...");
    }
    var asset = Session["Asset"] as Asset;
    asset.CalibrationCompany = CalCompany.Value.Sanitize();
    DateTime tmp = DateTime.Now;

```

```
asset.CalibrationPeriod = CalPeriod.Value;  
Push.Asset(asset);
```

```
}
```

```
protected void CalibrationBinderBtn_Click(object sender, EventArgs e)  
{  
    BindCalibration();  
}
```

```
protected void DeleteCalBtn_Command(object sender, CommandEventArgs e)  
{  
    var asset = Session["Asset"] as Asset;  
    CalibrationData rem = new CalibrationData();  
    foreach (var cal in asset.CalibrationHistory.Calibrations)  
    {  
        if (cal.Guid == e.CommandName )  
        {  
            rem = cal;  
        }  
    }  
    try  
    {  
        asset.CalibrationHistory.Calibrations.Remove(rem);  
    }  
    catch { }  
    Push.Asset(asset);  
    BindCalibration();  
}
```

```
protected void CheckInRepeater_ItemDataBound(object sender, RepeaterItemEventArgs e)  
{  
  
}
```

```
protected void CheckoutRepeater_ItemDataBound(object sender, RepeaterItemEventArgs e)  
{  
    var asset = e.Item.DataItem as Helpers.Asset;  
    var lit = e.Item.Controls[1] as Literal;
```

```

        lit.Text = asset.AssetNumber + " - " + asset.AssetName;
    }

    protected void tester_Click(object sender, EventArgs e)
    {
        PendingTransaction p = new PendingTransaction();
        p.Address = Global.Library.Settings.Customers[1];
        p.OrderNumber = "nopo";
        var isin = (from aa in Global.Library.Assets where aa.IsOut = false select
aa.AssetNumber).ToList();
        p.Assets = isin;
        Push.Transaction(p);
    }
}

public class UpdateRequestEvent : EventArgs
{
    public UpdateRequestEvent(object obj = null)
    {
        if (obj != null)
        {
            Data = obj;
        }
    }
    public object Data { get; set; }
}
}

```

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace Web_App_Master
{
    public partial class Site_Mobile : System.Web.UI.MasterPage
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            Response.Redirect(SwitcherAuto.AutoLinkUrl);
        }
    }
}
```

```
using Microsoft.Owin;  
using Owin;
```

```
[assembly: OwinStartupAttribute(typeof(Web_App_Master.Startup))]
```

```
namespace Web_App_Master
```

```
{
```

```
    public partial class Startup {
```

```
        public void Configuration(IAppBuilder app) {
```

```
            ConfigureAuth(app);
```

```
        }
```

```
    }
```

```
}
```

```
<%@ Page Title="" Language="C#" MasterPageFile="~/Site.Master" AutoEventWireup="true"  
CodeBehind="Sync.aspx.cs" Inherits="Web_App_Master.Sync" %>  
<asp:Content ID="Content1" ContentPlaceHolderID="MainContent" runat="server">  
</asp:Content>
```



```

using Sync;
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Text;
using System.Web;
using System.Web.Services;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace Web_App_Master
{
    public partial class Sync : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            try
            {
                if (Request.PathInfo==" /Pull" || Request.PathInfo==" /Push")
                {

                    var stream = this.Request.InputStream;
                    byte[] bytes = new byte[] { };
                    stream.Position = 0;
                    using (MemoryStream ms = new MemoryStream())
                    {
                        stream.CopyTo(ms);
                        ms.Position = 0;
                        bytes = ms.ToArray();
                    }
                    var xml = Encoding.UTF8.GetString(bytes);

                    if (Request.PathInfo == " /Pull")
                    {
                        Pull(xml);
                    }
                    if (Request.PathInfo == " /Push")
                    {
                        Push(xml);
                    }
                }
            }
        }
    }
}

```

```
}
```

```
}
```

```
catch {
```

```
}
```

```
}
```

```
public void Push(string xml)
```

```
{
```

```
    var query = Request.QueryString["t"];
```

```
    var blank = XmlHelper.BlankXml("Error");
```

```
    if (query != null)
```

```
    {
```

```
        try
```

```
        {
```

```
            switch (query)
```

```
            {
```

```
                case "asset":
```

```
                    var asset = new
```

```
Helpers.Asset().DeserializeFromXmlString<Helpers.Asset>(xml);
```

```
                    if (!Global.Library.Settings.TESTMODE)
```

```
                        {Web_App_Master.Push.Asset(asset); }
```

```
                    break;
```

```
                case "notice":
```

```
                    var notice = new
```

```
Notification.NotificationSystem.Notice().DeserializeFromXmlString<Notification.NotificationSystem.  
Notice>(xml);
```

```
                    Web_App_Master.Push.Notification(notice);
```

```
                    break;
```

```
                case "settings":
```

```
                    var settings = new
```

```
Helpers.Settings().DeserializeFromXmlString<Helpers.Settings>(xml);
```

```
                    Global.Library.Settings = settings;
```

```
                    Web_App_Master.Push.Globals();
```

```
        break;
    case "datastore":

        break;
    default:
        break;
}
```

```
        blank = XmlHelper.BlankXml("Success");
    }
    catch { }
}
var utf8 = Encoding.UTF8.GetBytes(blank);
try
{
    Response.ClearContent();

    Response.ClearHeaders();

    Response.AddHeader("Content-Disposition", "inline; filename=" + "update.xml");

    Response.ContentType = "text/xml";

    Response.BinaryWrite(utf8);

    Response.Flush(); // Sends all currently buffered output to the client.
    Response.SuppressContent = true; // Gets or sets a value indicating whether to send
HTTP content to the client.

    HttpContext.Current.ApplicationInstance.CompleteRequest(); // Causes ASP.NET to
bypass all events and filtering in the HTTP pipeline chain of execution and directly execute the
EndRequest event.

}
catch
{
}
```

```

}

public void Pull(string xml)
{try
{

    var ret = Global.Library.SerializeToXmlString(Global.Library);
    var utf8 = Encoding.UTF8.GetBytes(ret);

    Response.ClearContent();

    Response.ClearHeaders();

    Response.AddHeader("Content-Disposition", "inline; filename=" + "update.xml");

    Response.ContentType = "text/xml";

    Response.BinaryWrite(utf8);

    Response.Flush(); // Sends all currently buffered output to the client.
    Response.SuppressContent = true; // Gets or sets a value indicating whether to send
HTTP content to the client.
    HttpContext.Current.ApplicationInstance.CompleteRequest(); // Causes ASP.NET to
bypass all events and filtering in the HTTP pipeline chain of execution and directly execute the
EndRequest event.

}
catch
{
}
}
}
}

```

```
<%@ Control Language="C#" AutoEventWireup="true" CodeBehind="ViewSwitcher.ascx.cs"
Inherits="Web_App_Master.ViewSwitcher" %>
<div id="viewSwitcher">
    <%: CurrentView %> view | <a href="<%: SwitchUrl %>" data-ajax="false">Switch to <%:
AlternateView %></a>
</div>
```

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Routing;
using System.Web.UI;
using System.Web.UI.WebControls;
using Microsoft.AspNet.FriendlyUrls.Resolvers;

namespace Web_App_Master
{
    public partial class ViewSwitcher : System.Web.UI.UserControl
    {
        protected string CurrentView { get; private set; }

        protected string AlternateView { get; private set; }

        public string SwitchUrl { get; private set; }

        protected void Page_Load(object sender, EventArgs e)
        {
            // Determine current view
            var isMobile = WebFormsFriendlyUrlResolver.IsMobileView(new
HttpContextWrapper(Context));
            CurrentView = isMobile ? "Mobile" : "Desktop";

            // Determine alternate view
            AlternateView = isMobile ? "Desktop" : "Mobile";

            // Create switch URL from the route, e.g.
~/__FriendlyUrls_SwitchView/Mobile?ReturnUrl=/Page
            var switchViewRouteName = "AspNet.FriendlyUrls.SwitchView";
            var switchViewRoute = RouteTable.Routes[switchViewRouteName];
            if (switchViewRoute == null)
            {
                // Friendly URLs is not enabled or the name of the switch view route is out of sync
                this.Visible = false;
                return;
            }
            var url = GetRouteUrl(switchViewRouteName, new { view = AlternateView,
__FriendlyUrls_SwitchViews = true });
            url += "?ReturnUrl=" + HttpUtility.UrlEncode(Request.RawUrl);

```

```

        SwitchUrl = url;
    }
    public string AutoLinkUrl { get
    {
        var isMobile = WebFormsFriendlyUrlResolver.IsMobileView(new
HttpContextWrapper(Context));
        CurrentView = isMobile ? "Mobile" : "Desktop";

        // Determine alternate view
        AlternateView = isMobile ? "Desktop" : "Mobile";

        // Create switch URL from the route, e.g.
~/__FriendlyUrls_SwitchView/Mobile?ReturnUrl=/Page
        var switchViewRouteName = "AspNet.FriendlyUrls.SwitchView";
        var switchViewRoute = RouteTable.Routes[switchViewRouteName];
        if (switchViewRoute == null)
        {
            // Friendly URLs is not enabled or the name of the switch view route is out of sync
            this.Visible = false;
            return null;
        }
        var url = GetRouteUrl(switchViewRouteName, new { view = AlternateView,
__FriendlyUrls_SwitchViews = true });
        url += "?ReturnUrl=" + HttpUtility.UrlEncode(Request.RawUrl);
        return url;
    }
}
}

```

```

<div class="app-bar" >
    <asp:LoginView ID="brandSpot" runat="server" ViewStateMode="Disabled">
        <AnonymousTemplate>
            <a class="app-bar-element branding" style="text-decoration:none;" ><strong><i
class="glyphicon glyphicon-globe"></i>&nbsp;Starrag AWP</strong></a>
        </AnonymousTemplate>
        <LoggedInTemplate>
            <a class="app-bar-element branding" onclick="toggleMetroCharm('#assets')"><i
class="glyphicon glyphicon-tasks"></i><strong>&nbsp;Tasks</strong></a>
        </LoggedInTemplate>
    </asp:LoginView>

    <span class="app-bar-divider"></span>
    <ul class="nav navbar-nav small-dropdown">
        <asp:LoginView runat="server" ID="flexer" EnableViewState="false">
            <AnonymousTemplate>
                <li ><a runat="server" href="#"><i class="glyphicon glyphicon-
home"></i></a></li>

                </AnonymousTemplate>
                <LoggedInTemplate>
                    <li><a runat="server" href="#"><i class="glyphicon glyphicon-
home"></i></a></li>
                    <li>
                        <a runat="server" href="Account/PrintBarcodes.aspx"><i class="mif-
printer"></i>&nbsp;</a>
                    </li>
                    <li>
                        <a onclick="ShowLoader()" runat="server" href="Account/AssetView.aspx"><i
class="glyphicon glyphicon-dashboard"></i>&nbsp;<strong>Assets</strong></a>
                    </li>
                    <li>
                        <div class="input-control" data-role="input" style="vertical-align:top">
                            <input id="BarcodeSearchBox" type="number" style="width:130px;" />
                            <button class="button" id="BarcodeButton" type="button" onclick="
BarcodeScanned($('#BarcodeSearchBox').val());"><i id="barcodeIcon" style="font-size:1.7em;"
class="glyphicon glyphicon-barcode "></i></button>

                        </div>

                        <asp:CheckBox ClientIDMode="Static" id="BarcodeCheckBox"
runat="server" style="vertical-align:middle" /><i class="glyphicon glyphicon-check"
style="padding-left:5px"></i>

```



```

        </li>
        </LoggedInTemplate>
    </asp:LoginView>
</ul>

    <!--Right Side Logged In Menu Items-->
    <ul class="nav navbar-nav small-dropdown place-right">
        <asp:PlaceHolder ID ="NotificationsHolder" runat="server" ClientIDMode="Static"
Visible="false">
            <li>
                <a class="dropdown-toggle"><i id="NotificationIcon" runat="server" class="mif-
bell" onclick="Quiet('NotificationIcon')"></i></a>
                <ul class="d-menu place-right ShowScroll" data-role="dropdown" style=" text-
shadow:none!important;display: none; width:250px; height:auto; max-height:300px; z-index:90000;
opacity: 0.95;filter: alpha(opacity=95);">
                    <div id="ActiveNoticeContainer" style="padding:2px;"><!--Dynamic
Placeholder-->

                        <!--Repeater for notifications NON OVERFLOW SCROLLABLE
CONTAINER-->
                        <asp:UpdatePanel runat="server" ID ="UserNoticeUpdatePanel"
ClientIDMode="Static" ChildrenAsTriggers="true" UpdateMode="Conditional">
                            <ContentTemplate>
                                <asp:Repeater ID="UserNoticerRepeater" ClientIDMode="Static"
runat="server">
                                    <HeaderTemplate>
                                        <div class="listview set-border padding5 default" data-
role="listview" id="noticelv1" >
                                            </HeaderTemplate>
                                            <ItemTemplate>
                                                <div class="list">
                                                    
                                                    <span class="list-title"><a href='< %# Eval("Href")
%>'><strong>< %# Eval("TimeString")%> - < %# Eval("Entry") %></strong></a></span>
                                                        </div>
                                                    </ItemTemplate>
                                                    <FooterTemplate> </div></FooterTemplate>
                                                </asp:Repeater>
                                            </ContentTemplate>
                                            <Triggers>
                                                <asp:AsyncPostBackTrigger ControlID="UpdateAllCarts"
EventName="Click" />

```

```

        </Triggers>
    </asp:UpdatePanel>
</div>
</ul>
</li>
</asp:Placeholder>
<asp:LoginView ID="notifyView" EnableViewState="false" runat="server">
    <RoleGroups>
        <asp:RoleGroup Roles="superadmin,Admins,Users">
            <ContentTemplate>
                <li>
                    <a href="/Map.aspx" ><i class="mif-map"></i></a>
                </li>
                <li>
                    <a href="/Account/CheckIn.aspx" ><i class="glyphicon glyphicon-cloud-
upload"></i></a>
                </li>
                <li >
                    <a href="/Account/OutCart.aspx" ><i class="glyphicon glyphicon-cloud-
download"></i></a>
                </li>
                <li>
                    <a class="dropdown-toggle"><i id="UserIcon" runat="server" class="mif-user"
onclick="Quiet('UserIcon')"></i></a>
                    <ul class="d-menu place-right" data-role="dropdown" style=" text-
shadow:none!important;display: none; width:auto; height:auto; max-height:300px; z-index:90000;
opacity: 0.95;filter: alpha(opacity=95);">
                        <li>
                            <h4 class="text-light"><a runat="server"
href="~/Account/Admin/AdminDashboard" title="Admin settings">Admin</a></h4>
                        </li>
                        <li>
                            <h4 class="text-light"><a runat="server" href="~/Account/Manage"
title="Manage your account">Manage Account</a></h4>
                        </li>
                        <li>
                            <h4 class="text-light"><asp:LoginStatus runat="server"
LogoutAction="Redirect" LogoutText="Log off" LogoutPageUrl="~/
OnLoggingOut="Unnamed_LoggingOut" /></h4>
                        </li>
                    </ul>
                </li>
            </li>
        </asp:RoleGroup>
    </RoleGroups>
</asp:LoginView>

```

```

        </ContentTemplate>
    </asp:RoleGroup>
</RoleGroups>
<LoggedInTemplate>

    <li>
        <a href="/Account/CheckIn.aspx" ><i class="glyphicon glyphicon-cloud-
upload"></i></a>
    </li>
    <li >
        <a href="/Account/OutCart.aspx" ><i class="glyphicon glyphicon-cloud-
download"></i></a>
    </li>
    <!--<li data-flexorderorigin="3" data-flexorder="4" class=""><a
onclick="toggleMetroCharm('#settings-charm')" ><i class="glyphicon glyphicon-cog"></i></a></li>-
->
    <li>
        <a class="dropdown-toggle"><i id="UserIcon" runat="server" class="mif-user"
onclick="Quiet('UserIcon')"></i></a>
        <ul class="d-menu place-right" data-role="dropdown" style=" text-
shadow:none!important;display: none; width:auto; height:auto; max-height:300px; z-index:90000;
opacity: 0.95;filter: alpha(opacity=95);">
            <li>
                <h4 class="text-light"><a runat="server" href="~/Account/Manage"
title="Manage your account">Manage Account</a></h4>
            </li>
            <li>
                <h4 class="text-light"><asp:LoginStatus runat="server"
LogoutAction="Redirect" LogoutText="Log off" LogoutPageUrl="~/
OnLoggingOut="Unnamed_LoggingOut" /></h4>
            </li>
        </ul>
    </li>

</LoggedInTemplate>
<AnonymousTemplate>
    <li>
        <a class="dropdown-toggle"><i id="UserIcon" runat="server" class="mif-user"
onclick="Quiet('UserIcon')"></i></a>

```

```
<ul class="d-menu place-right" data-role="dropdown" style=" text-
shadow:none!important;display: none; width:auto; height:auto; max-height:300px; z-index:90000;
opacity: 0.95;filter: alpha(opacity=95);">
```

```
<li>
```

```
<h4 class="text-light"><a runat="server"
href="~/Account/Register">Register</a></h4>
```

```
</li>
```

```
<li>
```

```
<h4 class="text-light"><a runat="server" href="~/Account/Login">Log
in</a></h4>
```

```
</li>
```

```
</ul>
```

```
</li>
```

```
</AnonymousTemplate>
```

```
</asp:LoginView>
```

```
</ul>
```

```
<!-- End Right Items-->
```

```
<div class="app-bar-pullbutton automatic" style="display: none;"></div><div
class="clearfix" style="width: 0;"></div><nav class="app-bar-pullmenu hidden flexstyle-nav
navbar-nav" style="display: none;"><ul class="app-bar-pullmenubar hidden nav navbar-
nav"></ul></nav></div>
```

```
<%@ Control Language="C#" AutoEventWireup="true"
CodeBehind="WebUserControl1.ascx.cs" Inherits="Web_App_Master.WebUserControl1" %>
```

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace Web_App_Master
{
    public partial class WebUserControl1 : System.Web.UI.UserControl
    {
        protected void Page_Load(object sender, EventArgs e)
        {

        }
    }
}
```

```
<%@ Page Title="Phone Number" Language="C#" MasterPageFile="~/Site.Master"
AutoEventWireup="true" CodeBehind="AddPhoneNumber.aspx.cs"
Inherits="Web_App_Master.Account.AddPhoneNumber" %>
```

```
<asp:Content ID="BodyContent" ContentPlaceHolderID="MainContent" runat="server">
    <h2><%: Title %>.</h2>
```

```

    <div class="form-horizontal">
        <h4>Add a phone number</h4>
        <hr />
        <asp:ValidationSummary runat="server" CssClass="text-danger" />
        <p class="text-danger">
            <asp:Literal runat="server" ID="ErrorMessage" />
        </p>
        <div class="form-group">
            <asp:Label runat="server" AssociatedControlID="PhoneNumber" CssClass="col-md-2
control-label">Phone Number</asp:Label>
            <div class="col-md-10">
                <asp:TextBox runat="server" ID="PhoneNumber" CssClass="form-control"
TextMode="Phone" />
                <asp:RequiredFieldValidator runat="server" ControlToValidate="PhoneNumber"
                CssClass="text-danger" ErrorMessage="The PhoneNumber field is required." />
            </div>
        </div>
        <div class="form-group">
            <div class="col-md-offset-2 col-md-10">
                <asp:Button runat="server" OnClick="PhoneNumber_Click"
                Text="Submit" CssClass="btn btn-default" />
            </div>
        </div>
    </div>
</asp:Content>
```

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using Microsoft.AspNet.Identity;
using Microsoft.AspNet.Identity.Owin;
using System.Threading.Tasks;
using Web_App_Master.Models;

namespace Web_App_Master.Account
{
    public partial class AddPhoneNumber : System.Web.UI.Page
    {
        protected void PhoneNumber_Click(object sender, EventArgs e)
        {
            var manager = Context.GetOwinContext().GetUserManager<ApplicationUserManager>();
            var code = manager.GenerateChangePhoneNumberToken(User.Identity.GetUserId(),
PhoneNumber.Text);
            if (manager.SmsService != null)
            {
                var message = new IdentityMessage
                {
                    Destination = PhoneNumber.Text,
                    Body = "Your security code is " + code
                };

                manager.SmsService.Send(message);
            }

            Response.Redirect("/Account/VerifyPhoneNumber?PhoneNumber=" +
HttpUtility.UrlEncode(PhoneNumber.Text));
        }
    }
}

```

```
<%@ Page Title="" Language="C#" MasterPageFile="~/Site.Master" AutoEventWireup="true"  
CodeBehind="AssetController.aspx.cs" Inherits="Web_App_Master.Account.AssetController" %>  
<%@ MasterType VirtualPath="~/Site.Master" %>  
<asp:Content ID="Content1" ContentPlaceHolderID="MainContent" runat="server">  
</asp:Content>
```



```

using Helpers;
using System;
using System.Collections.Generic;
using System.Data;
using System.IO;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Web;
using System.Web.Services;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Xml;
using static Notification.NotificationSystem;

namespace Web_App_Master.Account
{
    public partial class AssetController : System.Web.UI.Page
    {

        protected void Page_Load(object sender, EventArgs e)
        {
            if (!Page.User.Identity.IsAuthenticated)
            {
                Response.Redirect("/Account/Login");
            }
        }
        public static bool IsNumber(string s)
        {
            return s.All(char.IsDigit);
        }
        private static string ParseScanInput(string num)
        {
            //get setting for asset length padding

            int assetcount = Global.Library.Settings.AssetNumberLength;

            var ret = num;
            if (ret.Length < assetcount)
            {
                do

```

```
    {  
        ret = "0" + ret;  
    } while (ret.Length != assetcount);  
}
```

```
    return ret;  
}  
public static string ParseBarCode(string input)  
{
```

```
    int assetcount = Global.Library.Settings.AssetNumberLength;
```

```
    var ret = input;  
    if (ret.Length < assetcount)  
    {  
        do  
        {  
            ret = "0" + ret;  
        } while (ret.Length != assetcount);  
    }
```

```
    return ret;  
}
```

[WebMethod]

```
public static bool UpdateAsset(Asset asset)  
{  
    try  
    {  
        SQL_Request req = new SQL_Request().OpenConnection();  
        req.UpdateAsset(asset);  
        if (req.Success==true)  
        {  
            return true;  
        }  
    }  
}
```

```

        else { return false; }

    }
    catch
    {
        return false;
    }

}

```

[WebMethod]

public static bool AssetOnHold(string num, string b)

```

{
    num = ParseBarCode(num);
    foreach(var asset in GetAllAssets())
    {
        if (asset.AssetNumber==num)
        {
            try
            {
                asset.OnHold = Convert.ToBoolean(b);
                SQL_Request req = new SQL_Request().OpenConnection();
                req.UpdateAsset(asset);
                if (req.Success == true)
                {
                    return true;
                }
                else { return false; }

            }
            catch
            {
                return false;
            }
        }
    }
    return false;
}

```

[WebMethod]

public static bool AssetIsDamaged(string num, string b)

```

{
    num = ParseBarCode(num);
    foreach (var asset in GetAllAssets())
    {
        if (asset.AssetNumber == num)
        {

            try
            {
                asset.IsDamaged = Convert.ToBoolean(b);
                SQL_Request req = new SQL_Request().OpenConnection();
                req.UpdateAsset(asset);
                if (req.Success == true)
                {
                    return true;
                }
                else { return false; }

            }
            catch
            {
                return false;
            }
        }
    }
    return false;
}

```

[WebMethod]

public static bool AssetIsCalibrated(string num, string b)

```

{
    num = ParseBarCode(num);
    foreach (var asset in GetAllAssets())
    {
        if (asset.AssetNumber == num)
        {

            try
            {
                asset.IsCalibrated = Convert.ToBoolean(b);
                SQL_Request req = new SQL_Request().OpenConnection();

```

```

        req.UpdateAsset(asset);
        if (req.Success == true)
        {
            return true;
        }
        else { return false; }

    }
    catch
    {
        return false;
    }
}
return false;
}

```

[WebMethod]

public static Settings GetSettings(string AppName="AWP_STARRAG_US")

```

{

    var connected = Extensions.CheckForInternetConnection();
    if (!connected)
    {
        return Global.Library.Settings;
    }
    //barcode was scanned or manually inputted

    SQL_Request req = new SQL_Request().OpenConnection();
    req.SettingsGet(AppName);
    if (req.Tag is SettingsDBData)
    {
        try
        {
            var dbdata = req.Tag as SettingsDBData;
            var settings = new Settings().DeserializeFromXmlString<Settings>(dbdata.XmlData);
            return settings;
        }
        catch { return new Settings(); }
    }
    else

```

```
        return null;
    }
}
```

[WebMethod]

```
public static bool PushSettings( Settings settings, string AppName = "AWP_STARRAG_US")
{
    string xmldata = settings.Serialize();
    var connected = Extensions.CheckForInternetConnection();
    if (!connected)
    {
        return false;
    }
    SQL_Request req = new SQL_Request().OpenConnection();
    //barcode was scanned or manually inputted
    var sql_settings = GetSettings();
    if (sql_settings==null)
    {
        var result = req.SettingsAdd(AppName,xmldata);
        return true;
    }

    try
    {
        var result = req.SettingsUpdate(AppName,xmldata);
    }
    catch
    {
        return false;
    }
    finally { req.CloseConnection(); }
    return true;
}
```

[WebMethod]

```
public static bool PushSetting(string xml, string guid)
{
    string xmldata = xml;
    var connected = Extensions.CheckForInternetConnection();
    if (!connected)
    {
        return false;
    }
}
```

```

    }
    SQL_Request req = new SQL_Request().OpenConnection();
    //barcode was scanned or manually inputted
    var sql_settings = GetSetting(guid);
    if (sql_settings == null)
    {
        var result = req.SettingsAdd(guid, xmldata);
        return true;
    }

```

```

    try
    {
        var result = req.SettingsUpdate(guid, xmldata);
    }
    catch
    {
        return false;
    }
    finally { req.CloseConnection(); }
    return true;
}

```

[WebMethod]

```

public static bool PushSetting(SettingsDBData data)
{
    var connected = Extensions.CheckForInternetConnection();
    if (!connected)
    {
        return false;
    }
    SQL_Request req = new SQL_Request().OpenConnection();
    //barcode was scanned or manually inputted
    var sql_settings = GetSetting(data.Appname);
    if (sql_settings == null)
    {
        var result = req.SettingsAdd(data.Appname, data.XmlData, data.XmlData2,
data.XmlData3, data.XmlData4, data.XmlData5);
        return true;
    }

```

```

try
{
    var result = req.SettingsUpdate(data.Appname, data.XmlData, data.XmlData2,
data.XmlData3, data.XmlData4, data.XmlData5);
}
catch
{
    return false;
}
finally { req.CloseConnection(); }
return true;
}

```

[WebMethod]

public static SettingsDBData GetSetting(string guid)

```

{

    var connected = Extensions.CheckForInternetConnection();
    if (!connected)
    {
        return null;
    }
    //barcode was scanned or manually inputted

    SQL_Request req = new SQL_Request().OpenConnection();
    req.SettingsGet(guid);
    if (req.Tag is SettingsDBData)
    {
        try
        {
            var dbdata = req.Tag as SettingsDBData;

            return dbdata;
        }
        catch { return null; }
    }
    else
        return null;
}

```

[WebMethod]


```

public static bool SetTestMode(string ischecked)
{
    try
    {
        if (Convert.ToBoolean(ischecked))
        {
            Global.Library.Settings.TESTMODE = true;
            Push.Globals();
        }
        else
        {
            Global.Library.Settings.TESTMODE = false;
            Push.Globals();
        }
        return true;
    }
    catch { return false; }
}

```

[WebMethod]

```

public static void BindCheckInOutBoxes()
{
    try
    {
        Page page = (Page)HttpContext.Current.Handler;
        var notifyView = page.Master.FindControl("notifyView") as LoginView;
        var notifyView2 = page.FindControl("notifyView") as LoginView;
        var _NoticeRepeater = notifyView.FindControl("NoticeRepeater") as Repeater;
        var _NoticeUpdate = notifyView.FindControl("NoticeUpdatePanel") as UpdatePanel;
        if (_NoticeUpdate != null)
        {
            var checkout = HttpContext.Current.Session["CheckOut"] as List<Asset>;
            var checkin = HttpContext.Current.Session["CheckIn"] as List<Asset>;
            if (checkout == null) checkout = new List<Asset>();
            if (checkin == null) checkin = new List<Asset>();
            _NoticeRepeater.DataSource = checkout.Concat(checkin);
            _NoticeRepeater.DataBind();
            _NoticeUpdate.Update();
        }
    }
    catch { }
}

```

[WebMethod]

```
public static Asset AddCheckoutItem(string num, object autocheck=null)
{
    //MAYBE USE COOKIES HERE instead of session data
    num = ParseBarcode(num);
    var checkout = HttpContext.Current.Session["CheckOut"] as List<Asset>;

    var asset = Pull.Assets().FindAssetByNumber(num);
    if (asset.IsOut == true) return null;
    if (checkout == null)
        checkout = new List<Asset>();
    foreach(var a in checkout)
    {
        //catch duplicates
        if (a.AssetId==asset.AssetId)
            { return null; }
    }
    if (asset.IsDamaged || asset.OnHold)
    {
        return null;
    }
    checkout.Add(asset);
    HttpContext.Current.Session["CheckOut"] = checkout;
    if (autocheck as string=="true")
    {
        try
        {
            var ud = HttpContext.Current.Session["PersistingUserData"] as Data.UserData;
            if (ud.IsAutoChecked == false)
            {
                ud.IsAutoChecked = true;
                ud.Log.Add(new Data.LogEntry("AutoCheck Changed:" + autocheck as string));
                SettingsDBData db = new SettingsDBData();
                db.Appname = ud.Guid;
                db.XmlData = ud.SerializeToXmlString(ud);
                Push.Setting(db);
            }
        }
        catch { }
    }
}
}
```

```

{
    try
    {
        var ud = HttpContext.Current.Session["PersistingUserData"] as Data.UserData;
        if (ud.IsAutoChecked == true)
        {
            ud.IsAutoChecked = false;
            ud.Log.Add(new Data.LogEntry("AutoCheck Changed:" + autocheck as string));
            SettingsDBData db = new SettingsDBData();
            db.Appname = ud.Guid;
            db.XmlData = ud.SerializeToXmlString(ud);
            Push.Setting(db);
        }
    }
    catch { }
}

return asset;
}

[WebMethod]
public static bool RemoveCheckoutItem(string num)
{
    //MAYBE USE COOKIES HERE instead of session data
    try
    {
        num = ParseBarCode(num);
        var checkout = HttpContext.Current.Session["CheckOut"] as List<Asset>;
        checkout.RemoveAt(Int32.Parse(num));

        HttpContext.Current.Session["CheckOut"] = checkout;
        return true;
    }
    catch (Exception)
    {
        return false;
    }
}

[WebMethod]
public static Asset AddCheckinItem(string num, object autocheck = null)
{

```

```

num = ParseBarCode(num);
var checkin = HttpContext.Current.Session["CheckIn"] as List<Asset>;
var asset = Pull.Assets().FindAssetByNumber(num);
if (asset.IsOut == false) return null;
if (checkin == null)
    checkin = new List<Asset>();
foreach (var a in checkin)
{
    //catch duplicates
    if (a.AssetNumber == asset.AssetNumber)
    { return null; }
}
checkin.Add(asset);
HttpContext.Current.Session["CheckIn"] = checkin;
if (autocheck as string == "true")
{
    try
    {
        var ud = HttpContext.Current.Session["PersistingUserData"] as Data.UserData;
        if (ud.IsAutoChecked == false)
        {
            ud.IsAutoChecked = true;
            ud.Log.Add(new Data.LogEntry("AutoCheck Changed:" + autocheck as string));
            SettingsDBData db = new SettingsDBData();
            db.Appname = ud.Guid;
            db.XmlData = ud.SerializeToXmlString(ud);
            Push.Setting(db);
        }
    }
    catch { }
}
else
{
    try
    {
        var ud = HttpContext.Current.Session["PersistingUserData"] as Data.UserData;
        if (ud.IsAutoChecked == true)
        {
            ud.IsAutoChecked = false;
            ud.Log.Add(new Data.LogEntry("AutoCheck Changed:" + autocheck as string));
            SettingsDBData db = new SettingsDBData();

```

```

        db.Appname = ud.Guid;
        db.XmlData = ud.SerializeToXmlString(ud);
        Push.Setting(db);
    }
}
catch { }
}
return asset;
}
[WebMethod]
public static bool RemoveCheckinItem(string num)
{
    //MAYBE USE COOKIES HERE instead of session data
    try
    {
        num = ParseBarCode(num);
        var checkin = HttpContext.Current.Session["CheckIn"] as List<Asset>;
        checkin.RemoveAt(Int32.Parse(num));

        HttpContext.Current.Session["CheckIn"] = checkin;
        return true;
    }
    catch (Exception)
    {
        return false;
    }
}

[WebMethod]
public static List<Asset> GetMenuItems(string num)
{
    var checkin = HttpContext.Current.Session["CheckIn"] as List<Asset>;
    if (checkin == null) checkin = new List<Asset>();

    var checkout = HttpContext.Current.Session["CheckOut"] as List<Asset>;
    if (checkout == null) checkout = new List<Asset>();

    List<Asset> list = new List<Asset>();
    list.AddRange(checkin);
    list.AddRange(checkout);

    return list;
}

```

```

}
[WebMethod]
public static bool RemoveNoticeItem(string num)
{
    //MAYBE USE COOKIES HERE instead of session data
    try
    {
        num = ParseBarCode(num);
        var notices = HttpContext.Current.Session["Notifications"] as List<MenuNotice>;
        notices.RemoveAt(Int32.Parse(num));

        HttpContext.Current.Session["Notifications"] = notices;
        return true;
    }
    catch (Exception)
    {
        return false;
    }
}

```

```

[WebMethod]
public static List<MenuNotice> GetNotifications(string num)
{
    var list = HttpContext.Current.Session["Notifications"] as List<MenuNotice>;

    return list;
}

```

```

[WebMethod]
public static bool SetAutoCheck(string autocheck)
{
    if (autocheck as string == "true")
    {
        try
        {
            var ud = HttpContext.Current.Session["PersistingUserData"] as Data.UserData;
            if (ud.IsAutoChecked == false)
            {
                ud.IsAutoChecked = true;
                ud.Log.Add(new Data.LogEntry("AutoCheck Changed:" + autocheck as string));
                SettingsDBData db = new SettingsDBData();
            }
        }
    }
}

```

```

        db.Appname = ud.Guid;
        db.XmlData = ud.SerializeToXmlString(ud);
        Push.Setting(db);
        return true;
    }
}
catch { return false; }
}
else
{
    try
    {
        var ud = HttpContext.Current.Session["PersistingUserData"] as Data.UserData;
        if (ud.IsAutoChecked == true)
        {
            ud.IsAutoChecked = false;
            ud.Log.Add(new Data.LogEntry("AutoCheck Changed:" + autocheck as string));
            SettingsDBData db = new SettingsDBData();
            db.Appname = ud.Guid;
            db.XmlData = ud.SerializeToXmlString(ud);
            Push.Setting(db);
            return true;
        }
    }
    catch { return false; }
}
return false;
}
[WebMethod]
public static Asset GetAsset(string num)
{
    num = ParseBarCode(num);
    var connected = Extensions.CheckForInternetConnection();
    if (!connected)
    {
        var a = from ass in Global.Library.Assets where ass.AssetNumber == num select ass;
        var al = a.ToList();
        var asset = al.FirstOrDefault();
        asset.Images = asset.Images.Replace(",", "<@#$>").Replace(" ", "<@#$>").Replace("<@#$>", " ").Replace("Images", "").Replace("\\", "");
        HttpContext.Current.Session["CurrentAsset"] = asset;
        HttpContext.Current.Session["Asset"] = asset;
    }
}

```

```

        return asset;
    }
    //barcode was scanned or manually inputted
    //num = ParseScanInput(num);
    SQL_Request req = new SQL_Request().OpenConnection();
    req.GetAsset(num);
    if (req.Tag is Asset)
    {
        (req.Tag as Asset).Images = (req.Tag as Asset).Images.Replace("Images",
        "").Replace("\\", "");
        try
        {
            HttpContext.Current.Session["CurrentAsset"] = req.Tag as Asset;
            HttpContext.Current.Session["Asset"] = req.Tag as Asset;
        }
        catch
        {

        }
        return req.Tag as Asset;
    }
    else
        return new Asset();
}

```

```

[WebMethod]
public static Asset GetHistory(string num, string date)
{
    try
    {
        num = ParseBarCode(num);
        var connected = Extensions.CheckForInternetConnection();
        if (!connected)
        {
            var a = from ass in Global.Library.Assets where ass.AssetNumber == num select ass;
            var al = a.ToList();
            var asset = al.FirstOrDefault();
            asset.Images = asset.Images.Replace(",", "<@#$>").Replace(" ",
            "").Replace("<@#$>", " ").Replace("Images", "").Replace("\\", "");
            HttpContext.Current.Session["CurrentAsset"] = asset;
            var ret = GetAssetByDateShipped(asset, date);
            return ret;
        }
    }
}

```



```

    }
    //barcode was scanned or manually inputted
    num = ParseScanInput(num);
    SQL_Request req = new SQL_Request().OpenConnection();
    req.GetAsset(num);
    if (req.Tag is Asset)
    {
        (req.Tag as Asset).Images = (req.Tag as Asset).Images.Replace(",,",
"<@#$>").Replace(",", "").Replace("<@#$>", ",").Replace("Images", "").Replace("\\", "");
        HttpContext.Current.Session["CurrentAsset"] = req.Tag as Asset;
        var ret = GetAssetByDateShipped((req.Tag as Asset), date);
        return ret;
    }
    else
        return new Asset();
}
catch { return new Asset(); }
}
private static Asset GetAssetByDateShipped(Asset a, string longdatestring)
{
    try
    {
        foreach (var asset in a.History.History)
        {
            if (asset.DateShipped.Ticks.ToString() == longdatestring)
            {
                return asset;
            }
        }
    }
    catch { return new Asset(); }
    return new Asset();
}
[WebMethod]
public static List<Asset> GetAllAssets()
{
    //check internet for online offline access

    var connected = Extensions.CheckForInternetConnection();
    if (!connected)
    {

```

```

        //load local
        var file =HttpContext.Current.Server.MapPath("/Account/library.xml");
        Extensions.ImportXmlLibraryFile(file);
        return Global.Library.Assets;
    }
    SQL_Request req = new SQL_Request().OpenConnection();

    //request all assets
    req.GetAssets(true);
    if (req.Tag != null)
    {
        Global.Library.Assets = req.Tag as List<Asset>;

    }
    else
    {
        var file = HttpContext.Current.Server.MapPath("/Account/library.xml");
        Extensions.ImportXmlLibraryFile(file);
        return Global.Library.Assets;
    }

    return Global.Library.Assets;
}
[WebMethod]
public static async Task< List<Asset>> GetAllAssetsAsync()
{
    //check internet for online offline access

    var connected = Extensions.CheckForInternetConnection();
    if (!connected)
    {

        //load local
        var file = HttpContext.Current.Server.MapPath("/Account/library.xml");
        Extensions.ImportXmlLibraryFile(file);
        return Global.Library.Assets;
    }
    SQL_Request req = await new SQL_Request().OpenConnectionAsync();

    //request all assets
    await req.GetAssetsAsync(true);
    if (req.Tag != null)

```

```

{
    return req.Tag as List<Asset>;
}
else
{
    var file = HttpContext.Current.Server.MapPath("/Account/library.xml");
    Extensions.ImportXmlLibraryFile(file);
    return Global.Library.Assets;
}
}

```

[WebMethod]

public static List<Asset> AssetSearch(string SearchTerm, string SearchFilter = "Asset")

```

{
    try
    {
        List<Asset> Results = new List<Asset>();
        if (SearchFilter == "Asset")
        {

            foreach (var asset in GetAllAssets())
            {

                try
                {
                    if (asset.AssetNumber != null)
                    {
                        if (asset.AssetNumber.ToUpper().Contains(SearchTerm.ToUpper()))
                        {

                            if (!Results.Contains(asset))
                                Results.Add((Asset)asset.Clone());
                        }
                    }
                }
            }
        }
        catch
        {

        }
    }
    try
    {

```

```

        if (asset.AssetName != null)
        {
            if (asset.AssetName.ToUpper().Contains(SearchTerm.ToUpper()))
            {
                if (!Results.Contains(asset))
                    Results.Add((Asset)asset.Clone());
            }
        }
    }
}
catch
{

}
try
{
    if (asset.Description != null)
    {
        if (asset.Description.ToUpper().Contains(SearchTerm.ToUpper()))
        {
            if (!Results.Contains(asset))
                Results.Add((Asset)asset.Clone());
        }
    }
}
catch
{

}
try
{
    if (asset.ShipTo != null)
    {
        if (asset.ShipTo.ToUpper().Contains(SearchTerm.ToUpper()))
        {
            if (!Results.Contains(asset))
                Results.Add((Asset)asset.Clone());
        }
    }
}
}
catch
{

```

```

    }
    try
    {
        if (asset.ServiceEngineer != null)
        {
            if (asset.ServiceEngineer.ToUpper().Contains(SearchTerm.ToUpper()))
            {
                if (!Results.Contains(asset))
                    Results.Add((Asset)asset.Clone());
            }
        }
    }
    catch
    {
    }
    try
    {
        if (asset.PersonShipping != null)
        {
            if (asset.PersonShipping.ToUpper().Contains(SearchTerm.ToUpper()))
            {
                if (!Results.Contains(asset))
                    Results.Add((Asset)asset.Clone());
            }
        }
    }
    catch
    {
    }
    try
    {
        if (asset.OrderNumber != "-1")
        {
            if
(asset.OrderNumber.ToString().ToUpper().Contains(SearchTerm.ToUpper()))
            {
                if (!Results.Contains(asset))
                    Results.Add((Asset)asset.Clone());
            }
        }
    }

```

```

    }
    catch
    {

    }
}
return Results;
}
if (SearchFilter == "History")
{

    foreach (var aaa in GetAllAssets())
    {
        try
        {
            foreach (var asset in aaa.History.History)
            {
                try
                {
                    if (asset.AssetName != null)
                    {
                        if (asset.AssetName.ToUpper().Contains(SearchTerm.ToUpper()))
                        {
                            if (!Results.Contains(asset))
                                Results.Add((Asset)asset.Clone());
                        }
                    }
                }
            }
        }
        catch
        {

        }
        try
        {
            if (asset.AssetName != null)
            {
                if (asset.AssetName.ToUpper().Contains(SearchTerm.ToUpper()))
                {
                    if (!Results.Contains(asset))
                        Results.Add((Asset)asset.Clone());
                }
            }
        }
    }
}

```

```

    }
}
catch
{

}
try
{
    if (asset.Description != null)
    {
        if (asset.Description.ToUpper().Contains(SearchTerm.ToUpper()))
        {
            if (!Results.Contains(asset))
                Results.Add((Asset)asset.Clone());
        }
    }
}
catch
{

}
try
{
    if (asset.ShipTo != null)
    {
        if (asset.ShipTo.ToUpper().Contains(SearchTerm.ToUpper()))
        {
            if (!Results.Contains(asset))
                Results.Add((Asset)asset.Clone());
        }
    }
}
catch
{

}
try
{
    if (asset.ServiceEngineer != null)
    {
        if (asset.ServiceEngineer.ToUpper().Contains(SearchTerm.ToUpper()))
        {

```

```

        if (!Results.Contains(asset))
            Results.Add((Asset)asset.Clone());
    }
}
catch
{
}
try
{
    if (asset.PersonShipping != null)
    {
        if (asset.PersonShipping.ToUpper().Contains(SearchTerm.ToUpper()))
        {
            if (!Results.Contains(asset))
                Results.Add((Asset)asset.Clone());
        }
    }
}
catch
{
}
try
{
    if (asset.OrderNumber != "-1")
    {
        if
(asset.OrderNumber.ToString().ToUpper().Contains(SearchTerm.ToUpper()))
        {
            if (!Results.Contains(asset))
                Results.Add((Asset)asset.Clone());
        }
    }
}
catch
{
}
}
}

```



```

        catch { }
    }
    return Results;
}

if (SearchFilter == "Date")
{
    return Results;
}
}
catch { }
return new List<Asset>();
}

```

[WebMethod]

```

public static bool ExportLibrary()
{
    try {

        var assetslibrary = Global.Library;
        var assets = assetslibrary.Assets;
        var length = assets.Count;
        XmlWriterSettings settings = new XmlWriterSettings();
        settings.Indent = true;
        StringBuilder builder = new StringBuilder();
        builder.Capacity = 5000;

        using (XmlWriter writer = XmlWriter.Create(builder, settings))
        {
            writer.WriteStartDocument();
            writer.WriteStartElement("AssetList");
            foreach (var s in assets)
            {

                try
                {
                    AddAssetToXml(writer, s);
                }
                catch { }

            }
            writer.WriteEndElement();

```

```

        writer.WriteEndDocument();
    }
    using (var sr = File.OpenWrite(assetslibrary.Name))
    using (StreamWriter s = new StreamWriter(sr))
    {
        builder.ToString();
        s.WriteLine(builder.ToString());
    }
    return true;
}
catch
{
    return false;
}
}

```

```

private static void AddAssetToXml(XmlWriter writer, Asset s)
{
    try
    {
        writer.WriteStartElement("Asset");
        try
        {
            writer.WriteAttributeString("Name", s.AssetName);
        }
        catch { }

        writer.WriteStartElement("AssetNumber");
        try
        {
            writer.WriteString(s.AssetName);
        }
        catch { }
        writer.WriteEndElement();

        writer.WriteStartElement("Description");

        try
        {

```

```

        if (s.Description.Contains("\n"))
        {
            s.Description = s.Description.Replace("\n", " ");
        }
        writer.WriteString(s.Description);
    }
    catch { }
    writer.WriteEndElement();

    writer.WriteStartElement("BarcodeImage");
    try
    {
        writer.WriteString(s.BarcodeImage);
    }
    catch { }
    writer.WriteEndElement();

    writer.WriteStartElement("Images");
    try
    {

        writer.WriteStartElement("Image");
        writer.WriteString(s.Images);
        writer.WriteEndElement();

    }
    catch (Exception ex) { }
    writer.WriteEndElement();

    writer.WriteStartElement("DateRecieved");
    try
    {
        writer.WriteString(s.DateRecieved.ToString());
    }
    catch { }
    writer.WriteEndElement();

    writer.WriteStartElement("DateShipped");
    try
    {
        writer.WriteString(s.DateShipped.ToString());
    }

```

```
}  
catch { }  
writer.WriteEndElement();  
  
writer.WriteStartElement("IsOut");  
try  
{  
    writer.WriteString(s.IsOut.ToString());  
}  
catch { }  
writer.WriteEndElement();  
  
writer.WriteStartElement("OrderNumber");  
try  
{  
    writer.WriteString(s.OrderNumber.ToString());  
}  
catch { }  
writer.WriteEndElement();  
  
writer.WriteStartElement("PersonShipping");  
try  
{  
    writer.WriteString(s.PersonShipping);  
}  
catch { }  
writer.WriteEndElement();  
  
writer.WriteStartElement("ServiceEngineer");  
try  
{  
    writer.WriteString(s.ServiceEngineer);  
}  
catch { }  
writer.WriteEndElement();  
  
writer.WriteStartElement("ShipTo");  
try  
{  
    writer.WriteString(s.ShipTo);  
}  
catch { }
```

```
writer.WriteEndElement();
```

```
writer.WriteStartElement("Weight");  
try  
{  
    writer.WriteString(s.weight.ToString());  
}  
catch { }  
writer.WriteEndElement();
```

```
writer.WriteStartElement("OnHold");  
try  
{  
    writer.WriteString(s.OnHold.ToString());  
}  
catch { }  
writer.WriteEndElement();
```

```
writer.WriteStartElement("IsDamaged");  
try  
{  
    writer.WriteString(s.IsDamaged.ToString());  
}  
catch { }  
writer.WriteEndElement();
```

```
writer.WriteStartElement("ShippingInformation");
```

```
try  
{  
    writer.WriteStartElement("PackingSlip");  
    writer.WriteString(s.PackingSlip);  
    writer.WriteEndElement();  
}  
catch { }  
try  
{  
    writer.WriteStartElement("UPSLabel");  
    writer.WriteString(s.UpsLabel);  
    writer.WriteEndElement();
```

```

    }
    catch { }
    try
    {
        writer.WriteStartElement("ReturnReport");
        writer.WriteString(s.ReturnReport);
        writer.WriteEndElement();
    }
    catch { }

```

```

writer.WriteEndElement();

```

```

    writer.WriteEndElement();
}
catch (Exception ex)
{

}
}

```

[WebMethod]

```

public static List<string> GetCustomerAddresses(string num)
{
    var list = new List<string>();
    int i = 0;
    foreach(var c in Global.Library.Settings.Customers)
    {
        if (i!=0)
            list.Add( c.Address + " " + c.Address2 + " " + c.Address3 + "," + c.City + "," + c.State +
", " + c.Postal + "," + c.Country);
        ++i;
    }
    return list;
}

```

[WebMethod]

```

public static List<string> GetOutAssetAddresses(string num)
{
    if ((bool)HttpContext.Current.Session["MapBool"])
    {
        var list = new List<string>();
        foreach (var asset in GetAllAssets())

```

```

    {
        if (asset.IsOut)
        {
            try
            {
                var test = Global.Library.Settings.Customers;
                var c = (from cust in Global.Library.Settings.Customers where
cust.CompanyName == asset.ShipTo select cust).FirstOrDefault();
                list.Add(c.Address + "," + c.City + "," + c.State + "," + c.Postal + "," + c.Country);
            }
            catch { }
        }
    }
    return list;
}
else
{

    return GetCustomerAddresses(num);
}

}

```

```

//TRANSACTIONS
[WebMethod]
public static List<PendingTransaction> GetAllTransactions(string AppName =
"AWP_STARRAG_US")
{
    var connected = Extensions.CheckForInternetConnection();
    if (!connected)
    {

        return null;
    }
    SQL_Request req = new SQL_Request().OpenConnection();
    //request all assets
    req.TransactionsGetAll(AppName, true);
    if (req.Tag != null)
    {
        var data = req.Tag as List<TransationDBData>;
        var list = new List<PendingTransaction>();
        foreach(var d in data)

```

```

        {
            PendingTransaction pending = new PendingTransaction();
            pending = pending.DeserializeFromXmlString<PendingTransaction>(d.XmlData);
            list.Add(pending);
        }

        return list;

    }else
    return new List<PendingTransaction>();

}
[WebMethod]
public static PendingTransaction PullTransaction(string transactionID,string AppName =
"AWP_STARRAG_US")
{
    var connected = Extensions.CheckForInternetConnection();
    if (!connected)
    {

        return null;
    }
    SQL_Request req = new SQL_Request().OpenConnection();
    //request all assets
    req.TransactionGet(transactionID, AppName, true);
    if (req.Tag != null)
    {
        if (req.Tag is List<TransationDBData>)
        {
            var data = req.Tag as List<TransationDBData>;
            var list = new List<PendingTransaction>();
            foreach (var d in data)
            {
                PendingTransaction pending = new PendingTransaction();
                pending = pending.DeserializeFromXmlString<PendingTransaction>(d.XmlData);
                list.Add(pending);
            }

            return list.First();
        }
        if (req.Tag is TransationDBData)

```



```

        {
            var data = req.Tag as TransationDBData;
            PendingTransaction pending = new PendingTransaction();
            pending = pending.DeserializeFromXmlString<PendingTransaction>(data.XmlData);
            return pending;
        }
        return new PendingTransaction();

    }
    else
        return null;

}

[WebMethod]
public static bool PushTransaction(PendingTransaction data)
{

    var connected = Extensions.CheckForInternetConnection();
    if (!connected)
    {
        return false;
    }
    SQL_Request req = new SQL_Request().OpenConnection();
    //barcode was scanned or manually inputted
    var sdata = new TransationDBData(data);

    var sql_transaction = PullTransaction(sdata.TransactionID, sdata.Appname);
    if (sql_transaction == null)
    {
        var result = req.TransactionAdd(sdata.Appname, sdata.XmlData, sdata.TransactionID);
        return true;
    }

    try
    {
        var result = req.TransactionUpdate(sdata.Appname, sdata.XmlData,
sdata.TransactionID);
    }
    catch
    {

```

```

        return false;
    }
    finally { req.CloseConnection(); }
    return true;
}
[WebMethod]
public static bool RemoveTransaction(string transactionID )
{
    string AppName = "AWP_STARRAG_US";
    var connected = Extensions.CheckForInternetConnection();
    if (!connected)
    {

        return false;
    }
    SQL_Request req = new SQL_Request().OpenConnection();
    //request all assets
    req.TransactionDelete(transactionID, AppName, true);

    if (req.Success)
    {

        return true;
    }

    else
    {
        return false;
    }

}

}

}

```

```
<%@ Page Title="" Language="C#" MasterPageFile="~/Site.Master" AutoEventWireup="true"
CodeBehind="AssetView.aspx.cs" Inherits="Web_App_Master.Account.AssetView" %>
<%@ MasterType VirtualPath="~/Site.Master" %>
```

```
<asp:Content ID="Content1" ContentPlaceHolderID="MainContent" runat="server">
```

```
<!--menu-->
```

```
<div class="app-bar shadow-bottom" style="margin-top:0px !important; left:0px!important;
position:fixed !important; z-index:25000;">
```

```
<ul class="app-bar-menu">
```

```
<li><h1></h1> </li>
```

```
<li>
```

```
<a href="#" class="dropdown-toggle">View</a>
```

```
<ul class="d-menu" data-role="dropdown">
```

```
<li><a href="#" onclick="ChangeAssetView('list'); ShowLoader()">Default</a></li>
```

```
<li><a href="#" onclick="ChangeAssetViewListType('list-type-icons');
ShowLoader()">Icon</a></li>
```

```
<li><a href="#" onclick="ChangeAssetViewListType('list-type-tiles');
ShowLoader()">Detail</a></li>
```

```
<li><a href="#" onclick="ChangeAssetViewListType('list-type-listing');
ShowLoader()">Listing</a></li>
```

```
<li><a href="#" onclick="ChangeAssetView('smtile'); ShowLoader()">List Tile</a></li>
```

```
<li><a href="#" onclick="ChangeAssetView('mdtile'); ShowLoader()">Medium
```

```
Tile</a></li>
```

```
<li><a href="#" onclick="ChangeAssetView('lgtile'); ShowLoader()">large Tile</a></li>
```

```
</ul>
```

```
</li>
```

```
<li>
```

```
<div class="input-control text" data-role="input" style="width:250px !important;">
```

```
<asp:TextBox ID="avSearchString" runat="server" ClientIDMode="Static"
```

```
CssClass="text"></asp:TextBox>
```

```
</div>
```

```
</li>
```

```
<li>
```

```
<asp:LinkButton ID="AssetSearchBtn" ClientIDMode="Static" runat="server"
```

```
OnClick="ShowLoader()" OnClick="AssetSearchBtn_Click"><span class="glyphicon
glyphicon-search"></span></asp:LinkButton>
```

```

</li>
<li>
    <a href="#" style="text-decoration:none" class="dropdown-toggle">&nbsp;<span
title="Search Filter" class="mif-settings-ethernet"></span></a>
    <ul class="d-menu" data-role="dropdown">
        <li><a id ="avAssetSearchBtn" runat="server" onclick="SetSearchType('asset')" >Asset
Search</a> </li>
        <li><a id ="avHistorySearchBtn" runat="server" onclick="SetSearchType('history')"
>History Search</a> </li>

    </ul>
</li>
<li>
    <a href="#" style="text-decoration:none" class="dropdown-toggle">&nbsp;<span title="View
Filter" class="glyphicon glyphicon-filter"></span></a>
    <ul class="d-menu" data-role="dropdown">
        <li><asp:LinkButton ID ="ViewAll" ClientIDMode="Static" runat="server"
OnClick="ShowLoader()" OnClick="ViewAll_Click">View All</asp:LinkButton> </li>
        <li><asp:LinkButton ID ="FilterIsOut" ClientIDMode="Static" runat="server"
OnClick="ShowLoader()" OnClick="FilterIsOut_Click">Checked Out Only</asp:LinkButton>
</li>
        <li><asp:LinkButton ID ="FilterIsIn" ClientIDMode="Static" runat="server"
OnClick="ShowLoader()" OnClick="FilterIsIn_Click">Checked In Only</asp:LinkButton> </li>
        <li><asp:LinkButton ID ="FilterIsDamaged" ClientIDMode="Static" runat="server"
OnClick="ShowLoader()" OnClick="FilterIsDamaged_Click">Damaged
Only</asp:LinkButton> </li>
        <li><asp:LinkButton ID ="FilterOnHold" ClientIDMode="Static" runat="server"
OnClick="ShowLoader()" OnClick="FilterOnHold_Click">On Hold Only</asp:LinkButton>
</li>
        <li><asp:LinkButton ID ="FilterCalibrated" ClientIDMode="Static" runat="server"
OnClick="ShowLoader()" OnClick="FilterCalibrated_Click">Calibrated
Only</asp:LinkButton> </li>

    </ul>
</li>

</ul>

</div>

```

```

<!--updatepanel-->
    <asp:UpdatePanel runat="server" ID="CheckoutUpdatePanel"
UpdateMode="Conditional" ChildrenAsTriggers="true">
    <ContentTemplate>
        <!--REPEATER-->
        <div style=" padding-top:50px !important">
            <asp:Repeater ID="AssetViewRepeater" runat="server" EnableViewState="false"
OnItemDataBound="AssetViewRepeater_ItemDataBound">
                <HeaderTemplate> </HeaderTemplate>
                <ItemTemplate></ItemTemplate>
                <FooterTemplate> </FooterTemplate>
            </asp:Repeater>
        </div>
    </ContentTemplate>
    <Triggers>
        <asp:AsyncPostBackTrigger ControlID="avChangeView" EventName="Click" />
        <asp:AsyncPostBackTrigger ControlID="AssetSearchBtn" EventName="Click" />
        <asp:AsyncPostBackTrigger ControlID="FilterIsOut" EventName="Click" />
        <asp:AsyncPostBackTrigger ControlID="FilterIsIn" EventName="Click" />
        <asp:AsyncPostBackTrigger ControlID="FilterIsDamaged" EventName="Click" />
        <asp:AsyncPostBackTrigger ControlID="FilterOnHold" EventName="Click" />
        <asp:AsyncPostBackTrigger ControlID="FilterCalibrated" EventName="Click" />
    </Triggers>
</asp:UpdatePanel>

    <div id="maincontentTiles" style="height:100%;width:100%;min-height:100%;min-
width:100%;background-color:#2d89ef;">
    </div>
    <asp:TextBox runat="server" ID ="avSelectedSearch" ClientIDMode="Static"
style="display:none"></asp:TextBox>
    <asp:TextBox runat="server" ID ="avSelectedView" ClientIDMode="Static"
style="display:none"></asp:TextBox>
    <asp:Button ID="avChangeView" runat="server" Text="CLICK ME" OnClick="ChangeView_Click"
style="display:none" ClientIDMode="Static"/>

</asp:Content>

```

```
<%@ Page Title="Portal" Async="true" Language="C#" MasterPageFile="~/Site.Master"
AutoEventWireup="true" CodeBehind="AssetView.aspx.cs"
Inherits="Web_App_Master.Account.AssetView" %>
<%@ MasterType VirtualPath="~/Site.Master" %>
```

```
<asp:Content ID="Portal" ContentPlaceHolderID="MainContent" runat="server">
```

```
    <link href="../../Content/browser.css" rel="stylesheet" />
<%--    <link href="../../Content/jquery-ui.css" rel="stylesheet" />
    <script src="../../Scripts/jquery-ui.js"></script>
```

```
    <link href="../../Content/jquery.contextMenu.css" rel="stylesheet" />
<script src="../../Scripts/jquery.contextMenu.js"></script>
<script src="../../Scripts/jquery.ui.position.js"></script>
<script src="../../Scripts/FileBrowserContextMenuActions.js"></script>--%>
    <script type="text/javascript">
        $(document).ready(function () {
            Sys.WebForms.PageRequestManager.getInstance().add_endRequest(InitialiseSettings);
        });
        function InitialiseSettings() {
            HideLoader();
        }
    </script>
```

```
    <!--menu-->
<nav class="navbar navbar-inverse navbar-fixed-top shadow-bottom bg-grayDark" style="
padding-right:25px; margin-top:50px !important; text-align:left !important; left:0px!important;
position:fixed !important; z-index:25000; vertical-align:middle">
    <div class="container-fluid">
        <div class="navbar-header">
            <button type="button" class="navbar-toggle" data-toggle="collapse" data-target="#AVmenu"
style="float:left !important">
                <span class="icon-bar"></span>
                <span class="icon-bar"></span>
                <span class="icon-bar"></span>
            </button>
        </div>
        <div id="AVmenu" class="collapse navbar-collapse" >
            <ul class="nav navbar-nav starrag-menu">
                <li class="dropdown">
```

```

<a class="dropdown-toggle" data-toggle="dropdown" href="#">View
<span class="caret"></span></a>
<ul class="dropdown-menu starrag-menu">
    <li><a href="#" onclick="ChangeAssetView('list'); ">Default</a></li>
    <li><a href="#" onclick=" ChangeAssetView('icon')">Icon</a></li>
    <li><a href="#" onclick="ChangeAssetViewListType('list-type-tiles'); ">Detail</a></li>
    <li><a href="#" onclick="ChangeAssetViewListType('list-type-listing'); ">Listing</a></li>
    <li><a href="#" onclick="ChangeAssetView('smtile'); ">List Tile</a></li>
    <li><a href="#" onclick="ChangeAssetView('mdtile'); ">Medium Tile</a></li>
<%--      <li><a href="#" onclick="ChangeAssetView('lgtile'); ">large Tile</a></li>--%>
    </ul>
</li>
<li style="margin-top:10px;">
    <div class=" text" data-role="input" style="width:250px !important; vertical-
align:middle!important">
        <asp:TextBox ID="avSearchString" runat="server" ClientIDMode="Static"
CssClass="form-control"></asp:TextBox>
    </div>
</li>
<li>
    <asp:LinkButton ID ="AssetSearchBtn" ClientIDMode="Static" runat="server"
OnClick="AssetSearchBtn_Click"><span class="glyphicon glyphicon-
search""></span></asp:LinkButton>
</li>
<li class="dropdown">
    <a class="dropdown-toggle" data-toggle="dropdown" href="#">Type
    <span class="caret"></span></a>
    <ul class="dropdown-menu starrag-menu">
        <li><a id ="avAssetSearchBtn" runat="server" onclick="SetSearchType('asset') " >Asset
Search</a> </li>
        <li><a id ="avHistorySearchBtn" runat="server" onclick="SetSearchType('history') "
>History Search</a> </li>
    </ul>
</li>
<li class="dropdown">
    <a class="dropdown-toggle" data-toggle="dropdown" href="#">Filter
    <span class="caret"></span></a>
    <ul class="dropdown-menu starrag-menu">
        <li><asp:LinkButton OnClientClick="ShowLoader()" ID ="ViewAll" ClientIDMode="Static"
runat="server" OnClick="ViewAll_Click">View All</asp:LinkButton> </li>
        <li><asp:LinkButton ID ="FilterIsOut" ClientIDMode="Static" runat="server"
OnClick="FilterIsOut_Click">Checked Out Only</asp:LinkButton> </li>

```

```

        <li><asp:LinkButton ID ="FilterIsIn" ClientIDMode="Static" runat="server"
OnClick="FilterIsIn_Click">Checked In Only</asp:LinkButton> </li>
        <li><asp:LinkButton ID ="FilterIsDamaged" ClientIDMode="Static" runat="server"
OnClick="FilterIsDamaged_Click">Damaged Only</asp:LinkButton> </li>
        <li><asp:LinkButton ID ="FilterOnHold" ClientIDMode="Static" runat="server"
OnClick="FilterOnHold_Click">On Hold Only</asp:LinkButton> </li>
        <li><asp:LinkButton ID ="FilterCalibrated" ClientIDMode="Static" runat="server"
OnClick="FilterCalibrated_Click">Calibrated Only</asp:LinkButton> </li>
    </ul>
</li>
<li>
    <asp:LinkButton runat="server" OnClientClick="ShowLoader()" ID="RefreshBtn"
OnClick="RefreshBtn_Click" ><span class="mif-loop2"></span> </asp:LinkButton>
</li>
</ul>
    <asp:LoginView ID="createLoginView" runat="server" >
<RoleGroups>
<asp:RoleGroup Roles="superadmin,Admins">
<ContentTemplate>
<ul class="nav navbar-nav navbar-right starrag-menu">
    <li><a href="CreateAsset.aspx"><i class="fas fa-plus"></i><strong>Create</strong></a></li>
</ul>

</ContentTemplate>
</asp:RoleGroup>
</RoleGroups>
</asp:LoginView>
</div>
</div>
</nav>

<div class="bg-shaded"></div>
<!--updatepanel-->
    <asp:UpdatePanel runat="server" ID="CheckoutUpdatePanel"
UpdateMode="Conditional" ChildrenAsTriggers="true">
    <ContentTemplate>
        <!--REPEATER-->
        <div style=" padding-top:50px !important; background-color:black !important">
            <asp:Repeater ID="AssetViewRepeater" runat="server" EnableViewState="false"
OnItemDataBound="AssetViewRepeater_ItemDataBound">
                <HeaderTemplate> </HeaderTemplate>
                <ItemTemplate></ItemTemplate>

```



```
<FooterTemplate> </FooterTemplate>
</asp:Repeater>
</div>
</ContentTemplate>
<Triggers>
  <asp:AsyncPostBackTrigger ControlID="avChangeView" EventName="Click" />
  <asp:AsyncPostBackTrigger ControlID="AssetSearchBtn" EventName="Click" />
  <asp:AsyncPostBackTrigger ControlID="FilterIsOut" EventName="Click" />
  <asp:AsyncPostBackTrigger ControlID="FilterIsIn" EventName="Click" />
  <asp:AsyncPostBackTrigger ControlID="FilterIsDamaged" EventName="Click" />
  <asp:AsyncPostBackTrigger ControlID="FilterOnHold" EventName="Click" />
  <asp:AsyncPostBackTrigger ControlID="FilterCalibrated" EventName="Click" />
  <asp:AsyncPostBackTrigger ControlID="RefreshBtn" EventName="Click" />
</Triggers>
</asp:UpdatePanel>
```

```
<div id="maincontentTiles" style="height:100%;width:100%;min-height:100%;min-
width:100%;background-color:#2d89ef;">
```

```
</div>
<asp:TextBox runat="server" ID ="avSelectedSearch" ClientIDMode="Static"
style="display:none"></asp:TextBox>
<asp:TextBox runat="server" ID ="avSelectedView" ClientIDMode="Static"
style="display:none"></asp:TextBox>
<asp:Button OnClientClick="ShowLoader()" ID="avChangeView" runat="server" Text="CLICK ME"
OnClick="ChangeView_Click" style="display:none" ClientIDMode="Static"/>
<asp:Button OnClientClick="ShowLoader()" ID="AjaxRefresherBtn" runat="server" Text="CLICK
ME" OnClick="Refresher_Click" style="display:none" ClientIDMode="Static"/>
```

```
</asp:Content>
```

```

using Helpers;
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Web;
using System.Web.Services;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Xml;

namespace Web_App_Master.Account
{
    public partial class AssetView : System.Web.UI.Page
    {
        public string IsSelected(object num)
        {
            if ((bool)num == true)
                return "element-selected";
            else
                return "";
        }

        protected void Page_Load(object sender, EventArgs e)
        {
            if (!Page.User.Identity.IsAuthenticated)
            {
                Response.Redirect("/Account/Login");
            }

            var currentview = Session["CurrentAvView"] as string;
            UpdateView(currentview);

            // Placeholder lbl_UserName = this.Master.FindControl("RibbonBar") as Placeholder;
            //lbl_UserName.Visible = true;
        }
    }
}

```

```

private List<Asset> AssetSearch(string SearchTerm, string SearchFilter="asset")
{
    try
    {
        List<Asset> Results = new List<Asset>();
        if (SearchFilter.ToLower() == "asset")
        {

            foreach (var asset in AssetController.GetAllAssets())
            {

                try
                {
                    if (asset.AssetName != null)
                    {
                        if (asset.AssetName.ToUpper().Contains(SearchTerm.ToUpper()))
                        {

                            if (!Results.Contains(asset))
                                Results.Add((Asset)asset.Clone());
                        }
                    }
                }
                catch
                {

                }
            }
            try
            {
                if (asset.AssetName != null)
                {
                    if (asset.AssetName.ToUpper().Contains(SearchTerm.ToUpper()))
                    {
                        if (!Results.Contains(asset))
                            Results.Add((Asset)asset.Clone());
                    }
                }
            }
            catch
            {

            }
        }
    }
}

```

```

try
{
    if (asset.Description != null)
    {
        if (asset.Description.ToUpper().Contains(SearchTerm.ToUpper()))
        {
            if (!Results.Contains(asset))
                Results.Add((Asset)asset.Clone());
        }
    }
}
catch
{

}

try
{
    if (asset.ShipTo != null)
    {
        if (asset.ShipTo.ToUpper().Contains(SearchTerm.ToUpper()))
        {
            if (!Results.Contains(asset))
                Results.Add((Asset)asset.Clone());
        }
    }
}
catch
{

}

try
{
    if (asset.ServiceEngineer != null)
    {
        if (asset.ServiceEngineer.ToUpper().Contains(SearchTerm.ToUpper()))
        {
            if (!Results.Contains(asset))
                Results.Add((Asset)asset.Clone());
        }
    }
}
catch

```

```

    {

    }
    try
    {
        if (asset.PersonShipping != null)
        {
            if (asset.PersonShipping.ToUpper().Contains(SearchTerm.ToUpper()))
            {
                if (!Results.Contains(asset))
                    Results.Add((Asset)asset.Clone());
            }
        }
    }
    catch
    {

    }
    try
    {
        if (asset.OrderNumber != "-1")
        {
            if
(asset.OrderNumber.ToString().ToUpper().Contains(SearchTerm.ToUpper()))
            {
                if (!Results.Contains(asset))
                    Results.Add((Asset)asset.Clone());
            }
        }
    }
    catch
    {

    }
}
return Results;
}
if (SearchFilter.ToLower() == "history")
{

    foreach (var aaa in AssetController.GetAllAssets())
    {

```

```

try
{
    foreach (var asset in aaa.History.History)
    {
        try
        {
            if (asset.AssetName != null)
            {
                if (asset.AssetName.ToUpper().Contains(SearchTerm.ToUpper()))
                {
                    if (!Results.Contains(asset))
                        Results.Add((Asset)asset.Clone());
                }
            }
        }
        catch
        {
        }
    }
    try
    {
        if (asset.AssetName != null)
        {
            if (asset.AssetName.ToUpper().Contains(SearchTerm.ToUpper()))
            {
                if (!Results.Contains(asset))
                    Results.Add((Asset)asset.Clone());
            }
        }
    }
    catch
    {
    }
}
try
{
    if (asset.Description != null)
    {
        if (asset.Description.ToUpper().Contains(SearchTerm.ToUpper()))
        {
            if (!Results.Contains(asset))

```

```

        Results.Add((Asset)asset.Clone());
    }
}
catch
{
}
try
{
    if (asset.ShipTo != null)
    {
        if (asset.ShipTo.ToUpper().Contains(SearchTerm.ToUpper()))
        {
            if (!Results.Contains(asset))
                Results.Add((Asset)asset.Clone());
        }
    }
}
catch
{
}
try
{
    if (asset.ServiceEngineer != null)
    {
        if (asset.ServiceEngineer.ToUpper().Contains(SearchTerm.ToUpper()))
        {
            if (!Results.Contains(asset))
                Results.Add((Asset)asset.Clone());
        }
    }
}
catch
{
}
try
{
    if (asset.PersonShipping != null)
    {

```

```

        if (asset.PersonShipping.ToUpper().Contains(SearchTerm.ToUpper()))
        {
            if (!Results.Contains(asset))
                Results.Add((Asset)asset.Clone());
        }
    }
}
catch
{
}
try
{
    if (asset.OrderNumber != "-1")
    {
        if
(asset.OrderNumber.ToString().ToUpper().Contains(SearchTerm.ToUpper()))
        {
            if (!Results.Contains(asset))
                Results.Add((Asset)asset.Clone());
        }
    }
}
catch
{
}
}
}
catch { }
}
foreach (var item in Results)
{
    item.IsHistoryItem = true;
}
return Results;
}

if (SearchFilter == "Date")
{
    return Results;
}

```



```

    }
    catch { }
    return new List<Asset>();
}

```

```

protected void AssetViewRepeater_ItemDataBound(object sender, RepeaterItemEventArgs
e)
{
    try
    {
        if (e.Item.ItemType == ListItemType.Item || e.Item.ItemType ==
ListItemType.AlternatingItem)
        {
            var a = e.Item.Controls[0];
            if (a == null) return;
            if (a is Templates.av_sm_tile_template)
            {
                var smtile = e.Item.Controls[0] as Templates.av_sm_tile_template;
                var isoutcontainer = smtile.FindControl("IsOutContainer") as HtmlGenericControl;
                if ((e.Item.DataItem as Asset).IsOut)
                {
                    var icl = isoutcontainer.Attributes["class"];
                    icl += " element-selected";
                    isoutcontainer.Attributes["class"] = icl;
                }
            }
            else
            {
                var icl = isoutcontainer.Attributes["class"];
                icl += " element-available";
                isoutcontainer.Attributes["class"] = icl;
            }
        }
        if (a is Templates.av_list_template)
        {
            var avlist = e.Item.Controls[0] as Templates.av_list_template;
            var ioi = avlist.FindControl("InOutIndicator") as HtmlGenericControl;
            if (ioi != null)
            {
                if ((e.Item.DataItem as Asset).IsOut)
                {
                    var icl = ioi.Attributes["class"];
                    icl += " bg-red";
                }
            }
        }
    }
}

```

```

        ioi.Attributes["class"] = icl;
    }
    else
    {
        var icl = ioi.Attributes["class"];
        icl += " bg-black";
        ioi.Attributes["class"] = icl;
    }

}

var isoutcontainer = avlist.FindControl("IsOutContainer") as HtmlGenericControl;
if ((e.Item.DataItem as Asset).IsOut)
{
    var icl = isoutcontainer.Attributes["class"];
    icl += " element-selected";
    isoutcontainer.Attributes["class"] = icl;

}
else
{
    var icl = isoutcontainer.Attributes["class"];
    icl += " element-available";
    isoutcontainer.Attributes["class"] = icl;
}
}

if (a is Templates.av_default_template)
{
    var d = e.Item.Controls[0] as Templates.av_default_template;
    var isoutcontainer = d.Controls[1] as HtmlGenericControl;
    if ((e.Item.DataItem as Asset).IsOut)
    {
        var icl = isoutcontainer.Attributes["class"];
        icl = icl.Replace("fg-white", "fg-red");
        isoutcontainer.Attributes["class"] = icl;
        return;
    }

}

}

}
catch {}
}

```

```

protected void UpdateView(string view, object datasource=null)
{
    if (datasource == null)
    { datasource = AssetController.GetAllAssets(); }
    switch (view)
    {
        case "list":
            AssetViewRepeater.HeaderTemplate =
Page.LoadTemplate("/Account/Templates/av_default_header_template.ascx");
            AssetViewRepeater.ItemTemplate =
Page.LoadTemplate("/Account/Templates/av_default_template.ascx");
            AssetViewRepeater.FooterTemplate =
Page.LoadTemplate("/Account/Templates/av_default_footer_template.ascx");
            AssetViewRepeater.DataSource = datasource;
            AssetViewRepeater.DataBind();
            Session["CurrentAvView"] = avSelectedView.Text;
            break;
        case "detail":

            break;
        case "smtile":
            AssetViewRepeater.ItemTemplate =
Page.LoadTemplate("/Account/Templates/av_list_template.ascx");
            AssetViewRepeater.DataSource = datasource;
            AssetViewRepeater.DataBind();
            Session["CurrentAvView"] = avSelectedView.Text;
            break;
        case "mdtile":
            AssetViewRepeater.HeaderTemplate =
Page.LoadTemplate("/Account/Templates/av_header_template.ascx");
            AssetViewRepeater.ItemTemplate =
Page.LoadTemplate("/Account/Templates/av_sm_tile_template.ascx");
            AssetViewRepeater.FooterTemplate =
Page.LoadTemplate("/Account/Templates/av_footer_template.ascx");
            AssetViewRepeater.DataSource = datasource;
            AssetViewRepeater.DataBind();
            Session["CurrentAvView"] = avSelectedView.Text;
            break;
        case "lgtile":
            AssetViewRepeater.HeaderTemplate =
Page.LoadTemplate("/Account/Templates/av_header_template.ascx");

```

```

        AssetViewRepeater.ItemTemplate =
Page.LoadTemplate("/Account/Templates/av_sm_tile_template.ascx");
        AssetViewRepeater.FooterTemplate =
Page.LoadTemplate("/Account/Templates/av_footer_template.ascx");
        AssetViewRepeater.DataSource = datasource;
        AssetViewRepeater.DataBind();
        Session["CurrentAvView"] = avSelectedView.Text;
        break;
    default:
        AssetViewRepeater.HeaderTemplate =
Page.LoadTemplate("/Account/Templates/av_default_header_template.ascx");
        AssetViewRepeater.ItemTemplate =
Page.LoadTemplate("/Account/Templates/av_default_template.ascx");
        AssetViewRepeater.FooterTemplate =
Page.LoadTemplate("/Account/Templates/av_default_footer_template.ascx");
        AssetViewRepeater.DataSource = datasource;
        AssetViewRepeater.DataBind();
        Session["CurrentAvView"] = avSelectedView.Text;
        break;
    }
}

protected void ChangeView_Click(object sender, EventArgs e)
{
    UpdateView(avSelectedView.Text);
}

protected void AssetSearchBtn_Click(object sender, EventArgs e)
{
    var searchfilter = "asset";
    var userinput = avSelectedSearch.Text;
    if (userinput!=null && userinput!="")
    {
        searchfilter = userinput;
    }
    UpdateView((Session["CurrentAvView"] as string),
AssetSearch(avSearchString.Text,searchfilter));
}

protected void FilterIsOut_Click(object sender, EventArgs e)
{
    var ret = from a in AssetController.GetAllAssets() where a.IsOut==true select a;
    UpdateView((Session["CurrentAvView"] as string), ret.ToList());
}

```

```

    }
    protected void FilterIsIn_Click(object sender, EventArgs e)
    {
        var ret = from a in AssetController.GetAllAssets() where a.IsOut==false select a;
        UpdateView((Session["CurrentAvView"] as string), ret.ToList());
    }

    protected void FilterIsDamaged_Click(object sender, EventArgs e)
    {
        var ret = from a in AssetController.GetAllAssets() where a.IsDamaged == true select a;
        UpdateView((Session["CurrentAvView"] as string), ret.ToList());
    }

    protected void FilterOnHold_Click(object sender, EventArgs e)
    {
        var ret = from a in AssetController.GetAllAssets() where a.OnHold == true select a;
        UpdateView((Session["CurrentAvView"] as string), ret.ToList());
    }

    protected void FilterCalibrated_Click(object sender, EventArgs e)
    {
        var ret = from a in AssetController.GetAllAssets() where a.IsCalibrated == true select a;
        UpdateView((Session["CurrentAvView"] as string), ret.ToList());
    }

    protected void ViewAll_Click(object sender, EventArgs e)
    {
        UpdateView((Session["CurrentAvView"] as string), AssetController.GetAllAssets());
    }
}
}

```

```
using Helpers;
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Threading.Tasks;
using System.Web;
using System.Web.Services;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Xml;
```

```
namespace Web_App_Master.Account
```

```
{
    public partial class AssetView : System.Web.UI.Page
    {
        public async Task UpdateAsync()
        {
            var currentview = Session["CurrentAvView"] as string;
            var x = await Web_App_Master.Pull.AssetsAsync();
            Session["SessionAssets"] = x.OrderBy(w => w.AssetNumber).ToList();
            var list = Session["SessionAssets"] as List<Asset>;
            if (Page.User.IsInRole("Admins") || Page.User.IsInRole("superadmin"))
            {
                UpdateView(currentview, list.ToList().OrderBy(w => w.AssetNumber));
            }
            else
            {
                var inonly = from a in list where a.IsOut == false select a;
                UpdateView(currentview, inonly.ToList().OrderBy(w => w.AssetNumber));
            }
        }

    }

    public async Task SearchAsync()
    {
        var searchfilter = "asset";
        var userInput = avSelectedSearch.Text;
        if (userinput != null && userInput != "")
```

```

{
    searchfilter = userInput;
}
var currentview = Session["CurrentAvView"] as string;
var x = await Web_App_Master.Pull.AssetsAsync();
Session["SessionAssets"] = x.OrderBy(w => w.AssetNumber).ToList();
var search = Request.QueryString["q"];
UpdateView(currentview, AssetSearch(search, searchfilter).OrderBy(w =>
w.AssetNumber));
}
public string IsSelected(object num)
{
    if ((bool)num == true)
        return "element-selected";
    else
        return "";
}
protected void Page_Load(object sender, EventArgs e)
{
    if (IsAsync)
    {
    }
    if (!IsPostBack)
    {
        if (Request.QueryString["q"]!=null)
        {
            RegisterAsyncTask(new PageAsyncTask(SearchAsync));
        }
        else
        {
            RegisterAsyncTask(new PageAsyncTask(UpdateAsync));
        }
    }
}

private List<Asset> AssetSearch(string SearchTerm, string SearchFilter="asset")
{

```

```

try
{
    List<Asset> Results = new List<Asset>();
    if (SearchFilter.ToLower() == "asset")
    {

        foreach (var asset in Session["SessionAssets"] as List<Asset>)
        {

            try
            {
                if (asset.AssetName != null)
                {
                    if (asset.AssetName.ToUpper().Contains(SearchTerm.ToUpper()))
                    {

                        if (!Results.Contains(asset))
                            Results.Add((Asset)asset.Clone());
                    }
                }
            }
            catch
            {

            }
            try
            {
                if (asset.AssetName != null)
                {
                    if (asset.AssetName.ToUpper().Contains(SearchTerm.ToUpper()))
                    {
                        if (!Results.Contains(asset))
                            Results.Add((Asset)asset.Clone());
                    }
                }
            }
            catch
            {

            }
            try
            {

```



```

        if (asset.Description != null)
        {
            if (asset.Description.ToUpper().Contains(SearchTerm.ToUpper()))
            {
                if (!Results.Contains(asset))
                    Results.Add((Asset)asset.Clone());
            }
        }
    }
    catch
    {
    }
    try
    {
        if (asset.ShipTo != null)
        {
            if (asset.ShipTo.ToUpper().Contains(SearchTerm.ToUpper()))
            {
                if (!Results.Contains(asset))
                    Results.Add((Asset)asset.Clone());
            }
        }
    }
    catch
    {
    }
    try
    {
        if (asset.ServiceEngineer != null)
        {
            if (asset.ServiceEngineer.ToUpper().Contains(SearchTerm.ToUpper()))
            {
                if (!Results.Contains(asset))
                    Results.Add((Asset)asset.Clone());
            }
        }
    }
    catch
    {
    }

```

```

    }
    try
    {
        if (asset.PersonShipping != null)
        {
            if (asset.PersonShipping.ToUpper().Contains(SearchTerm.ToUpper()))
            {
                if (!Results.Contains(asset))
                    Results.Add((Asset)asset.Clone());
            }
        }
    }
    catch
    {
    }
    try
    {
        if (asset.OrderNumber != "-1")
        {
            if
(asset.OrderNumber.ToString().ToUpper().Contains(SearchTerm.ToUpper()))
            {
                if (!Results.Contains(asset))
                    Results.Add((Asset)asset.Clone());
            }
        }
    }
    catch
    {
    }
}
Results = Results.GroupBy(x => x.AssetNumber).Select(x => x.First()).ToList();
return Results;
}
if (SearchFilter.ToLower() == "history")
{

    foreach (var aaa in Session["SessionAssets"] as List<Asset>)
    {
        try

```

```

{
    foreach (var asset in aaa.History.History)
    {
        try
        {
            if (asset.AssetNumber != null)
            {
                if (asset.AssetNumber.ToUpper().Contains(SearchTerm.ToUpper()))
                {
                    if (!Results.Contains(asset))
                        Results.Add((Asset)asset.Clone());
                }
            }
        }
        catch
        {
        }
    }
    try
    {
        if (asset.AssetName != null)
        {
            if (asset.AssetName.ToUpper().Contains(SearchTerm.ToUpper()))
            {
                if (!Results.Contains(asset))
                    Results.Add((Asset)asset.Clone());
            }
        }
    }
    catch
    {
    }
}
try
{
    if (asset.Description != null)
    {
        if (asset.Description.ToUpper().Contains(SearchTerm.ToUpper()))
        {
            if (!Results.Contains(asset))
                Results.Add((Asset)asset.Clone());
        }
    }
}

```

```

        }
    }
}
catch
{

}
try
{
    if (asset.ShipTo != null)
    {
        if (asset.ShipTo.ToUpper().Contains(SearchTerm.ToUpper()))
        {
            if (!Results.Contains(asset))
                Results.Add((Asset)asset.Clone());
        }
    }
}
catch
{

}
try
{
    if (asset.ServiceEngineer != null)
    {
        if (asset.ServiceEngineer.ToUpper().Contains(SearchTerm.ToUpper()))
        {
            if (!Results.Contains(asset))
                Results.Add((Asset)asset.Clone());
        }
    }
}
catch
{

}
try
{
    if (asset.PersonShipping != null)
    {
        if (asset.PersonShipping.ToUpper().Contains(SearchTerm.ToUpper()))

```

```

        {
            if (!Results.Contains(asset))
                Results.Add((Asset)asset.Clone());
        }
    }
}
catch
{
}
try
{
    if (asset.OrderNumber != "-1")
    {
        if
(asset.OrderNumber.ToString().ToUpper().Contains(SearchTerm.ToUpper()))
        {
            if (!Results.Contains(asset))
                Results.Add((Asset)asset.Clone());
        }
    }
}
catch
{
}
}
}
catch { }
}
foreach (var item in Results)
{
    item.IsHistoryItem = true;
}

return Results;
}

if (SearchFilter == "Date")
{
    return Results;
}

```

```

    }
    catch { }
    return new List<Asset>();
}

```

```

e) protected void AssetViewRepeater_ItemDataBound(object sender, RepeaterItemEventArgs
{
    try
    {
        if (e.Item.ItemType == ListItemType.Item || e.Item.ItemType ==
ListItemType.AlternatingItem)
        {
            var a = e.Item.Controls[0];
            if (a == null) return;
            if (a is Templates.av_sm_tile_template)
            {
                var smtile = e.Item.Controls[0] as Templates.av_sm_tile_template;
                var isoutcontainer = smtile.FindControl("IsOutContainer") as HtmlGenericControl;
                if ((e.Item.DataItem as Asset).IsOut)
                {
                    var icl = isoutcontainer.Attributes["class"];
                    icl += " element-selected";
                    isoutcontainer.Attributes["class"] = icl;
                }
                if ((e.Item.DataItem as Asset).OnHold)
                {
                    var icl = isoutcontainer.Attributes["class"];
                    icl += " element-onhold";
                    isoutcontainer.Attributes["class"] = icl;
                }
                if ((e.Item.DataItem as Asset).IsDamaged)
                {
                    var icl = isoutcontainer.Attributes["class"];
                    icl += " element-damaged";
                    isoutcontainer.Attributes["class"] = icl;
                }
            }
            if (a is Templates.av_list_template)
            {
                var avlist = e.Item.Controls[0] as Templates.av_list_template;

```

```

var ioi = avlist.FindControl("InOutIndicator") as HtmlGenericControl;
if (ioi!=null)
{
    if ((e.Item.DataItem as Asset).IsOut)
    {
        var icl = ioi.Attributes["class"];
        icl += " bg-red";
        ioi.Attributes["class"] = icl;
    }
    else
    {
        var icl = ioi.Attributes["class"];
        icl += " bg-black";
        ioi.Attributes["class"] = icl;
    }
    if ((e.Item.DataItem as Asset).OnHold)
    {
        var icl = ioi.Attributes["class"];
        icl += " bg-violet";
        ioi.Attributes["class"] = icl;
    }
    if ((e.Item.DataItem as Asset).IsDamaged)
    {
        var icl = ioi.Attributes["class"];
        icl += " bg-amber";
        ioi.Attributes["class"] = icl;
    }
}

var isoutcontainer = avlist.FindControl("IsOutContainer") as HtmlGenericControl;
if ((e.Item.DataItem as Asset).IsOut)
{
    var icl = isoutcontainer.Attributes["class"];
    icl += " element-selected";
    isoutcontainer.Attributes["class"] = icl;
}
else
{
    var icl = isoutcontainer.Attributes["class"];
    icl += " element-available";
    isoutcontainer.Attributes["class"] = icl;
}

```

```

    }
}
if (a is Templates.av_default_template)
{
    var d = e.Item.Controls[0] as Templates.av_default_template;
    var isoutcontainer = d.Controls[1] as HtmlGenericControl;
    if ((e.Item.DataItem as Asset).IsOut)
    {
        var icl = isoutcontainer.Attributes["class"];
        icl = icl.Replace("fg-white", "fg-red");
        isoutcontainer.Attributes["class"] = icl;

    }
    if ((e.Item.DataItem as Asset).OnHold)
    {
        var icl = isoutcontainer.Attributes["class"];
        icl = icl.Replace("fg-white", "fg-violet");
        isoutcontainer.Attributes["class"] = icl;

    }
    if ((e.Item.DataItem as Asset).IsDamaged)
    {
        var icl = isoutcontainer.Attributes["class"];
        icl = icl.Replace("fg-white", "fg-amber");
        isoutcontainer.Attributes["class"] = icl;

    }
    return;
}
if (a is Templates.icon_view)
{
    var template = e.Item.Controls[0] as Templates.icon_view;
    var dl_link = template.Controls[5] as HtmlGenericControl;
    var inoutcontainer = template.Controls[3] as HtmlGenericControl;
    if ((e.Item.DataItem as Asset).IsOut)
    {
        dl_link.InnerHtml=dl_link.InnerHtml.Replace("c-lime", "c-red");
        dl_link.InnerHtml = dl_link.InnerHtml.Replace("download", "upload");
    }
    //if ((e.Item.DataItem as Asset).OnHold)
    //{
    //    var icl = inoutcontainer.Attributes["class"];

```



```

        // icl += " bg-sg-violet";
        // inoutcontainer.Attributes["class"] = icl;

        //}
        //if ((e.Item.DataItem as Asset).IsDamaged)
        //{
        //    var icl = inoutcontainer.Attributes["class"];
        //    icl += " bg-sg-red";
        //    inoutcontainer.Attributes["class"] = icl;

        //}
        return;
    }
}
}
catch { }
}

```

```

protected void UpdateView(string view, object datasource=null)
{
    if (datasource == null)
    { datasource = AssetController.GetAllAssets().OrderBy(w => w.AssetNumber); }
    switch (view)
    {
        case "icon":
            AssetViewRepeater.HeaderTemplate = null;
            AssetViewRepeater.FooterTemplate = null;
            AssetViewRepeater.ItemTemplate =
Page.LoadTemplate("/Account/Templates/icon_view.ascx");
            AssetViewRepeater.DataSource = datasource;
            AssetViewRepeater.DataBind();
            Session["CurrentAvView"] = avSelectedView.Text;
            break;
        case "list":
            AssetViewRepeater.HeaderTemplate =
Page.LoadTemplate("/Account/Templates/av_default_header_template.ascx");
            AssetViewRepeater.ItemTemplate =
Page.LoadTemplate("/Account/Templates/av_default_template.ascx");
            AssetViewRepeater.FooterTemplate =
Page.LoadTemplate("/Account/Templates/av_default_footer_template.ascx");
            AssetViewRepeater.DataSource = datasource;
            AssetViewRepeater.DataBind();

```

```

        Session["CurrentAvView"] = avSelectedView.Text;
        break;
    case "detail":

        break;
    case "smtile":
        AssetViewRepeater.ItemTemplate =
Page.LoadTemplate("/Account/Templates/av_list_template.ascx");
        AssetViewRepeater.DataSource = datasource;
        AssetViewRepeater.DataBind();
        Session["CurrentAvView"] = avSelectedView.Text;
        break;
    case "mdtile":
        AssetViewRepeater.HeaderTemplate =
Page.LoadTemplate("/Account/Templates/av_header_template.ascx");
        AssetViewRepeater.ItemTemplate =
Page.LoadTemplate("/Account/Templates/av_sm_tile_template.ascx");
        AssetViewRepeater.FooterTemplate =
Page.LoadTemplate("/Account/Templates/av_footer_template.ascx");
        AssetViewRepeater.DataSource = datasource;
        AssetViewRepeater.DataBind();
        Session["CurrentAvView"] = avSelectedView.Text;
        break;
    case "lgtile":
        AssetViewRepeater.HeaderTemplate =
Page.LoadTemplate("/Account/Templates/av_header_template.ascx");
        AssetViewRepeater.ItemTemplate =
Page.LoadTemplate("/Account/Templates/av_sm_tile_template.ascx");
        AssetViewRepeater.FooterTemplate =
Page.LoadTemplate("/Account/Templates/av_footer_template.ascx");
        AssetViewRepeater.DataSource = datasource;
        AssetViewRepeater.DataBind();
        Session["CurrentAvView"] = avSelectedView.Text;
        break;
    default:
        AssetViewRepeater.HeaderTemplate = null;
        AssetViewRepeater.FooterTemplate = null;
        AssetViewRepeater.ItemTemplate =
Page.LoadTemplate("/Account/Templates/icon_view.ascx");
        AssetViewRepeater.DataSource = datasource;
        AssetViewRepeater.DataBind();
        Session["CurrentAvView"] = avSelectedView.Text;

```

```

        break;
    }
}
protected void ChangeView_Click(object sender, EventArgs e)
{
    UpdateView(avSelectedView.Text);
}

protected void AssetSearchBtn_Click(object sender, EventArgs e)
{
    var searchfilter = "asset";
    var userinput = avSelectedSearch.Text;
    if (userinput!=null && userinput!="")
    {
        searchfilter = userinput;
    }
    UpdateView((Session["CurrentAvView"] as string),
AssetSearch(avSearchString.Text,searchfilter).OrderBy(w => w.AssetNumber));
}
protected void FilterIsOut_Click(object sender, EventArgs e)
{
    var ret = from a in Session["SessionAssets"] as List<Asset> where a.IsOut==true select a;
    UpdateView((Session["CurrentAvView"] as string), ret.ToList().OrderBy(w =>
w.AssetNumber));
}
protected void FilterIsIn_Click(object sender, EventArgs e)
{
    var ret = from a in Session["SessionAssets"] as List<Asset> where a.IsOut==false select a;
    UpdateView((Session["CurrentAvView"] as string), ret.ToList().OrderBy(w =>
w.AssetNumber));
}
protected void FilterIsDamaged_Click(object sender, EventArgs e)
{
    var ret = from a in Session["SessionAssets"] as List<Asset> where a.IsDamaged == true
select a;
    UpdateView((Session["CurrentAvView"] as string), ret.ToList().OrderBy(w =>
w.AssetNumber));
}

protected void FilterOnHold_Click(object sender, EventArgs e)

```

```

    {
        var ret = from a in Session["SessionAssets"] as List<Asset> where a.OnHold == true select
a;
        UpdateView((Session["CurrentAvView"] as string), ret.ToList().OrderBy(w =>
w.AssetNumber));
    }

    protected void FilterCalibrated_Click(object sender, EventArgs e)
    {
        var ret = from a in Session["SessionAssets"] as List<Asset> where a.IsCalibrated == true
select a;
        UpdateView((Session["CurrentAvView"] as string), ret.ToList().OrderBy(w =>
w.AssetNumber));
    }

    protected void ViewAll_Click(object sender, EventArgs e)
    {
        var x = Session["SessionAssets"] as List<Asset>;
        x = x.OrderBy(w => w.AssetNumber).ToList();
        UpdateView((Session["CurrentAvView"] as string), x);
    }

    protected void RefreshBtn_Click(object sender, EventArgs e)
    {
        RegisterAsyncTask(new PageAsyncTask(UpdateAsync));
    }

    protected void Refresher_Click(object sender, EventArgs e)
    {
        RegisterAsyncTask(new PageAsyncTask(UpdateAsync));
    }

    protected void CreateAssetBtn_Click(object sender, EventArgs e)
    {
    }
}
}

```

```
//-----  
// <auto-generated>  
//   This code was generated by a tool.  
//  
//   Changes to this file may cause incorrect behavior and will be lost if  
//   the code is regenerated.  
// </auto-generated>  
//-----
```

```
namespace Web_App_Master.Account {
```

```
    public partial class AssetView {
```

```
        /// <summary>  
        /// avSearchString control.  
        /// </summary>  
        /// <remarks>  
        /// Auto-generated field.  
        /// To modify move field declaration from designer file to code-behind file.  
        /// </remarks>
```

```
        protected global::System.Web.UI.WebControls.TextBox avSearchString;
```

```
        /// <summary>  
        /// AssetSearchBtn control.  
        /// </summary>  
        /// <remarks>  
        /// Auto-generated field.  
        /// To modify move field declaration from designer file to code-behind file.  
        /// </remarks>
```

```
        protected global::System.Web.UI.WebControls.LinkButton AssetSearchBtn;
```

```
        /// <summary>  
        /// avAssetSearchBtn control.  
        /// </summary>  
        /// <remarks>  
        /// Auto-generated field.  
        /// To modify move field declaration from designer file to code-behind file.  
        /// </remarks>
```

```
        protected global::System.Web.UI.HtmlControls.HtmlAnchor avAssetSearchBtn;
```

```
        /// <summary>
```

```
/// avHistorySearchBtn control.  
/// </summary>  
/// <remarks>  
/// Auto-generated field.  
/// To modify move field declaration from designer file to code-behind file.  
/// </remarks>  
protected global::System.Web.UI.HtmlControls.HtmlAnchor avHistorySearchBtn;
```

```
/// <summary>  
/// ViewAll control.  
/// </summary>  
/// <remarks>  
/// Auto-generated field.  
/// To modify move field declaration from designer file to code-behind file.  
/// </remarks>  
protected global::System.Web.UI.WebControls.LinkButton ViewAll;
```

```
/// <summary>  
/// FilterIsOut control.  
/// </summary>  
/// <remarks>  
/// Auto-generated field.  
/// To modify move field declaration from designer file to code-behind file.  
/// </remarks>  
protected global::System.Web.UI.WebControls.LinkButton FilterIsOut;
```

```
/// <summary>  
/// FilterIsIn control.  
/// </summary>  
/// <remarks>  
/// Auto-generated field.  
/// To modify move field declaration from designer file to code-behind file.  
/// </remarks>  
protected global::System.Web.UI.WebControls.LinkButton FilterIsIn;
```

```
/// <summary>  
/// FilterIsDamaged control.  
/// </summary>  
/// <remarks>  
/// Auto-generated field.  
/// To modify move field declaration from designer file to code-behind file.  
/// </remarks>
```

protected global::System.Web.UI.WebControls.LinkButton FilterIsDamaged;

/// <summary>

/// FilterOnHold control.

/// </summary>

/// <remarks>

/// Auto-generated field.

/// To modify move field declaration from designer file to code-behind file.

/// </remarks>

protected global::System.Web.UI.WebControls.LinkButton FilterOnHold;

/// <summary>

/// FilterCalibrated control.

/// </summary>

/// <remarks>

/// Auto-generated field.

/// To modify move field declaration from designer file to code-behind file.

/// </remarks>

protected global::System.Web.UI.WebControls.LinkButton FilterCalibrated;

/// <summary>

/// CheckoutUpdatePanel control.

/// </summary>

/// <remarks>

/// Auto-generated field.

/// To modify move field declaration from designer file to code-behind file.

/// </remarks>

protected global::System.Web.UI.UpdatePanel CheckoutUpdatePanel;

/// <summary>

/// AssetViewRepeater control.

/// </summary>

/// <remarks>

/// Auto-generated field.

/// To modify move field declaration from designer file to code-behind file.

/// </remarks>

protected global::System.Web.UI.WebControls.Repeater AssetViewRepeater;

/// <summary>

/// avSelectedSearch control.

/// </summary>

/// <remarks>

```

/// Auto-generated field.
/// To modify move field declaration from designer file to code-behind file.
/// </remarks>
protected global::System.Web.UI.WebControls.TextBox avSelectedSearch;

/// <summary>
/// avSelectedView control.
/// </summary>
/// <remarks>
/// Auto-generated field.
/// To modify move field declaration from designer file to code-behind file.
/// </remarks>
protected global::System.Web.UI.WebControls.TextBox avSelectedView;

/// <summary>
/// avChangeView control.
/// </summary>
/// <remarks>
/// Auto-generated field.
/// To modify move field declaration from designer file to code-behind file.
/// </remarks>
protected global::System.Web.UI.WebControls.Button avChangeView;

/// <summary>
/// Master property.
/// </summary>
/// <remarks>
/// Auto-generated property.
/// </remarks>
public new Web_App_Master.SiteMaster Master {
    get {
        return ((Web_App_Master.SiteMaster)(base.Master));
    }
}
}
}

```



```

        </asp:PlaceHolder>
    </nav>

    <div class="bg-metro" style=" padding-top:60px !important"></div>
    <asp:MultiView ID="CheckInMultiView" ActiveViewIndex="0" runat="server">
        <asp:View ID="CheckInView" runat="server">
            <div class="row">
                <div class="col-md-12">
                    <div class="awp_box rounded bg-sg-title shadow">
                        <div class="awp_box_title bg-sg-title">
                            <span class="fg-white shadow-metro-black">Items:</span></div>
                            <div class="awp_box_content bg-sg-box">
                                <div class="row">
                                    <asp:UpdatePanel runat="server" ID="CheckInPageUpdatePanel"
ClientIDMode="Static" ChildrenAsTriggers="true" UpdateMode="Conditional">
                                        <ContentTemplate>
                                            <asp:Repeater runat="server" ID="FCIrepeater"
ClientIDMode="Static" OnItemCommand="FinalCheckInRepeater_ItemCommand" >
                                                <ItemTemplate>
                                                    <div class="file " title=" draggable="true"
onclick="BarcodeScanned('<## Eval("AssetNumber")%>', '<## Eval("IsHistoryItem")%>', '<##
Eval("DateShippedTicks")%>');">
                                                        <div id=" file-color " class="file-color <## Eval("Color") %>"
style="width:100%; height:100%;"></div>

                                                        <div class="i-icon">
                                                            <img style="box-shadow: rgba(0, 0, 0, 0.70) 0px 0px 10px;"
src='<## Eval("FirstImage")+".r?w=133&h=100" %>' />
                                                        </div>

                                                        <div runat="server" class="i-title"><strong class="fg-white
shadow-metro-black"><## Eval("AssetName") %></strong> </div>

                                                        <div class="i-check">
                                                            <span id="InOutIndicator" runat="server" class="edge-badge fg-
white shadow-metro-black icon-badge " style="z-index:3 !important;"><strong><##
Eval("AssetNumber")%></strong></span></div>

                                                            <div runat="server" id="DownloadLink" class="i-download"
style="margin-top:-4px;padding-left:2px" onclick="event.stopPropagation();"> <asp:LinkButton
ID="RemoveFromCheckOutButton" CommandName="Delete" CommandArgument='<##

```

```

Eval("AssetNumber")%>' runat="server" > <i title="Add To Task List" id="dl_link" style="font-
size:1.5em;" class="glyphicon glyphicon-remove c-red av-hand-cursor"></i></asp:LinkButton>

        </div>

    </div>

    </ItemTemplate>
</asp:Repeater>
</ContentTemplate>

</asp:UpdatePanel>

</div>
</div>
</div>
</div>

</div>

</asp:View>
<asp:View ID="ReportView" runat="server">
    <script type="text/javascript">

        $(document).ready(function () {

            function calcIframeDivHeight() {
                try {
                    var newHeight = 0;
                    var myViewHeight = jQuery(".main-content").height();
                    newHeight = myViewHeight - 160;
                    $("#IframeDiv").height(newHeight);
                    $("#MainContent_ReportFrame").height(newHeight);
                    $("#MainContent_ReportFrame").width($("#IframeDiv").width());
                } catch (er) { }
            }

        }

        // run on page load

```

```
try {calcIframeDivHeight(); } catch (er) { }
```

```
// run on window resize event
```

```
$(window).resize(function () {  
    try {calcIframeDivHeight(); } catch (er) { }  
});
```

```
});
```

```
</script>
```

```
<div class="row">
```

```
<div class="col-md-12">
```

```
<!-- <span class="awp-save-btn bg-green fg-white shadow mif-ani-flash "  
onclick="printpackingslip()"><i title="Print" style="font-size:2em; vertical-align:top" class="mif-  
printer av-hand-cursor fg-white shadow-metro-black"></i></span>-->
```

```
<div class="awp_box rounded bg-sg-title shadow">
```

```
<div class="awp_box_title bg-sg-title">
```

```
<span class="fg-white shadow-metro-black">Receiving Report:</span>
```

```
</div>
```

```
<div class="awp_box_content bg-sg-box">
```

```
<div id="ReceivingHidden" style="display:none"><asp:Literal ID="ReceivingLink"  
runat="server"></asp:Literal> </div>
```

```
<div id="IframeDiv" style="min-height:100px;height:300px;">
```

```
<iframe frameborder="0" style="position:absolute; min-height:100px;"  
id="ReportFrame" runat="server" src="#" ></iframe>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</asp:View>
```

```
</asp:MultiView>
```

```
</asp:Content>
```

```

using Helpers;
using Microsoft.AspNet.Identity.Owin;
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace Web_App_Master.Account
{
    public partial class CheckIn : System.Web.UI.Page
    {
        public void NotifyCheckInEmail(List<List<Asset>> assets)
        {
            if (Global.Library.Settings.TESTMODE == false)
            {
                foreach(var group in assets)
                {
                    try
                    {
                        EmailHelper.SendCheckOutNoticeAsync(group,
Global.Library.Settings.CheckInMessage);
                    }
                    catch { }

                }

            }
        }
        public void SaveToUserPersistantLog()
        {
            try
            {
                var ud = Session["PersistingUserData"] as Data.UserData;
                string pdf = Session["CurrentCheckinReport"] as string;
                Data.Attachment a = new Data.Attachment();
                a.Email = ud.Email;
                a.Name = ud.Name;
                a.Files.Add(new Data.FileReference(Session["CurrentCheckInOrdernumber"] as string,
pdf));
            }
        }
    }
}

```

```
        ud.Attachments.Add(a);
        ud.Log.Add(new Data.LogEntry("Checked In Order #" +
Session["CurrentCheckInOrdernumber"] as string, Session["CurrentCheckInOrdernumber"] as
string));
```

```
        SettingsDBData db = new SettingsDBData();
        db.Appname = ud.Guid;
        db.XmlData = ud.SerializeToXmlString(ud);

        Push.Setting(db);
    }
    catch { }
}
```

```
protected void Page_PreInit(object sender, EventArgs e)
{
    this.SiteMaster().OnPanelUpdate += Checkout_OnPanelUpdate;
}
private void Checkout_OnPanelUpdate(object sender, UpdateRequestEvent e)
{
    FCRepeater.DataBind();
    CheckInPageUpdatePanel.Update();
}
```

```
public void BindCheckIn()
{
    var checkin = Session["CheckIn"] as List<Asset>;
    if (checkin == null) checkin = new List<Asset>();
    FCRepeater.DataSource = checkin;
    FCRepeater.DataBind();
}
```

```
protected void Page_Load(object sender, EventArgs e)
{
    if (!IsPostBack)
    {
        if (!User.Identity.IsAuthenticated)
        {
            Response.Redirect("/Account/Login");
        }
    }
}
```

```

    }
}
    this.PreInit += Page_PreInit;
BindCheckIn();
Button btn = (Button)this.Master.FindControl("UpdateAllCarts");
AsyncPostBackTrigger Trigger1 = new AsyncPostBackTrigger();
Trigger1.ControlID = btn.ID;
Trigger1.EventName = "Click";
if (CheckInPageUpdatePanel.Triggers.Count == 1)
    CheckInPageUpdatePanel.Triggers.Add(Trigger1);
}

protected void Finalize_Click(object sender, EventArgs e)
{
    //Generate report
    //sort assets in list into groups of OrderNumber
    var list = (from a in Session["CheckIn"] as List<Asset> select
a.OrderNumber).Distinct().ToList();
    List<Asset> finalized = new List<Asset>();
    List<string> filenames = new List<string>();
    List<List<Asset>> subEmails = new List<List<Asset>>();
    foreach(var number in list)
    {
        var sublist = (from a in Session["CheckIn"] as List<Asset> where a.OrderNumber ==
number select a).ToList();
        var files = CreateReceivingReport(sublist);
        SaveToUserPersistantLog();
        filenames.AddRange(files);
        finalized.AddRange(sublist);
        subEmails.Add(sublist);
    }
    FinalizeCheckIn(finalized);
    Session["CheckInReportFileNameList"] = filenames;
    //combine all reports into one and display
    CombineReports(filenames);
    ReportIcon.Shake();
    CheckInMultiView.ActiveViewIndex = 1;
    if (!Global.Library.Settings.TESTMODE)
    {
        Finalize.Enabled = false;
        NotifyCheckInEmail(subEmails);
    }
}

```

ApplyChangesButton.Visible = false;

LeavePlaceholder.Visible = true;

}

protected void FinalizeCheckIn(List<Asset> assets)

{

try

{

foreach (var asset in assets)

{

asset.IsOut = false;

var history = asset.Clone() as Asset;

history.IsHistoryItem = true;

asset.History.History.Add(history);

asset.ShipTo = "";

asset.PersonShipping = "";

asset.ServiceEngineer = "";

asset.DateRecieved = DateTime.Now;

//save

Asset rem = null;

foreach (var a in Global.Library.Assets)

{

if (a.AssetNumber == asset.AssetNumber) rem = a;

}

if (rem != null)

{

Global.Library.Assets.Remove(rem);

Global.Library.Assets.Add(asset);

}

AssetController.UpdateAsset(asset);

}

Session["CheckIn"] = new List<Asset>();

}

catch {}

}

protected void CombineReports(List<string> files)

{

try

{


```

var guid = Guid.NewGuid().ToString();
var filename = guid + ".pdf";
Session["CombinedReportPdf"] = ReportFrame.Src = "/Account/Receiving/" + filename; ;
var dest = Server.MapPath("/Account/Receiving/" + filename);
if (files.Count == 0) return;
if (files.Count==1)
{

    Pdf.Flatten(files[0],dest);
    //File.Copy(files[0], dest);
    ReportFrame.Src = "/Account/Receiving/" + filename;
    return;
}

var result = Pdf.Merge(files, dest,true);
if (result)
{
    ReportFrame.Src = "/Account/Receiving/" + filename;

}
}
catch { }

}

```

```

protected void FinalCheckInRepeater_ItemCommand(object source,
RepeaterCommandEventArgs e)
{
    if (e.CommandName.ToLower() == "delete")
    {
        Asset rem = null;
        foreach (var ass in Session["CheckIn"] as List<Asset>)
        {
            if (ass.AssetId == e.CommandArgument as string)
            {
                rem = ass;
            }
        }
    }
    try

```

```

        {
            (Session["CheckIn"] as List<Asset>).Remove(rem);
        }
        catch { }
        this.UpdateAll();
    }

}

protected void Unnamed_CheckedChanged(object sender, EventArgs e)
{
    var cb = sender as CheckBox;
    if (cb.Checked)
    {
        foreach (var ass in Session["CheckIn"] as List<Asset>)
        {
            if (ass.AssetId == cb.ID)
            {
                ass.IsDamaged = true;
                AssetController.UpdateAsset(ass);
            }
        }
        this.UpdateAll();
    }
    else
    {
        foreach (var ass in Session["CheckIn"] as List<Asset>)
        {
            if (ass.AssetId == cb.ID)
            {
                ass.IsDamaged = false;
                AssetController.UpdateAsset(ass);
            }
        }
        this.UpdateAll();
    }
}

protected void CheckInViewBtn_Click(object sender, EventArgs e)
{
    CheckInMultiView.ActiveViewIndex = 0;
}

```

```

protected void ReportViewBtn_Click(object sender, EventArgs e)
{
    ReportIcon.Quiet();
    CheckInMultiView.ActiveViewIndex = 1;
}

int filecount = 1;
protected List<string> CreateReceivingReport(List<Asset> assets)
{
    List<string> filenames = new List<string>();
    try
    {
        //get filename
        string n = DateTime.Now.ToString().Replace("/", "-");
        n = n.Replace(":", "-");
        string info = n + "_" + assets.FirstOrDefault().OrderNumber;
        info = info.Replace("\\", "").Replace(":", "").Replace("*", "").Replace("?", "").Replace("\",",
        "").Replace("<", "").Replace(">", "").Replace("|", "").Replace(" ", "_");
        string path = "/Account/Receiving/" + info + ".pdf";
        Session["CurrentCheckInOrdernumber"] = assets[0].OrderNumber;
        var ud = Session["SessionUserData"] as Data.UserData;
        ud.Log.Add(new Data.LogEntry("Checked In Order#" + assets[0].OrderNumber, path));
        ReceivingLink.Text = path;
        var checker = Session["CurrentCheckinReport"] as string;
        if (checker==path)
        {
            path = path.Replace(".pdf", "");
            path += filecount.ToString();
            filecount++;
            path += ".pdf";
        }
        Session["CurrentCheckinReport"] = path;
        foreach(var ass in assets)
        {
            ass.ReturnReport = "/Account/Receiving/" + info + ".pdf";
        }
        var path2 = Server.MapPath(path);
        filenames.Add(path2);
        string from = "";
        string customerattn = assets[0].ShipTo;
        try
    
```

```

{
    foreach (var customer in Global.Library.Settings.Customers)
    {
        if (customer.CompanyName == assets[0].ShipTo)
        {
            from = customer.CompanyName + Environment.NewLine;
            from += customer.Address + Environment.NewLine;
            from += customer.Address2 + Environment.NewLine;
            from += customer.Address3 + Environment.NewLine;
            from += customer.City + Environment.NewLine;
            from += customer.State + ", " + customer.Postal + Environment.NewLine;
            from += customer.Country + Environment.NewLine;
            customerattn = customer.Attn;
        }
    }
}
catch { from += customerattn; }
//From

```

//Define Keys

```

var headerKeys = new Dictionary<string, string>() {
{"ShipDate", assets.FirstOrDefault().DateShippedString},
{"Tracking", assets.FirstOrDefault().PersonShipping},
{"Signature", customerattn},
{"Attn", ""},
{"Po", ""},
{"To", from},
{"From", ""},
{"Weight", "Report #::"+ info+".pdf"},
{"Ordernumber", "Order #::"+ assets.FirstOrDefault().OrderNumber} };

```

```

var keys = new Dictionary<string, string>();
headerKeys.ToList().ForEach(x => keys.Add(x.Key, x.Value));
int i = 1;
foreach (var asset in assets)

```

```

{
    if (i > 13)
    {
        //create templated pdf packingslip with default stamping action
        Pdf.PackingSlip p = new Pdf.PackingSlip(new
Templates.StarragReceivingReportTemplate());

```

```

        p.CreateAndFill(path2, keys, p.StampAction);
        //fill new page with new assets
        keys = new Dictionary<string, string>();
        headerKeys.ToList().ForEach(x => keys.Add(x.Key, x.Value));
        i = 1;
        path2 = path2.Replace(".pdf", "") + i.ToString() + ".pdf";
        filenames.Add(path2);
    }
    keys.Add("Desc" + i.ToString(), asset.AssetName);
    keys.Add("PartNo" + i.ToString(), asset.AssetName);
    keys.Add("Qty" + i.ToString(), "1");

    ++i;
}

//create templated pdf packingslip with default stamping action
Pdf.PackingSlip ps = new Pdf.PackingSlip(new
Templates.StarragReceivingReportTemplate());
var result = ps.CreateAndFill(path2, keys, ps.StampAction);

}
catch {}
return filenames;
}

}
}

```

```
<%@ Page Title="Check Out" Language="C#" MasterPageFile="~/Site.Master" Async="true"
AutoEventWireup="true" CodeBehind="Checkout.aspx.cs"
Inherits="Web_App_Master.Account.Checkout1" %>
<asp:Content ID="CheckOutContent" ContentPlaceHolderID="MainContent" runat="server">
```

```
    <nav class="navbar navbar-inverse bg-grayDark" style="margin-top:0px !important;
height:auto; width:100%; left:0px!important; border-radius:0px!important; position:fixed !important;
z-index:25000;">
```

```
        <ul class="nav navbar-nav starrag-menu">
```

```
            <li class="fg-white"><h2>Checkout</h2> </li>
```

```
            <li>
```

```
                <asp:LinkButton ID="ShippingViewBtn" OnClick="ShippingViewBtn_Click"
ToolTip="Shipping Options" ClientIDMode="Static" runat="server"><span id="ShipLink"
runat="server" class=" glyphicon glyphicon-cloud-download md-glyph"></span></asp:LinkButton>
```

```
            </li>
```

```
            <li>
```

```
                <asp:LinkButton ID="PackingSlipViewBtn" OnClick="PackingSlipViewBtn_Click"
ToolTip="Report" ClientIDMode="Static" runat="server"><span id="ReportLink" runat="server"
class=" glyphicon glyphicon-file md-glyph"></span></asp:LinkButton>
```

```
            </li>
```

```
        </ul>
```

```
        <ul class="nav navbar-nav navbar-right starrag-menu">
```

```
            <asp:PlaceHolder ID="FinalizeHolder" runat="server" Visible="true">
```

```
                <li>
```

```
                    <a href="#" onclick="ShowDiv('FinalizeConfirmDialog')">
```

```
                        <span class="shadow-metro-black"><i title="Finalize" style="font-size:1em;"
class="glyphicon glyphicon-floppy-disk av-hand-cursor shadow-metro-
black"></i><strong>&nbsp;&nbsp;  Finalize</strong></span></a>
```

```
                </li>
```

```
            </asp:PlaceHolder>
```

```
            <asp:PlaceHolder ID="LeavePlaceHolder" runat="server" Visible="false">
```

```
                <li>
```

```
                    <a href="/Default">
```

```
                        <span class="shadow-metro-black"><i title="Leave" style="font-size:1em;" class="mif-
cross av-hand-cursor shadow-metro-black"></i><strong>&nbsp;&nbsp; 
Leave</strong></span></a>
```

```
                </li>
```

```
            </asp:PlaceHolder>
```

```
        </ul>
```

```
    </nav>
```

```

<div style="padding-top:60px;"></div>
<asp:PlaceHolder ID ="MessagePlaceHolder" ClientIDMode="Static" runat="server"
Visible="false">
    <div class="row">
        <div class="col-md-12">
            <div class="awp_box rounded bg-sg-title shadow">
                <div class="awp_box_title bg-sg-title">
                    <span class="fg-white shadow-metro-black"><span class="mif-warning mif-ani-
flash mif-ani-slow"></span></span>
                </div>
                <div class="awp_box_content bg-red fg-white shadow-metro-black">
                    <asp:Literal ID="ErrorMessage" runat="server"></asp:Literal>
                </div>
            </div>
        </div>
    </div>
</div>

</asp:PlaceHolder>
<asp:MultiView ID="CheckoutView" ActiveViewIndex="0" runat="server">
    <asp:View ID="ShippingView" ClientIDMode="Static" runat="server">
        <div id ="ShipmentInformationGroup" class="row">
            <div class="col-md-3">

                <div class="awp_box rounded bg-sg-title shadow">
                    <div class="awp_box_title bg-sg-title">
                        <span class="fg-white shadow-metro-black bold"><span class="mif-home mif-
2x"></span>Source</span>
                    </div>
                    <div class="awp_box_content bg-sg-box">
                        <input runat="server" id="SprCompany" type="text" class="form-control"
placeholder="CompanyName">
                        <input runat="server" id="SprAddr" type="text" class="form-control"
placeholder="Address">
                        <input runat="server" id="SprAddr2" type="text" class="form-control"
placeholder="Address Line #2">
                        <input runat="server" id="SprCty" type="text" class="form-control"
placeholder="City">
                        <input runat="server" id="SprState" type="text" class="form-control"
placeholder="State">
                        <input runat="server" id="SprPostal" type="text" class="form-control"
placeholder="Postal">
                        <input runat="server" id="SprCountry" type="text" class="form-control"

```

```

placeholder="Country">
    <input runat="server" id="SprName" type="text" class="form-control"
placeholder="Attn">
    <input runat="server" id="SprEmail" type="text" class="form-control"
placeholder="Email">
    <input runat="server" id="SprPhone" type="text" class="form-control"
placeholder="Phone">
    </div>
</div>
</div>
<div class="col-md-3">

    <div class="awp_box rounded bg-sg-title shadow">
        <div class="awp_box_title bg-sg-title">
            <span class="fg-white shadow-metro-black bold"><span class="mif-my-
location mif-2x"></span>Destination</span></div>
            <div class="awp_box_content bg-sg-box">
                <input runat="server" id="ToCompany" type="text" class="form-control"
placeholder="CompanyName">
                <input runat="server" id="ToAddr" type="text" class="form-control"
placeholder="Address">
                <input runat="server" id="ToAddr2" type="text" class="form-control"
placeholder="Address Line #2">
                <input runat="server" id="ToCty" type="text" class="form-control"
placeholder="City">
                <input runat="server" id="ToState" type="text" class="form-control"
placeholder="State">
                <input runat="server" id="ToPostal" type="text" class="form-control"
placeholder="Postal">
                <input runat="server" id="ToCountry" type="text" class="form-control"
placeholder="Country">
                <input runat="server" id="ToName" type="text" class="form-control"
placeholder="Attn">
                <input runat="server" id="ToEmail" type="text" class="form-control"
placeholder="Email">
                <input runat="server" id="ToPhone" type="text" class="form-control"
placeholder="Phone">

            </div>
        </div>
    </div>
</div>
<div class="col-md-3">

```



```

<div class="awp_box rounded bg-sg-title shadow">
  <div class="awp_box_title bg-sg-title">
    <span class="fg-white shadow-metro-black bold"><span class="mif-widgets
mif-2x"></span>Shipping Options</span></div>
    <div class="awp_box_content bg-sg-box" style="padding-left:0px !important;
padding-right:0px !important">
      <div style="padding-left:3px !important; padding-right:3px!important;">
        <asp:DropDownList
OnSelectedIndexChanged="ShippingMethodDropDownList_SelectedIndexChanged"
AutoPostBack="true" Width="100%" ClientIDMode="Static" ID='ShippingMethodDropDownList'
AppendDataBoundItems="true" runat="server" CssClass="input-control text">
          <asp:ListItem Text="No Label" Value="00-NoLabel" />
          <asp:ListItem Text="01-Next Day" Value="01-NextDayAir" />
          <asp:ListItem Text="02-2nd Day" Value="02-2ndDayAir" />
          <asp:ListItem Text="03-Ground" Value="03-Ground" />
          <asp:ListItem Text="12-3 Day Select" Value="12-3DaySelect" />
          <asp:ListItem Text="13-Next Day Air Saver" Value="13-
NextDayAirSaver" />
          <asp:ListItem Text="14-Next Day Air Saver Early" Value="14-
NextDayAirEarly" />
          <asp:ListItem Text="59-2nd Day Early AM" Value="59-2ndDayAirA.M." />

          <asp:ListItem Text="65-UPS Saver" Value="65-UPSSaver" />
        </asp:DropDownList>
      </div>
      <div class="row">
        <div class="col-md-12" style="padding-left:3px !important; padding-
right:3px!important;">
          <span class="awp-save-btn fg-white shadow">
            <asp:LinkButton runat="server" ID="RefreshCost" CssClass="fg-white"
ClientIDMode="Static" OnClick="RefreshCost_Click">
              <i title="Save" style="font-size:1em; padding:3px" class="mif-loop2 av-
hand-cursor bg-green rounded shadow"></i></asp:LinkButton>
            </span>
            <div class="awp_box rounded bg-sg-title shadow">
              <div class="awp_box_title bg-sg-title">
                <span class="fg-white shadow-metro-black bold"><span class="mif-
dollars mif-2x"></span>Estimated Cost</span></div>
                <div class="awp_box_content bg-sg-box">
                  <input runat="server" id="EstimatedCost" type="text" class="form-
control" placeholder="$0.00">

```

```

        </div>
    </div>
</div>
<div class="col-md-12" style="padding-left:3px !important; padding-
right:3px!important;">
    <span class="awp-save-btn fg-white shadow">
        <asp:LinkButton runat="server" ID="RefreshArrival" CssClass="fg-white"
ClientIDMode="Static" OnClick="RefreshArrival_Click">
            <i title="Save" style="font-size:1em; padding:3px" class="mif-loop2 av-
hand-cursor bg-green rounded shadow"></i></asp:LinkButton>
        </span>
        <div class="awp_box rounded bg-sg-title shadow">
            <div class="awp_box_title bg-sg-title">
                <span class="fg-white shadow-metro-black bold"><span class="mif-
calendar mif-2x"></span>Estimated Arrival</span></div>
                <div class="awp_box_content bg-sg-box">
                    <input runat="server" id="ArrivalDate" type="text" class="form-control"
placeholder="Arrival Date">
                    <input runat="server" id="ArrivalTime" type="text" class="form-control"
placeholder="Arrival Time">
                </div>
            </div>
        </div>
    </div>

    </div>
</div>
<div class="col-md-3">

    <div class="awp_box rounded bg-sg-title shadow">
        <div class="awp_box_title bg-sg-title">
            <span class="fg-white shadow-metro-black bold"><span class="mif-dropbox
mif-2x"></span>Package Options</span></div>
            <div class="awp_box_content bg-sg-box">
                <input runat="server" id="PkgWeight" type="text" class="form-control"
placeholder="Weight">
                <input runat="server" id="PkgHeight" type="text" class="form-control"
placeholder="Height">
                <input runat="server" id="PkgWidth" type="text" class="form-control"
placeholder="Width">
                <input runat="server" id="PkgLength" type="text" class="form-control"

```

```

placeholder="Length">
        <input runat="server" id="PkgReference" type="text" class="form-control"
placeholder="Reference">
        <input runat="server" id="PkgReference2" type="text" class="form-control"
placeholder="2nd Reference">
        <input runat="server" id="PkgTracking" type="text" class="form-control"
disabled="disabled" placeholder="Tracking Number">
        </div>
    </div>
</div>

```

```

</div>

</asp:View>
<asp:View ID="PackingSlipView" ClientIDMode="Static" runat="server">
<script type="text/javascript">
    $(document).ready(function () {

```

```

// create function to calculate ideal height values
function CalcDiv() {
    try {
        var newHeight = 0;
        var myViewHeight = jQuery(".main-content").height();
        newHeight = myViewHeight - 180;
        $("#SPD").height(newHeight);
        var aa = $("#SPD").height();
        var a = $("#SPD").width();
        var p = $("#PF");
        //$("#PF").width(aa - 30);
        var c = $("#MainContent_PF").width(a);
        var d = $("#MainContent_PF").height(aa);
        var e = $("#MainContent_PF");
        $("#PF").height(a-30);
    } catch (er)
    { var eee = er; }
} // end calcHeights function

```

```

// run on page load

```

```
CalcDiv();
```

```
// run on window resize event
```

```
$(window).resize(function () {
```

```
    CalcDiv();
```

```
});
```

```
});
```

```
</script>
```

```
<div class="row">
```

```
<div class="col-md-12" style="margin-bottom:0px!important;">
```

```
<!-- <span class="awp-save-btn bg-green fg-white shadow mif-ani-flash "
onclick="printpackingslip()"><i title="Print" style="font-size:2em; vertical-align:top" class="mif-
printer av-hand-cursor fg-white shadow-metro-black"></i></span>-->
```

```
<div class="awp_box rounded bg-sg-title shadow">
```

```
<div class="awp_box_title bg-sg-title">
```

```
<span class="fg-white shadow-metro-black">Packing Slip + Tracking: <asp:Label
ID="TrackingNumberLabel" runat="server" Text=""></asp:Label></span>
```

```
</div>
```

```
<div class="awp_box_content bg-sg-box">
```

```
<div id="SPD" style="text-align:center;">
```

```
<div id="PackingSlipHidden" style="display:none"><asp:Literal
ID="PackingSlipLink" runat="server"></asp:Literal> </div>
```

```
<iframe style="height:80%; position:relative; z-index:-1;" id="PF"
runat="server" src="#" ></iframe>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</asp:View>
```

```
<asp:View ID="ShippingLabelView" ClientIDMode="Static" runat="server">
```

```
<div class="row">
```

```
<div class="col-md-12">
```

```
<span class="awp-save-btn bg-green fg-white shadow mif-ani-flash "
onclick="printimage()"><i title="Print" style="font-size:2em; vertical-align:top" class="mif-printer
av-hand-cursor fg-white shadow-metro-black"></i></span>
```

```
<div class="awp_box rounded bg-sg-title shadow">
```

```
<div class="awp_box_title bg-sg-title">
```

```

        <span class="fg-white shadow-metro-black"></span>
    </div>
    <div class="awp_box_content bg-sg-box">
        <div id="UpsDiv" style="min-height:100px;height:600px;">
            <div id="ups_label_only">
                <div id="imgcontainer" runat="server"></div>
            </div>
        </div>
    </div>
</div>
</div>
</div>
</div>

</asp:View>
</asp:MultiView>
<div id="FinalizeConfirmDialog" class="awp-outer-dialog" style="display:none" >
    <div class=" awp-inner-dialog" style="opacity:initial !important">

        <span class="awp-dialog-close-btn bg-red fg-white shadow "
onclick="HideDiv('FinalizeConfirmDialog')"><i title="Close" style="vertical-align:top" class="mif-
cross av-hand-cursor fg-white shadow-metro-black"></i></span>
        <div class="awp_box rounded bg-sg-title shadow" style="left:50% !important;
top:30%">
            <div class="awp_box_title bg-sg-title">
                <span class="fg-white shadow-metro-black">Confirm Label Creation</span>
            </div>
            <div class="awp_box_content bg-sg-box">
                <asp:Button CssClass="control-button" ID="but_OK" runat="server"
Text="Create Labels" OnClientClick="ShowLoader()" OnClick="FinalizeBtn_Click" /><br />
            </div>
        </div>

    </div>
</div>

</asp:Content>

```

```

using Helpers;
using Microsoft.AspNet.Identity;
using Microsoft.AspNet.Identity.Owin;
using ShippingAPI;
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Web;
using System.Web.Http;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using static Notification.NotificationSystem;

namespace Web_App_Master.Account
{
    public partial class Checkout1 : System.Web.UI.Page
    {
        public void SaveToUserPersistantLog()
        {
            try
            {
                var ud = Session["PersistingUserData"] as Data.UserData;
                string pdf = Session["CombinedPdf"] as string;
                Data.Attachment a = new Data.Attachment();
                a.Email = ud.Email;
                a.Name = ud.Name;
                a.Files.Add(new Data.FileReference(Session["Ordernumber"] as string, pdf));
                ud.Attachments.Add(a);
                ud.Log.Add(new Data.LogEntry("Checked Out Order #" + Session["Ordernumber"] as string, Session["Ordernumber"] as string));

                SettingsDBData db = new SettingsDBData();
                db.Appname = ud.Guid;
                db.XmlData = ud.SerializeToXmlString(ud);

                Push.Setting(db);
            }
            catch { }
        }
    }
}

```

```

public void NotifyCheckoutEmail(List<Asset> assets)
{
    if(Global.Library.Settings.TESTMODE==false)
    {
        try
        {
            EmailHelper.SendCheckOutNoticeAsync(assets,
Global.Library.Settings.CheckOutMessage);
        }
        catch { ShowError("Problem Sending Emails"); }

    }

}

public string Customer = null;
public string Shipper = null;
public string Engineer = null;
public string Ordernumber = null;

private void DataBindShipping(string customer, string shipper, string engineer, string
ordernumber)
{
    try
    {
        Customer cust = (from temp in Global.Library.Settings.Customers where
temp.CompanyName == customer select temp).ToList().FirstOrDefault();
        if (cust != null)
        {
            ToCompany.Value = cust.CompanyName;
            ToAddr.Value = cust.Address;
            ToAddr2.Value = cust.Address2;
            ToCty.Value = cust.City;
            ToState.Value = cust.State;
            ToPostal.Value = cust.Postal;
            ToCountry.Value = cust.Country;
            ToName.Value = cust.Attn;
            ToEmail.Value = cust.Email;
            ToPhone.Value = cust.Phone;
        }
    }
}

```

```

        Session["CheckOutCustomer"] = cust;
    }
    catch { }
    var starrag = "Starrag USA Inc.";
    Customer defaultshipper = (from temp in Global.Library.Settings.Customers where
temp.CompanyName == starrag select temp).ToList().FirstOrDefault();
    FillDefaultShipper(defaultshipper);
}
protected bool Finalized = false;
protected void FinalizeBtn_Click(object sender, EventArgs e)
{

    if (ShippingMethodDropDownList.Text=="")
    {
        ShowError("Please Select A Shipping Option");
        return;
    }
    var checkoutdata = Session["CheckOutData"] as CheckOutData;
    checkoutdata = UpdateCheckOutData(checkoutdata);
    if (checkoutdata.CheckOutItems == null) return;
    if (checkoutdata.CheckOutItems.Count > 0)
    {
        if (ShippingMethodDropDownList.SelectedValue == "00-NoLabel")
        {
            Finalized = true;
            PackingSlipViewBtn.Enabled = true;

            CreatePackingSlip();

            //UpdateCost();
            // UpdateArrival();
            // UpdateUpsLabel();

            PackingSlipOnly();

            //notice
            try
            {
                EmailNotice notice = new EmailNotice();

                notice.Scheduled = DateTime.Now.AddDays(30);
                notice.NoticeType = NoticeType.Checkout;
            }
            catch { }
        }
    }
}

```



```

notice.NoticeAction = Global.CheckoutAction;
foreach (var asset in checkoutdata.CheckOutItems)
{
    notice.Assets.Add(asset.AssetNumber);
}
notice.NoticeControlNumber = checkoutdata.CheckOutItems[0].OrderNumber;
notice.Body = Global.Library.Settings.CheckOutMessage;
notice.Subject = "Asset Return Reminder";
var engineer = (from d in Global.Library.Settings.ServiceEngineers where d.Name
== Session["Engineer"] as string select d).FirstOrDefault();
var statics = (from d in Global.Library.Settings.StaticEmails select d).ToList();
if (engineer == null)
{
    engineer = new EmailAddress();
}
notice.Emails.Add(engineer);
notice.Emails.AddRange(statics);
notice.EmailAddress = engineer;
Global.NoticeSystem.Add(notice);
Push.NotificationSystem();
}
catch { ShowError("Problem Adding Timed Notice"); }

SaveToUserPersistantLog();

FinalizeAssets(Session["Checkout"] as List<Asset>);

Session["Checkout"] = new List<Asset>();

but_OK.Enabled = false;

FinalizeHolder.Visible = false;

LeavePlaceholder.Visible = true;

CheckoutView.ActiveViewIndex = 1; // goto report
}
else
{
    if (checkoutdata.Package.Weight == null) { ShowError("No Package Weight."); return;
}

if (checkoutdata.From.AddressLine[0] != "" && checkoutdata.To.AddressLine[0] != ""

```

```

&& checkoutdata.Shipper.AddressLine[0] != "" && checkoutdata.Package.Weight != "")
    {
        var ud = Session["SessionUserData"] as Data.UserData;
        ud.Log.Add(new Data.LogEntry("Order#" +
checkoutdata.CheckOutItems.FirstOrDefault().OrderNumber, Session["CombinedPdf"] as string));
        Finalized = true;
        PackingSlipViewBtn.Enabled = true;

        CreatePackingSlip();

        UpdateCost();
        UpdateArrival();
        UpdateUpsLabel();

        // PackingSlipOnly();

        //Shake(ReportLink);
        try
        {
            EmailNotice notice = new EmailNotice();

            notice.Scheduled = DateTime.Now.AddDays(30);
            notice.NoticeType = NoticeType.Checkout;
            notice.NoticeAction = Global.CheckoutAction;
            foreach (var asset in checkoutdata.CheckOutItems)
            {
                notice.Assets.Add(asset.AssetNumber);
            }
            notice.NoticeControlNumber = checkoutdata.CheckOutItems[0].OrderNumber;
            notice.Body = Global.Library.Settings.CheckOutMessage;
            notice.Subject = "Asset Return Reminder";
            var engineer = (from d in Global.Library.Settings.ServiceEngineers where
d.Name == Session["Engineer"] as string select d).FirstOrDefault();
            var statics = (from d in Global.Library.Settings.StaticEmails select d).ToList();
            if (engineer == null)
            {
                engineer = new EmailAddress();
            }
            notice.Emails.Add(engineer);
            notice.Emails.AddRange(statics);
            notice.EmailAddress = engineer;
            Global.NoticeSystem.Add(notice);

```

```

        Push.NotificationSystem();
    }
    catch { ShowError("Problem Adding Timed Notice"); }
    SaveToUserPersistantLog();
    FinalizeAssets(Session["Checkout"] as List<Asset>);
    Session["Checkout"] = new List<Asset>();

    but_OK.Enabled = false;
    FinalizeHolder.Visible = false;
    LeavePlaceHolder.Visible = true;

    CheckoutView.ActiveViewIndex = 1; // goto report
}
else
{
    ShowError("Please fill in form completely.");
}
}
else
{
    ShowError("There are no items in Check Out");
}
try
{
    if(Session["IsPending"] as string == "true")
    {
        AssetController.RemoveTransaction(Session["TransactionID"] as string);
    }
}
catch (Exception) { }
}

private void FinalizeAssets(List<Asset> assets)
{
    List<Asset> emailist = new List<Asset>();
    foreach (var A in assets)
    {
        A.ShipTo = Session["Customer"] as string;
        A.ServiceEngineer = Session["Engineer"] as string;
        try

```

```

{
    var t = Session["Ordernumber"] as string;
    if (t!=null)
    {
        A.OrderNumber = t;
    }

}
catch {
    ShowError("Could Not Add Order Number To Asset: " + A.AssetNumber);
}
A.DateShipped = DateTime.Now;
A.PackingSlip = Session["LastPackingSlip"] as string;
A.PersonShipping = Session["Shipper"] as string;
A.UpsLabel= Session["ShippingLabelPdf"] as string;
A.IsOut = true;
Asset rem=null;
foreach (var a in Global.Library.Assets)
{
    if (a.AssetNumber==A.AssetNumber)
    {
        rem = a;
    }
}
if (rem != null)
{
    Global.Library.Assets.Remove(rem);
    Global.Library.Assets.Add(A);
}
emaillist.Add(A);

//UPDATE ASSET IN SQL
//AssetController.UpdateAsset(A.Clone() as Asset);
Push.Asset(A);
}
if (!Global.Library.Settings.TESTMODE)
{
    Session["Customer"] = null;
    Session["Shipper"] = null;
    Session["Engineer"] = null;
    Session["Ordernumber"] = null;
}

```

```
    }  
    NotifyCheckoutEmail(emaillist);  
}
```

```
private void FillDefaultShipper(Customer cust)  
{  
    SprCompany.Value = cust.CompanyName;  
    SprAddr.Value = cust.Address;  
    SprAddr2.Value = cust.Address2;  
    SprCty.Value = cust.City;  
    SprState.Value = cust.State;  
    SprPostal.Value = cust.Postal;  
    SprCountry.Value = cust.Country;  
    SprName.Value = cust.Attn;  
    SprEmail.Value = cust.Email;  
    SprPhone.Value = cust.Phone;  
}  
public PendingTransaction Pending = null;
```

```
private void DataBindPacking()  
{
```

```
}  
private void DataBindShippingLabel()  
{  
  
}
```

```
protected void Page_Load(object sender, EventArgs e)  
{  
    Session["IsPending"] = "false";  
    try  
    {  
        var TransactionID = Request.QueryString["tid"];  
        Session["TransactionID"] = TransactionID;  
        if (TransactionID != null)  
        {  
            Pending = Web_App_Master.Pull.Transaction(TransactionID);  
            Session["Pending"] = Pending;  
            Session["IsPending"] = "true";  
        }  
    }  
}
```

```

catch (Exception) { }

if (!Page.User.Identity.IsAuthenticated)
{
    Response.Redirect("/Account/Login");
}
var checkoutdata = Session["CheckOutData"] as CheckOutData;
if (checkoutdata == null) checkoutdata = new CheckOutData();
var CheckoutItems = Session["Checkout"] as List<Asset>;
checkoutdata.CheckOutItems = CheckoutItems;
if (!IsPostBack)
{

    if (!Global.Library.Settings.TESTMODE)
    {
        //ShippingLabelBtn.Enabled = false;
        PackingSlipViewBtn.Enabled = false;
    }
    but_OK.Enabled = true;
    Customer = Session["Customer"] as string;
    Shipper = Session["Shipper"] as string;
    Engineer = Session["Engineer"] as string;
    Ordernumber = Session["Ordernumber"] as string;
    DataBindShipping(Customer,Shipper,Engineer,Ordernumber);

    var preferShip = Session["PreferedShipMethod"] as string;
    if (preferShip == null)
    {
        ShippingMethodDropDownList.Text= "No Label";
        //ShippingMethodDropDownList.SelectedIndex = 3;
        Session["PreferedShipMethod"] = "NoLabel";
    }
    else
    {

        ShippingMethodDropDownList.Text = preferShip;

    }
    decimal TotalWeight = 0;
    if (CheckoutItems != null)
    {
        foreach (var asset in CheckoutItems)

```

```

        {
            TotalWeight += asset.weight;
        }
        PkgWeight.Value = TotalWeight.ToString();
        checkoutdata.Package.Weight = TotalWeight.ToString();
    }

}
MessagePlaceholder.Visible = false;

```

```

UpdateCheckOutData(checkoutdata );
}

```

```

private CheckOutData UpdateCheckOutData(CheckOutData checkoutdata)
{
    //Code
    checkoutdata.Code = checkoutdata.ParseUpsCodeString(Session["PreferredShipMethod"]
as string);
    if (checkoutdata.Package.Weight == null )
    {
        checkoutdata.Package.Weight = "1";
        PkgWeight.Value = "1";
    }
    //pkg
    decimal TotalWeight = 0;
    var CheckoutItems = Session["Checkout"] as List<Asset>;
    if (CheckoutItems != null)
    {

    }
    //package
    string userInputweight = PkgWeight.Value;
    try
    {
        TotalWeight = Convert.ToDecimal(userinputweight);
        if (TotalWeight==0)
        {
            TotalWeight = 1;
            PkgWeight.Value = "1";
        }
    }
}

```

```

    }
    catch { }

    try
    {
        checkoutdata.Package.Weight = TotalWeight.ToString();
        checkoutdata.Package.H = Convert.ToInt32(string.IsNullOrEmpty(PkgHeight.Value) ?
"0" : PkgHeight.Value);
        checkoutdata.Package.L = Convert.ToInt32(string.IsNullOrEmpty(PkgLength.Value) ?
"0" : PkgHeight.Value);
        checkoutdata.Package.W = Convert.ToInt32(string.IsNullOrEmpty(PkgWidth.Value) ?
"0" : PkgHeight.Value);
        checkoutdata.Package.PackType = UPS_PackagingType.CustomerSupplied;
        checkoutdata.Package.reference = PkgReference.Value;
        checkoutdata.Package.reference2 = PkgReference2.Value;
    }
    catch { }

```

//From

```

checkoutdata.From.AddressLine = new string[] { SprAddr.Value, SprAddr2.Value };
checkoutdata.From.Name = SprCompany.Value;
checkoutdata.From.AttentionName = SprName.Value;
checkoutdata.From.City = SprCty.Value;
checkoutdata.From.State = SprState.Value;
checkoutdata.From.Postal = SprPostal.Value;
checkoutdata.From.Country = SprCountry.Value;
checkoutdata.From.Phone = "18595345201";

```

//To

```

checkoutdata.To.AddressLine = new string[] { ToAddr.Value, ToAddr2.Value };
checkoutdata.To.Name = ToCompany.Value;
checkoutdata.To.AttentionName = ToName.Value;
checkoutdata.To.City = ToCty.Value;
checkoutdata.To.State = ToState.Value;
checkoutdata.To.Postal = ToPostal.Value;
checkoutdata.To.Country = ToCountry.Value;
checkoutdata.To.Phone = "18595345201";

```

```

checkoutdata.Shipper = checkoutdata.From;

```

//save

```

Session["CheckOutData"] = checkoutdata;

```



```
    return checkoutdata;
}
```

```
protected void ShippingViewBtn_Click(object sender, EventArgs e)
{
    CheckoutView.ActiveViewIndex = 0;
    QuietLinks();
}
```

```
protected void PackingSlipViewBtn_Click(object sender, EventArgs e)
{
    CheckoutView.ActiveViewIndex = 1;
    var filename = Session["CombinedPdf"] as string;
    if (filename != null)
        PF.Src = filename;
    QuietLinks();
}
```

```
protected void ShippingLabelBtn_Click(object sender, EventArgs e)
{
    CheckoutView.ActiveViewIndex = 2;
    DataBindShippingLabel();
    QuietLinks();
}
```

```
protected void SaveShipperBtn_Click(object sender, EventArgs e)
{
}
}
```

```
protected void CombineShipAndPack()
{
    try
    {
        var filename = TrackingNumberLabel.Text + ".pdf";
        Session["CombinedPdf"] = PF.Src = "/Account/CheckOutPdf/" + filename; ;
        var dest = Server.MapPath("/Account/CheckOutPdf/" + filename);
        var pack = Session["LastPackingSlip"] as string;
        var ship = Session["ShippingLabelPdf"] as string;
        string[] combine = new string[2] { Server.MapPath(pack), Server.MapPath(ship) };
    }
}
```

```

        var result = Pdf.Merge(combine.ToList(), dest);
        if (result)
        {
            PF.Src = "/Account/CheckOutPdf/" + filename;

        }
    }
    catch { }

}

protected void PackingSlipOnly()
{
    try
    {
        var filename = Guid.NewGuid().ToString() + ".pdf";
        Session["CombinedPdf"] = PF.Src = "/Account/CheckOutPdf/" + filename; ;
        var dest = Server.MapPath("/Account/CheckOutPdf/" + filename);
        var pack = Session["LastPackingSlip"] as string;
        File.Copy(Server.MapPath(pack), dest);
        // string[] combine = new string[2] { Server.MapPath(pack), Server.MapPath(ship) };

        // var result = Pdf.Merge(combine.ToList(), dest);
        //if (result)
        // {
            PF.Src = "/Account/CheckOutPdf/" + filename;

        // }
    }
    catch { }

}

protected void ShowError(string error)
{
    try
    {
        MessagePlaceholder.Visible = true;
        ErrorMessage.Text = error;
    }
    catch { }

}

protected bool ValidateUpsAcct(UPSaccount acct)

```

```

{
    if (acct.P==null || acct.A == null || acct.I == null || acct.N == null )
    { return false; }
    if (acct.P == "" || acct.A == "" || acct.I == "" || acct.N == "")
    { return false; }
    return true;
}

private string GetInnerUpsAcctError(UPSAccount acct)
{
    string ret = "";
    if (acct.P == null)
    { ret+= "<No Password>"; }
    if (acct.A == null)
    { ret += "<No Account Number>"; }
    if (acct.I == null)
    { ret += "<No UserID>"; }
    if (acct.N == null)
    { ret += "<No Shipper Number>"; }
    if (acct.P == "")
    { ret += "<No Password>"; }
    if (acct.A == "")
    { ret += "<No Account Number>"; }
    if (acct.I == "")
    { ret += "<No UserID>"; }
    if (acct.N == "")
    { ret += "<No Shipper Number>"; }
    return ret;
}

protected void UpdateCost()
{
    try
    {
        var checkoutdata = Session["CheckOutData"] as CheckOutData;
        checkoutdata = UpdateCheckOutData(checkoutdata);

        if (checkoutdata != null)
        {
            if (checkoutdata.Package.Weight == null) { ShowError("No Package Weight."); return; }

            if (checkoutdata.From.AddressLine[0] != "" && checkoutdata.To.AddressLine[0] != ""
&& checkoutdata.Shipper.AddressLine[0] != "" && checkoutdata.Package.Weight != "")
            {

```

```

        if (Global.Library.Settings.TESTMODE)
        {
            checkoutdata.Pickup = DateTime.Now.AddHours(-8).AddDays(1);
        }
        //Check UPS Account
        if (!ValidateUpsAcct(Global.Library.Settings.UpsAccount))
        { ShowError("Please check your Shipper Account Settings in Admin Panel:
"+GetInnerUpsAcctError(Global.Library.Settings.UpsAccount)); return; }
        UPS ups = new UPS(Global.Library.Settings.UpsAccount,
Global.Library.Settings.TESTMODE);
        ups.OnException += Ups_OnException;
        ups.OnRateReturn += Ups_OnRateReturn;
        ups.SoapExceptionListener += Ups_SoapExceptionListener;
        var result = ups.GetRateAsync(checkoutdata.From, checkoutdata.To,
checkoutdata.From, checkoutdata.Code, checkoutdata.Package);
        if (result == null)
        {
            //error
        }
    }
    else
    {
        ShowError("Please fill in form completely.");
    }
}
}
catch
{
}
}
}

```

```

protected void CreatePackingSlip()
{
    var cd = Session["CheckOutData"] as CheckOutData;
    //get filename
    string n = DateTime.Now.ToString().Replace("/", "-");
    n = n.Replace(":", "-");
    string info = n + " " + (Session["Ordernumber"] as string);
    info = info.Replace("\\", "").Replace(":", "").Replace("*", "").Replace("?", "").Replace("\",
""").Replace("<", "").Replace(">", "").Replace("|", "").Replace(" ", "_");
    string path = "/Account/PackingLists/" + info + ".pdf";
}

```

```

PackingSlipLink.Text = path;
Session["LastPackingSlip"] = path;
path = Server.MapPath(path);

//To
string to = cd.To.Name + Environment.NewLine;
foreach(var line in cd.To.AddressLine)
{
    to += line + Environment.NewLine;
}
to += cd.To.City + Environment.NewLine;
to += cd.To.State + ", " + cd.To.Postal + Environment.NewLine;
to += cd.To.Country + Environment.NewLine;
//From
string from = cd.From.Name + Environment.NewLine;
foreach (var line in cd.From.AddressLine)
{
    from += line + Environment.NewLine;
}
from += cd.From.City + Environment.NewLine;
from += cd.From.State + ", " + cd.From.Postal + Environment.NewLine;
from += cd.From.Country + Environment.NewLine;
//Define Keys

var headerKeys = new Dictionary<string, string>() {
{"ShipDate", cd.Pickup.ToShortDateString()},
{"Tracking", TrackingNumberLabel.Text},
{"Signature", Session["Shipper"] as string},
{"Attn", cd.To.AttentionName},
{"Po", ""},
{"To", to},
{"From", from},
{"Weight", "Weight:" + cd.Package.Weight},
{"Ordernumber", "Order #:" + Session["Ordernumber"] as string} };

var keys = new Dictionary<string, string>();
headerKeys.ToList().ForEach(x => keys.Add(x.Key, x.Value));
int i = 1;
foreach (var asset in cd.CheckOutItems)
{
    if (i>13)
    {

```

```

        //create templated pdf packingslip with default stamping action
        Pdf.PackingSlip p = new Pdf.PackingSlip(new
Templates.StarragPackingSlipTemplate());
        p.CreateAndFill(path, keys, p.StampAction);
        //fill new page with new assets
        keys = new Dictionary<string, string>();
        headerKeys.ToList().ForEach(x => keys.Add(x.Key, x.Value));
        i = 1;
        path = path.Replace(".pdf", "")+i.ToString()+".pdf";
    }
    keys.Add("Desc" + i.ToString(), asset.AssetName);
    keys.Add("PartNo" + i.ToString(), asset.AssetName);
    keys.Add("Qty" + i.ToString(), "1");
    ++i;
}

//create templated pdf packingslip with default stamping action
Pdf.PackingSlip ps = new Pdf.PackingSlip(new Templates.StarragPackingSlipTemplate());
var result = ps.CreateAndFill(path, keys, ps.StampAction);
var bbbb = false;

}
protected void UpdateArrival()
{
    try
    {
        var checkoutdata = Session["CheckOutData"] as CheckOutData;
        checkoutdata = UpdateCheckOutData(checkoutdata);

        if (checkoutdata != null)
        {
            if (checkoutdata.Package.Weight == null) { ShowError("No Package Weight."); return;
}

            if (checkoutdata.From.AddressLine[0]!=""&& checkoutdata.To.AddressLine[0] != "" &&
checkoutdata.Shipper.AddressLine[0] != "" && checkoutdata.Package.Weight != "")
            {
                if (!ValidateUpsAcct(Global.Library.Settings.UpsAccount))
                { ShowError("Please check your Shipper Account Settings in Admin Panel:" +
GetInnerUpsAcctError(Global.Library.Settings.UpsAccount)); return; }
                UPS ups = new UPS(Global.Library.Settings.UpsAccount,
Global.Library.Settings.TESTMODE);
                ups.OnException += Ups_OnException;

```

```

        ups.OnRateReturn += Ups_OnRateReturn;
        ups.OnTransitReturn += Ups_OnTransitReturn;
        var result = ups.GetTransitAsync(checkoutdata.From, checkoutdata.To,
checkoutdata.Pickup, checkoutdata.Package.Weight);
        if (result == null)
        {
            //error
        }
    }
    else
    {
        ShowError("Please fill in form completely.");
    }
}
}
catch
{
}
}
protected void UpdateUpsLabel()
{
    try
    {
        var checkoutdata = Session["CheckOutData"] as CheckOutData;
        checkoutdata = UpdateCheckOutData(checkoutdata);

        if (checkoutdata != null)
        {
            if (checkoutdata.Package.Weight == null) { ShowError("No Package Weight."); return;
}

            if (checkoutdata.From.AddressLine.Count() != 0 &&
checkoutdata.To.AddressLine.Count() != 0 && checkoutdata.Shipper.AddressLine.Count() != 0 &&
checkoutdata.Package.Weight != "")
            {
                if (!ValidateUpsAcct(Global.Library.Settings.UpsAccount))
                { ShowError("Please check your Shipper Account Settings in Admin Panel:" +
GetInnerUpsAcctError(Global.Library.Settings.UpsAccount)); return; }
                UPS ups = new UPS(Global.Library.Settings.UpsAccount,
Global.Library.Settings.TESTMODE);
                ups.OnException += Ups_OnException;
                ups.OnRateReturn += Ups_OnRateReturn;

```

```

        ups.OnShipReturn += Ups_OnShipReturn;
        var result = ups.GetShipAsync(checkoutdata.From, checkoutdata.To,
checkoutdata.From, checkoutdata.Code, checkoutdata.Package,
checkoutdata.Package.reference, checkoutdata.Package.reference2);
        if (result == null)
        {
            ShowError("Problem Creating Label");
        }
    }
    else
    {
        ShowError("Please fill in form completely.");
    }
}
}
catch
{
    ShowError("Problem Getting Shipping Label");
}
}
protected void QuietLinks()
{
    Quiet(ShipLink);
    //Quiet(LabelLink);
    Quiet(ReportLink);
}
protected void Shake(HtmlGenericControl control)
{
    var a = control.Attributes["class"];
    a += " mif-ani-flash";
    control.Attributes["class"] = a ;
}
protected void Quiet(HtmlGenericControl control)
{
    var a = control.Attributes["class"];
    a = a.Replace("mif-ani-flash", "").Replace("mif-ani-slow", "");
    control.Attributes["class"] = a;
}

protected void RefreshCost_Click(object sender, EventArgs e)
{
    UpdateCost();
}

```



```
}
```

```
private void Ups_OnTransitReturn(object sender, UPS.ReturnTransitEvent e)  
{
```

```
    Transit.RESPONSE r = e.Response;
```

```
    var code = ShippingMethodDropDownList.Text;
```

```
    string sum = "*****\r\n";
```

```
    try
```

```
    {
```

```
        code = (code.Split('-'))[1];
```

```
    }
```

```
    catch { }
```

```
    foreach (var unit in r.Summary)
```

```
    {
```

```
        var cd = unit.Service.Description.Replace("UPS ", "").Replace(" ", "");
```

```
        if (cd == code)
```

```
        {
```

```
            try
```

```
            {
```

```
                System.Globalization.CultureInfo provider = new
```

```
System.Globalization.CultureInfo("en-US");
```

```
                var d = DateTime.ParseExact(unit.EstimatedArrival.Arrival.Date, "yyyyMMdd",
```

```
provider);
```

```
                ArrivalDate.Value = d.ToShortDateString();
```

```
                string dn = "PM";
```

```
                int a = Convert.ToInt32(unit.EstimatedArrival.Arrival.Time);
```

```
                if (a <= 119999)
```

```
                    dn = "AM";
```

```
                else
```

```
                    a = a - 120000;
```

```
                var t = a.ToString();
```

```
                t = t.Insert(2, ":");
```

```
                t = t.Insert(5, ":");
```

```
                ArrivalTime.Value = t + " " + dn;
```

```
            }
```

```
        catch { }
```

```
    return;
```

```

    }
    sum += unit.Service.Description +
        "\r\nBusiness Days: "
        + unit.EstimatedArrival.BusinessDaysInTransit +
        "\r\nArrival Date: " + unit.EstimatedArrival.Arrival.Date + " @ " +
unit.EstimatedArrival.Arrival.Time + "\r\n*****\r\n";
    }

}
private void Ups_OnShipReturn(object sender, UPS.ReturnShipEvent e)
{
    try
    {
        var img = e.Response.Label;
        using (MemoryStream ms = new MemoryStream())
        {
            //img.RotateFlip(System.Drawing.RotateFlipType.Rotate90FlipNone);
            img.Save(ms, System.Drawing.Imaging.ImageFormat.Png);
            Session["LastShippingLabel"] = "/Account/Labels/" + e.Response.TrackingNumber +
".gif";

            var filename = Server.MapPath("/Account/Labels/" + e.Response.TrackingNumber +
".gif");

            PkgTracking.Value = TrackingNumberLabel.Text = e.Response.TrackingNumber;
            Session["LastTrackingNumber"] = e.Response.TrackingNumber;

            var shortname = "/Account/Labels/" + e.Response.TrackingNumber + ".gif";
            try
            {
                img.Save(filename, System.Drawing.Imaging.ImageFormat.Png);
            }
            catch { }
            try
            {
                Session["ShippingLabelPdf"] = "/Account/Labels/" + e.Response.TrackingNumber +
".pdf";

                var dest = Server.MapPath("/Account/Labels/" + e.Response.TrackingNumber +
".pdf");

                Pdf.ImageToPdf(img, dest);
                img.Save(filename, System.Drawing.Imaging.ImageFormat.Png);
            }
            catch { }
            UpsLabelImgBox.Attributes["height"] = "100%";

```

```

        try
        {

            imgcontainer.Style["height"] = "100%";
        }
        catch { }
        UpsLabellmgBox.Alt = DateTime.Now.ToString();
        UpsLabellmgBox.Src = shortname;
        Session["ShippingLabelFileName"] = shortname;

        //create pdf

        if (Finalized)
        {
            CombineShipAndPack();
        }
    }
}
catch { }

}
private void Ups_OnRateReturn(object sender, UPS.ReturnRateEvent e)
{
    try
    {
        EstimatedCost.Value = e.Response.Negotiated;
    }
    catch { }
}
private void Ups_OnException(object sender, UPS.ExceptionOccured e)
{
    try
    {
        ShowError("A problem has occurred, please check that all forms are filled in correctly
and try again");
    }
    catch { }
}
private void Ups_SoapExceptionListener(object sender, UPS.SoapExceptionOccured e)
{
    ShowError(e.Exception.Detail.InnerText + e.Exception.Detail.InnerXml);
}

```

```

protected void RefreshArrival_Click(object sender, EventArgs e)
{
    try
    {
        UpdateArrival();
    }
    catch {

        ShowError("Problem Updating Arrival");
    }
}

protected void ShippingMethodDropDownList_SelectedIndexChanged(object sender,
EventArgs e)
{
    Session["PreferredShipMethod"] = ShippingMethodDropDownList.SelectedValue;
    if (ShippingMethodDropDownList.SelectedValue != "00-NoLabel")
    {
        UpdateCost();
        UpdateArrival();

    }

}

protected void GetUpsLabel_Click(object sender, EventArgs e)
{
    UpdateUpsLabel();
}
}
}

```

```
<%@ Page Title="Account Confirmation" Language="C#" MasterPageFile="~/Site.Master"
AutoEventWireup="true" CodeBehind="Confirm.aspx.cs"
Inherits="Web_App_Master.Account.Confirm" Async="true" %>
```

```
<asp:Content runat="server" ID="BodyContent" ContentPlaceHolderID="MainContent">
```

```
    <div class="bg-shaded"></div>
```

```
    <h2><%= Title %>.</h2>
```

```
    <div class="row">
```

```
        <div class="col-md-offset-4 col-md-4 fg-white shadow-metro-black">
```

```
            <asp:Placeholder runat="server" ID="successPanel" ViewStateMode="Disabled"
Visible="true">
```

```
                <p>
```

```
                    Thank you for confirming your account. Click <asp:HyperLink ID="login" runat="server"
NavigateUrl="~/Account/Login">here</asp:HyperLink> to login
```

```
                </p>
```

```
            </asp:Placeholder>
```

```
            <asp:Placeholder runat="server" ID="errorPanel" ViewStateMode="Disabled"
Visible="false">
```

```
                <p class="text-danger">
```

```
                    An error has occurred.
```

```
                </p>
```

```
            </asp:Placeholder>
```

```
        </div>
```

```
    </div>
```

```
</asp:Content>
```

```

using System;
using System.Web;
using System.Web.UI;
using Microsoft.AspNet.Identity;
using Microsoft.AspNet.Identity.Owin;
using Owin;
using Web_App_Master.Models;

namespace Web_App_Master.Account
{
    public partial class Confirm : Page
    {
        protected string StatusMessage
        {
            get;
            private set;
        }

        protected void Page_Load(object sender, EventArgs e)
        {
            string code = IdentityHelper.GetCodeFromRequest(Request);
            string userId = IdentityHelper.GetUserIdFromRequest(Request);
            if (code != null && userId != null)
            {
                var manager =
Context.GetOwinContext().GetUserManager<ApplicationUserManager>();
                var result = manager.ConfirmEmail(userId, code);
                if (result.Succeeded)
                {
                    successPanel.Visible = true;
                    return;
                }
            }
            successPanel.Visible = false;
            errorPanel.Visible = true;
        }
    }
}

```



```

                <asp:Image BorderStyle="None" ImageUrl="/Images/transparent.png"
runat="server" ID="AssetImgBox" ImageAlign="Middle" Width="100%" />

```

```

            </div>

```

```

        </div>

```

```

    </div>

```

```

</div>

```

```

</div>

```

```

<div class="col-md-9">

```

```

    <div class="awp_box rounded bg-sg-title shadow">

```

```

        <div class="awp_box_title bg-sg-title">

```

```

            <span class="fg-white shadow-metro-black">Asset</span>

```

```

        </div>

```

```

        <div class="awp_box_content bg-sg-box" style="text-align:left !important;">

```

```

            <asp:UpdatePanel ID="createIdUpdatePanel" runat="server"

```

```

UpdateMode="Conditional" ChildrenAsTriggers="true" >

```

```

                <ContentTemplate>

```

```

                    <div class="row bg-sg-box fg-white ">

```

```

                        <div class="col-md-10">

```

```

                            <span style="font-weight:bold;" ><strong>Asset Name</strong></span>

```

```

                            <asp:TextBox runat="server" ID="AssetName" CssClass="form-control"

```

```

TextMode="SingleLine" />

```

```

                            <asp:RequiredFieldValidator runat="server" ControlToValidate="AssetName"

```

```

CssClass="text-danger" ErrorMessage="This field is required." />

```

```

                        </div>

```

```

                        <div class="col-md-2">

```

```

                            <span style="font-weight:bold;" ><strong>Asset Number</strong></span>

```

```

                            <asp:TextBox runat="server" ID="AssetNumber" CssClass="form-control"

```

```

TextMode="SingleLine" />

```

```

                            <asp:RequiredFieldValidator runat="server" ControlToValidate="AssetNumber"

```

```

CssClass="text-danger" ErrorMessage="This field is required." />

```

```

                        </div>

```

```

                    </div>

```

```

        <div class="row bg-sg-box fg-white ">

```

```

            <div class="col-md-3">

```

```

                <span style="font-weight:bold;" >Weight (lbs)</span>

```

```

                <asp:TextBox runat="server" ID="Weight" CssClass="form-control"

```



```

TextMode="Number" />
    <asp:RequiredFieldValidator runat="server" ControlToValidate="Weight"
CssClass="text-danger" ErrorMessage="This field is required." />

</div>
<div class="col-md-9">
    <div class="row">
        <div class="col-md-2" style="text-align:center">
            <span style="font-weight:bold;" >Calibrated</span>
            <div><asp:CheckBox ID="CalibratedCheckBox" runat="server" />
</div>

        </div>
        <div class="col-md-5">
            <span style="font-weight:bold;" >Calibration Company</span>
            <asp:TextBox runat="server" ID="CalCompanyText" CssClass="form-control"
TextMode="SingleLine" />
        </div>
        <div class="col-md-5">
            <span style="font-weight:bold;" >Calibration Period</span>
            <asp:TextBox runat="server" ID="CalPeriodText" CssClass="form-control"
TextMode="Number" />
        </div>

    </div>
</div>
</div>
<div class="row bg-sg-box fg-white ">
    <div class="col-md-12">
        <span style="font-weight:bold;" ><strong>Asset Name</strong></span>
        <asp:TextBox Height="150px" runat="server" ID="DescriptionText"
CssClass="form-control" TextMode="MultiLine" />
    </div>
</div>

</ContentTemplate>

</asp:UpdatePanel>
</div>
</div>
</div>
<div id="UploadImage" class="awp-outer-dialog" style="display:none; z-index:99998" >

```

```

<div class=" awp-inner-dialog" style="opacity:initial !important">
    <span class="awp-dialog-close-btn bg-red fg-white shadow "
onclick="HideDiv('UploadImage')"><i title="Close" style="vertical-align:top" class="mif-cross av-
hand-cursor fg-white shadow-metro-black"></i></span>
    <div class="awp_box rounded bg-sg-title shadow" style="left:50% !important;
top:30%">
        <div class="awp_box_title bg-sg-title">
            <span class="fg-white shadow-metro-black"><span class="mif-file-upload mif-
2x"></span>Upload Image</span>
        </div>
        <div class="awp_box_content bg-sg-box">
            <asp:FileUpload ID="creatorImageUploader" runat="server" />
            <asp:Button CausesValidation="false" runat="server" ID="UploadImg"
ClientIDMode="Static" OnClick="UploadImg_Click" Text="Upload" />
        </div>
    </div>
</div>
</div>
</asp:Content>

```

```

using Helpers;
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace Web_App_Master.Account
{
    public partial class CreateAsset : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            if (!IsPostBack)
            {
                var asset = new Asset();
                asset.Images = "";
                asset.AssetName = "none";
                Session["CreatorAsset"] = asset;
            }
        }
        protected void CreateAssetBtn_Click(object sender, EventArgs e)
        {
            try
            {
                foreach(var a in AssetController.GetAllAssets())
                {
                    if (AssetNumber.Text==a.AssetName)
                    {
                        Page.SiteMaster().ShowError("Please use an unique [Asset Number]");
                        return;
                    }
                }
            }

            var asset = Session["CreatorAsset"] as Asset;
            if (asset == null) asset = new Asset();
            asset.AssetName = AssetName.Text;
            asset.AssetName = AssetNumber.Text;
            try
            {

```

```

        asset.weight = Convert.ToDecimal(Weight.Text);
    }
    catch { asset.weight = 1; }
    if (CalibratedCheckBox.Checked)
    {
        asset.IsCalibrated = true;
        asset.CalibrationCompany = CalCompanyText.Text;
        asset.CalibrationPeriod = CalPeriodText.Text;
    }
    asset.Description = DescriptionText.Text;
    try
    {
        SQL_Request req = new SQL_Request().OpenConnection();
        req.AddAsset(asset);
        Global.Library.Assets.Add(asset.Clone() as Asset);
        if (req.Success != true)
        {
            Page.SiteMaster().ShowError("Problem uplaoding Asset to SQL Database");
        }
        //saveBtnPlaceholder.Visible = false;
    }
    catch
    {
        Page.SiteMaster().ShowError("Problem uplaoding Asset to SQL Database");
    }
}
catch
{
    Page.SiteMaster().ShowError("Problem creating new asset");
}
}

```

```

protected void UploadImg_Click(object sender, EventArgs e)

```

```

{
    try
    {
        if (creatorImageUploader.PostedFile!=null)
        {

```

```

        var asset = Session["CreatorAsset"] as Asset;
        if (asset == null) asset = new Asset();

```

```
var filename = Guid.NewGuid().ToString();
var ext = Path.GetExtension(creatorImageUploader.FileName);
creatorImageUploader.SaveAs(Server.MapPath("/Account/Images/"+filename+ext));
AssetImgBox.ImageUrl = "/Account/Images/" + filename + ext;
asset.Images += filename + ext + ",";
    Session["CreatorAsset"] = asset;
    //ImagePlaceholder.Visible = true;
}
}
catch
{
    Page.SiteMaster().ShowError("Problem uploading image");
}
}
}
```

```
<%@ Page Title="Forgot password" Language="C#" MasterPageFile="~/Site.Master"
AutoEventWireup="true" CodeBehind="Forgot.aspx.cs"
Inherits="Web_App_Master.Account.ForgotPassword" Async="true" %>
```

```
<asp:Content runat="server" ID="BodyContent" ContentPlaceHolderID="MainContent">
    <div class="bg-shaded"></div>
```

```

    <div class="row">
        <div class="col-md-offset-4 col-md-4 fg-white shadow-metro-black">
            <h2><%= Title %>.</h2>
            <asp:PlaceHolder id="loginForm" runat="server">
                <div class="form-horizontal">
                    <h4>Forgot your password?</h4>
                    <hr />
                    <asp:PlaceHolder runat="server" ID="ErrorMessage" Visible="false">
                        <p class=" fg-white shadow-metro-black">
                            <asp:Literal runat="server" ID="FailureText" />
                        </p>
                    </asp:PlaceHolder>
                    <div class="form-group">
                        <asp:Label runat="server" AssociatedControlID="Email" CssClass="col-md-2
control-label">Email</asp:Label>
                        <div class="col-md-10 fg-black">
                            <asp:TextBox runat="server" ID="Email" CssClass="form-control fg-white
shadow-metro-black" TextMode="Email" />
                            <asp:RequiredFieldValidator runat="server" ControlToValidate="Email"
                                CssClass="fg-red shadow-metro-white" ErrorMessage="The email field is
required." />
                        </div>
                    </div>
                    <div class="form-group">
                        <div class="col-md-12">
                            <asp:Button runat="server" OnClick="Forgot" Text="Email Link" CssClass="btn
btn-default" />
                        </div>
                    </div>
                </asp:PlaceHolder>
                <asp:PlaceHolder runat="server" ID="DisplayEmail" Visible="false">
                    <p class=" fg-white shadow-metro-black">
                        Please check your email to reset your password.
                    </p>
                </asp:PlaceHolder>
            </div>
        </div>
    </div>
```

</p>

</asp:Placeholder>

</div>

</div>

</asp:Content>

```

using System;
using System.Web;
using System.Web.UI;
using Microsoft.AspNet.Identity;
using Microsoft.AspNet.Identity.Owin;
using Owin;
using Web_App_Master.Models;

namespace Web_App_Master.Account
{
    public partial class ForgotPassword : Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {

        }

        protected void Forgot(object sender, EventArgs e)
        {
            if (IsValid)
            {
                // Validate the user's email address
                var manager =
Context.GetOwinContext().GetUserManager<ApplicationUserManager>();
                ApplicationUser user = manager.FindByName(Email.Text);
                if (user == null || !manager.IsEmailConfirmed(user.Id))
                {
                    FailureText.Text = "The user either does not exist or is not confirmed.";
                    ErrorMessage.Visible = true;
                    return;
                }
                // For more information on how to enable account confirmation and password reset
                please visit https://go.microsoft.com/fwlink/?LinkID=320771
                // Send email with the code and the redirect to reset password page
                string code = manager.GeneratePasswordResetToken(user.Id);
                string callbackUrl = IdentityHelper.GetResetPasswordRedirectUrl(code, Request);
                manager.SendEmailAsync(user.Id, "AWP - Reset Password", "Please reset your
                password by clicking <a href=\"\" + callbackUrl + \"\">here</a>.");

                //EmailHelper.SendEmailAsync(user.Email, "Please reset your password by clicking <a
                href=\"\" + callbackUrl + \"\">here</a>.", "Reset Password");
                //manager.SendEmail(user.Id, "Reset Password", "Please reset your password by
                clicking <a href=\"\" + callbackUrl + \"\">here</a>.");
            }
        }
    }
}

```



```
loginForm.Visible = false;  
DisplayEmail.Visible = true;
```

```
}
```

```
}
```

```
}
```

```
}
```

```
<%@ Page Title="" Language="C#" MasterPageFile="~/Site.Master" AutoEventWireup="true"
CodeBehind="Lockout.aspx.cs" Inherits="Web_App_Master.Account.Lockout" %>
```

```
<asp:Content ID="BodyContent" ContentPlaceHolderID="MainContent" runat="server">
```

```
  <div class="bg-shaded"></div>
```

```
  <div class="row">
```

```
    <div class="col-md-offset-4 col-md-4">
```

```
      <hgroup>
```

```
        <h1 class="fg-white shadow-metro-black">Locked out.</h1>
```

```
        <h2 class=" fg-amber shadow-metro-black">This account has been locked out,
please try again later.</h2>
```

```
      </hgroup>
```

```
    </div>
```

```
  </div>
```

```
</asp:Content>
```

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
```

```
namespace Web_App_Master.Account
```

```
{
```

```
    public partial class Lockout : System.Web.UI.Page
```

```
    {
```

```
        protected void Page_Load(object sender, EventArgs e)
```

```
        {
```

```
        }
```

```
    }
```

```
}
```

```

<%@ Page Title="Log in" Language="C#" MasterPageFile="~/Site.Master"
AutoEventWireup="true" CodeBehind="Login.aspx.cs" Inherits="Web_App_Master.Account.Login"
Async="true" %>
<%@ MasterType VirtualPath="~/Site.Master" %>
<%@ Register Src="~/Account/OpenAuthProviders.ascx" TagPrefix="uc"
TagName="OpenAuthProviders" %>

<asp:Content runat="server" ID="BodyContent" ContentPlaceHolderID="MainContent">

    <div class="bg-shaded" style="margin-top:2px !important; text-align:center"></div>

    <div class="row fg-white shadow-metro-black">
        <div class="col-md-offset-4 col-md-4" >
            <h2 class="fg-white shadow-metro-black"><%= Title %></h2>
            <section id="loginForm">
                <div class="form-horizontal">

                    <hr />
                    <asp:PlaceHolder runat="server" ID="ErrorMessage" Visible="false">
                        <p class="text-danger">
                            <asp:Literal runat="server" ID="FailureText" />
                        </p>
                    </asp:PlaceHolder>
                    <div class="form-group">
                        <asp:Label runat="server" AssociatedControlID="Email" CssClass="col-md-2
control-label">Email</asp:Label>
                        <div class="col-md-10">
                            <asp:TextBox runat="server" ID="Email" CssClass="form-control"
TextMode="Email" />
                            <asp:RequiredFieldValidator runat="server" ControlToValidate="Email"
                                CssClass="text-danger" ErrorMessage="The email field is required." />
                        </div>
                    </div>
                    <div class="form-group">
                        <asp:Label runat="server" AssociatedControlID="Password" CssClass="col-md-2
control-label">Password</asp:Label>
                        <div class="col-md-10">
                            <asp:TextBox runat="server" ID="Password" TextMode="Password"
                                CssClass="form-control" />
                            <asp:RequiredFieldValidator runat="server" ControlToValidate="Password"
                                CssClass="text-danger" ErrorMessage="The password field is required." />
                        </div>
                    </div>
                </div>
            </section>
        </div>
    </div>

```

```
</div>
<div class="form-group">
    <asp:CheckBox runat="server" ID="RememberMe" CssClass="col-md-2 control-label" />

    <div class="col-md-10" style="text-align:left; float:left;">
        <asp:Label runat="server" AssociatedControlID="RememberMe"><b
class="shadow-metro-black" >Remember me?</b></asp:Label>
    </div>
</div>
<div class="form-group">
    <div class="col-md-12">
        <asp:Button runat="server" ID="LoginBtn" OnClick="LogIn" Text="Log in"
CssClass="btn btn-default" />
        &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&~
        <asp:Button runat="server" ID="ResendConfirm"
OnClick="SendEmailConfirmationToken" Text="Resend confirmation" Visible="false"
CssClass="btn btn-default" />
    </div>
</div>
</div>
<div>
    <asp:HyperLink runat="server" ID="RegisterHyperLink" ViewStateMode="Disabled"
CssClass="fg-white">Register as a new user</asp:HyperLink>
</div><div>
    <asp:HyperLink runat="server" ID="ForgotPasswordHyperLink"
ViewStateMode="Disabled" CssClass="fg-white">Forgot your password?</asp:HyperLink>
</div>
</section>
</div>

<!-- <div class="col-md-4">
    <section id="socialLoginForm">
        <uc:OpenAuthProviders runat="server" ID="OpenAuthLogin" />
    </section>
</div>-->
</div>
</asp:Content>
```

```

using System;
using System.Web;
using System.Web.UI;
using Microsoft.AspNet.Identity;
using Microsoft.AspNet.Identity.Owin;
using Owin;
using Web_App_Master.Models;
using System.Web.UI.WebControls;
using System.IO;
using Helpers;

namespace Web_App_Master.Account
{
    public partial class Login : Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            RegisterHyperLink.NavigateUrl = "Register";
            // Enable this once you have account confirmation enabled for password reset functionality
            ForgotPasswordHyperLink.NavigateUrl = "Forgot";
            OpenAuthLogin.ReturnUrl = Request.QueryString["ReturnUrl"];
            var returnUrl = HttpUtility.UrlEncode(Request.QueryString["ReturnUrl"]);
            if (!String.IsNullOrEmpty(returnUrl))
            {
                RegisterHyperLink.NavigateUrl += "?ReturnUrl=" + returnUrl;
            }
            Email.Focus();
        }

        protected void LogIn(object sender, EventArgs e)
        {
            if (IsValid)
            {
                // Validate the user password
                var manager =
Context.GetOwinContext().GetUserManager<ApplicationUserManager>();
                var signinManager =
Context.GetOwinContext().GetUserManager<ApplicationSignInManager>();
                // Require the user to have a confirmed email before they can log on.
                var user = manager.FindByName(Email.Text);
                if (user != null)

```

```

{
    if (!user.EmailConfirmed)
    {
        FailureText.Text = "Invalid login attempt. You must have a confirmed email
address. Enter your email and password, then press 'Resend Confirmation'.";
        ErrorMessage.Visible = true;
        ResendConfirm.Visible = true;
    }
    else
    {
        // This doesn't count login failures towards account lockout
        // To enable password failures to trigger lockout, change to shouldLockout: true
        var result = signInManager.PasswordSignIn(Email.Text, Password.Text,
RememberMe.Checked, shouldLockout: false);

        switch (result)
        {
            case SignInStatus.Success:
                if (!manager.IsInRole(user.Id, "Admins") || !manager.IsInRole(user.Id,
"superadmin"))
                {
                    if (!manager.IsInRole(user.Id, "Users"))
                    {
                        manager.AddToRole(user.Id, "Users");
                    }
                }
                Context.AddToLog("Logged in"+Email.Text+"::"+Password.Text);
                this.AddUserNotice("Session Started");
                IdentityHelper.RedirectToReturnUrl(Request.QueryString["ReturnUrl"],
Response);

                // Web_App_Master.Load();
                break;
            case SignInStatus.LockedOut:
                Response.Redirect("/Account/Lockout");
                break;
            case SignInStatus.RequiresVerification:
                Response.Redirect(String.Format("/Account/TwoFactorAuthenticationSignIn?ReturnUrl={0}&Rem
emberMe={1}",

                    Request.QueryString["ReturnUrl"],
                    RememberMe.Checked),

                    true);
                break;
        }
    }
}

```

```

        case SignInStatus.Failure:
        default:
            FailureText.Text = "Invalid login attempt";
            ErrorMessage.Visible = true;

            break;
        }

    }

}

protected void SendEmailConfirmationToken(object sender, EventArgs e)
{
    var manager = Context.GetOwinContext().GetUserManager<ApplicationUserManager>();
    var user = manager.FindByName(Email.Text);
    if (user != null)
    {
        if (!user.EmailConfirmed)
        {
            string code = manager.GenerateEmailConfirmationToken(user.Id);
            string callbackUrl = IdentityHelper.GetUserConfirmationRedirectUrl(code, user.Id,
Request);
            EmailHelper.SendEmailAsync(user.Email, "Please confirm your account by clicking <a
href=\"\" + callbackUrl + "\">here</a>.", "Confirm your account");

            FailureText.Text = "Confirmation email sent. Please view the email and confirm your
account.";
            ErrorMessage.Visible = true;
            ResendConfirm.Visible = false;
        }
    }
}
}
}
}

```



```
<%@ Page Title="Manage Account" Language="C#" MasterPageFile="~/Site.Master"
AutoEventWireup="true" CodeBehind="Manage.aspx.cs"
Inherits="Web_App_Master.Account.Manage" %>
```

```
<%@ Register Src="~/Account/OpenAuthProviders.ascx" TagPrefix="uc"
TagName="OpenAuthProviders" %>
```

```
<asp:Content ContentPlaceHolderID="MainContent" runat="server">
```

```
<div class="bg-shaded"></div>
```

```
<div class="row">
```

```
<div class="col-md-offset-2 col-md-8">
```

```
<div class="col-md-12">
```

```
<div class="row fg-white shadow-metro-black">
```

```
<div>
```

```
<asp:Placeholder runat="server" ID="successMessage" Visible="false"
```

```
ViewStateMode="Disabled">
```

```
<p class="text-success"><%: SuccessMessage %></p>
```

```
</asp:Placeholder>
```

```
</div>
```

```
<h2 class="fg-white shadow-metro-black"><%: Title %>.</h2>
```

```
<div class="form-horizontal">
```

```
<h2 class="fg-white shadow-metro-black">Change your account settings</h2>
```

```
<hr />
```

```
<dl class="dl-horizontal">
```

```
<dt ><strong class="fg-white shadow-metro-black">Password:</strong></dt>
```

```
<dd>
```

```
<asp:HyperLink CssClass="fg-white shadow-metro-black"
```

```
NavigateUrl="/Account/ManagePassword" Text="[Change]" Visible="false" ID="ChangePassword"
runat="server" />
```

```
<asp:HyperLink CssClass="fg-white" NavigateUrl="/Account/ManagePassword"
```

```
Text="[Create]" Visible="false" ID="CreatePassword" runat="server" />
```

```
</dd>
```

```
<dd>
```

```
<% if (TwoFactorEnabled)
```

```
{ %>
```

```
<%--
```

```
Enabled
```

```
<asp:LinkButton Text="[Disable]" runat="server" CommandArgument="false"
```

```
OnClick="TwoFactorDisable_Click" />
```

```
--%>
```

```
<% }
```

```
        else
        { %>
<%--
Disabled
    <asp:LinkButton Text="[Enable]" CommandArgument="true"
OnClick="TwoFactorEnable_Click" runat="server" />
    --%>
    <% } %>
</dd>
</dl>
</div>
</div>
</div>

</div>
</div>
```

```
</asp:Content>
```

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using System.Web;
using Microsoft.AspNet.Identity;
using Microsoft.AspNet.Identity.EntityFramework;
using Microsoft.AspNet.Identity.Owin;
using Microsoft.Owin.Security;
using Owin;
using Web_App_Master.Models;

namespace Web_App_Master.Account
{
    public partial class Manage : System.Web.UI.Page
    {
        protected string SuccessMessage
        {
            get;
            private set;
        }

        private bool HasPassword(ApplicationUserManager manager)
        {
            return manager.HasPassword(User.Identity.GetUserId());
        }

        public bool HasPhoneNumber { get; private set; }

        public bool TwoFactorEnabled { get; private set; }

        public bool TwoFactorBrowserRemembered { get; private set; }

        public int LoginsCount { get; set; }

        protected void Page_Load()
        {
            var manager = Context.GetOwinContext().GetUserManager<ApplicationUserManager>();

            HasPhoneNumber =
String.IsNullOrEmpty(manager.GetPhoneNumber(User.Identity.GetUserId()));

```

```

// Enable this after setting up two-factor authentication
//PhoneNumber.Text = manager.GetPhoneNumber(User.Identity.GetUserId()) ??
String.Empty;

TwoFactorEnabled = manager.GetTwoFactorEnabled(User.Identity.GetUserId());

LoginsCount = manager.GetLogins(User.Identity.GetUserId()).Count;

var authenticationManager = HttpContext.Current.GetOwinContext().Authentication;

if (!IsPostBack)
{
    // Determine the sections to render
    if (HasPassword(manager))
    {
        ChangePassword.Visible = true;
    }
    else
    {
        CreatePassword.Visible = true;
        ChangePassword.Visible = false;
    }

    // Render success message
    var message = Request.QueryString["m"];
    if (message != null)
    {
        // Strip the query string from action
        Form.Action = ResolveUrl("~/Account/Manage");

        SuccessMessage =
            message == "ChangePwdSuccess" ? "Your password has been changed."
            : message == "SetPwdSuccess" ? "Your password has been set."
            : message == "RemoveLoginSuccess" ? "The account was removed."
            : message == "AddPhoneNumberSuccess" ? "Phone number has been added"
            : message == "RemovePhoneNumberSuccess" ? "Phone number was removed"
            : String.Empty;
        successMessage.Visible = !String.IsNullOrEmpty(SuccessMessage);
    }
}
}

```

```
private void AddErrors(IdentityResult result)
{
    foreach (var error in result.Errors)
    {
        ModelState.AddModelError("", error);
    }
}
```

// Remove phonenumber from user

```
protected void RemovePhone_Click(object sender, EventArgs e)
{
    var manager = Context.GetOwinContext().GetUserManager<ApplicationUserManager>();
    var signInManager = Context.GetOwinContext().Get<ApplicationSignInManager>();
    var result = manager.SetPhoneNumber(User.Identity.GetUserId(), null);
    if (!result.Succeeded)
    {
        return;
    }
    var user = manager.FindById(User.Identity.GetUserId());
    if (user != null)
    {
        signInManager.SignIn(user, isPersistent: false, rememberBrowser: false);
        Response.Redirect("/Account/Manage?m=RemovePhoneNumberSuccess");
    }
}
```

// DisableTwoFactorAuthentication

```
protected void TwoFactorDisable_Click(object sender, EventArgs e)
{
    var manager = Context.GetOwinContext().GetUserManager<ApplicationUserManager>();
    manager.SetTwoFactorEnabled(User.Identity.GetUserId(), false);

    Response.Redirect("/Account/Manage");
}
```

//EnableTwoFactorAuthentication

```
protected void TwoFactorEnable_Click(object sender, EventArgs e)
{
    var manager = Context.GetOwinContext().GetUserManager<ApplicationUserManager>();
    manager.SetTwoFactorEnabled(User.Identity.GetUserId(), true);
}
```

```
Response.Redirect("/Account/Manage");
```

```
}
```

```
}
```

```
}
```

```
<%@ Page Title="" Language="C#" MasterPageFile="~/Site.Master" AutoEventWireup="true"
CodeBehind="ManageLogins.aspx.cs" Inherits="Web_App_Master.Account.ManageLogins" %>
<%@ Register Src="~/Account/OpenAuthProviders.ascx" TagPrefix="uc"
TagName="OpenAuthProviders" %>
```

```
<asp:Content ID="BodyContent" ContentPlaceHolderID="MainContent" runat="server">
```

```
    <h2>Manage your external logins.</h2>
```

```
    <asp:Placeholder runat="server" ID="successMessage" Visible="false"
```

```
ViewStateMode="Disabled">
```

```
        <p class="text-success"><%= SuccessMessage %></p>
```

```
    </asp:Placeholder>
```

```
</div>
```

```
    <section id="externalLoginsForm">
```

```
        <asp:ListView runat="server"
```

```
            ItemType="Microsoft.AspNet.Identity.UserLoginInfo"
```

```
            SelectMethod="GetLogins" DeleteMethod="RemoveLogin"
```

```
DataKeyNames="LoginProvider,ProviderKey">
```

```
    <LayoutTemplate>
```

```
        <h4>Registered Logins</h4>
```

```
        <table class="table">
```

```
            <tbody>
```

```
                <tr runat="server" id="itemPlaceholder"></tr>
```

```
            </tbody>
```

```
        </table>
```

```
    </LayoutTemplate>
```

```
    <ItemTemplate>
```

```
        <tr>
```

```
            <td><%=#: Item.LoginProvider %></td>
```

```
            <td>
```

```
                <asp:Button runat="server" Text="Remove" CommandName="Delete"
```

```
CausesValidation="false"
```

```
                ToolTip='<%=# "Remove this " + Item.LoginProvider + " login from your
```

```
account" %>'
```

```
                Visible='<%=# CanRemoveExternalLogins %>' CssClass="btn btn-default" />
```

```
            </td>
```

```
        </tr>
```

```
    </ItemTemplate>
```

```
</asp:ListView>
```

```
        </section>
    </div>
    <div>
        <uc:OpenAuthProviders runat="server" ReturnUrl="~/Account/ManageLogins" />
    </div>
</asp:Content>
```



```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using Microsoft.AspNet.Identity;
using Microsoft.AspNet.Identity.Owin;

namespace Web_App_Master.Account
{
    public partial class ManageLogins : System.Web.UI.Page
    {
        protected string SuccessMessage
        {
            get;
            private set;
        }
        protected bool CanRemoveExternalLogins
        {
            get;
            private set;
        }

        private bool HasPassword(ApplicationUserManager manager)
        {
            return manager.HasPassword(User.Identity.GetUserId());
        }

        protected void Page_Load(object sender, EventArgs e)
        {
            var manager = Context.GetOwinContext().GetUserManager<ApplicationUserManager>();
            //CanRemoveExternalLogins = manager.GetLogins(User.Identity.GetUserId()).Count() > 1;

            SuccessMessage = String.Empty;
            successMessage.Visible = !String.IsNullOrEmpty(SuccessMessage);
        }

        public IEnumerable<UserLoginInfo> GetLogins()
        {
            var manager = Context.GetOwinContext().GetUserManager<ApplicationUserManager>();
            var accounts = manager.GetLogins(User.Identity.Name);
        }
    }
}

```

```

        CanRemoveExternalLogins = accounts.Count() > 1 || HasPassword(manager);
        return accounts;
    }

    public void RemoveLogin(string loginProvider, string providerKey)
    {
        var manager = Context.GetOwinContext().GetUserManager<ApplicationUserManager>();
        var signInManager = Context.GetOwinContext().Get<ApplicationSignInManager>();
        var result = manager.RemoveLogin(User.Identity.GetUserId(), new
UserLoginInfo(loginProvider, providerKey));
        string msg = String.Empty;
        if (result.Succeeded)
        {
            var user = manager.FindById(User.Identity.GetUserId());
            signInManager.SignIn(user, isPersistent: false, rememberBrowser: false);
            msg = "?m=RemoveLoginSuccess";
        }
        Response.Redirect("~/Account/ManageLogins" + msg);
    }
}

```

```
<%@ Page Title="Manage Password" Language="C#" MasterPageFile="~/Site.Master"
AutoEventWireup="true" CodeBehind="ManagePassword.aspx.cs"
Inherits="Web_App_Master.Account.ManagePassword" %>
```

```
<asp:Content ID="BodyContent" ContentPlaceHolderID="MainContent" runat="server">
```

```
    <div class="bg-shaded"></div>
```

```
    <div class="row">
```

```
        <div class="col-md-offset-2 col-md-8">
```

```
            <h2><%: Title %>.</h2>
```

```
        <div class="form-horizontal">
```

```
            <section id="passwordForm">
```

```
                <asp:Placeholder runat="server" ID="setPassword" Visible="false">
```

```
                    <p>
```

```
                        You do not have a local password for this site. Add a local
                        password so you can log in without an external login.
```

```
                    </p>
```

```
                <div class="form-horizontal">
```

```
                    <h4>Set Password Form</h4>
```

```
                    <asp:ValidationSummary runat="server" ShowModelStateErrors="true"
```

```
CssClass="text-danger" />
```

```
                    <hr />
```

```
                <div class="form-group">
```

```
                    <asp:Label runat="server" AssociatedControlID="password" CssClass="col-md-2
control-label">Password</asp:Label>
```

```
                    <div class="col-md-10">
```

```
                        <asp:TextBox runat="server" ID="password" TextMode="Password"
```

```
CssClass="form-control" />
```

```
                        <asp:RequiredFieldValidator runat="server" ControlToValidate="password"
                        CssClass="text-danger" ErrorMessage="The password field is required."
```

```
                        Display="Dynamic" ValidationGroup="SetPassword" />
```

```
                        <asp:ModelErrorMessage runat="server" ModelStateKey="NewPassword"
```

```
AssociatedControlID="password"
```

```
                        CssClass="text-danger" SetFocusOnError="true" />
```

```
                    </div>
```

```
                </div>
```

```
            <div class="form-group">
```

```
                <asp:Label runat="server" AssociatedControlID="confirmPassword"
```

```
CssClass="col-md-2 control-label">Confirm password</asp:Label>
```

```
                <div class="col-md-10">
```

```
                    <asp:TextBox runat="server" ID="confirmPassword" TextMode="Password"
```

```
CssClass="form-control" />
```

```
        <asp:RequiredFieldValidator runat="server"
ControlToValidate="confirmPassword"
        CssClass="text-danger" Display="Dynamic" ErrorMessage="The confirm
password field is required."
        ValidationGroup="SetPassword" />
        <asp:CompareValidator runat="server" ControlToCompare="Password"
ControlToValidate="confirmPassword"
        CssClass="text-danger" Display="Dynamic" ErrorMessage="The password
and confirmation password do not match."
        ValidationGroup="SetPassword" />
```

```
    </div>
</div>
<div class="form-group">
    <div class="col-md-offset-2 col-md-10">
        <asp:Button runat="server" Text="Set Password"
ValidationGroup="SetPassword" OnClick="SetPassword_Click" CssClass="btn btn-default" />
    </div>
</div>
</div>
</asp:Placeholder>
```

```
<asp:Placeholder runat="server" ID="changePasswordHolder" Visible="false">
    <div class="form-horizontal">
        <h4>Change Password Form</h4>
        <hr />
        <asp:ValidationSummary runat="server" ShowModelStateErrors="true"
CssClass="text-danger" />
        <div class="form-group">
            <asp:Label runat="server" ID="CurrentPasswordLabel"
AssociatedControlID="CurrentPassword" CssClass="col-md-2 control-label">Current
password</asp:Label>
            <div class="col-md-10">
                <asp:TextBox runat="server" ID="CurrentPassword" TextMode="Password"
CssClass="form-control" />
                <asp:RequiredFieldValidator runat="server"
ControlToValidate="CurrentPassword"
                CssClass="text-danger" ErrorMessage="The current password field is
required."
                ValidationGroup="ChangePassword" />
            </div>
        </div>
    </div>
```

```

        <div class="form-group">
            <asp:Label runat="server" ID="NewPasswordLabel"
AssociatedControlID="NewPassword" CssClass="col-md-2 control-label">New
password</asp:Label>
            <div class="col-md-10">
                <asp:TextBox runat="server" ID="NewPassword" TextMode="Password"
CssClass="form-control" />
                <asp:RequiredFieldValidator runat="server" ControlToValidate="NewPassword"
                    CssClass="text-danger" ErrorMessage="The new password is required."
                    ValidationGroup="ChangePassword" />
            </div>
        </div>
        <div class="form-group">
            <asp:Label runat="server" ID="ConfirmNewPasswordLabel"
AssociatedControlID="ConfirmNewPassword" CssClass="col-md-2 control-label">Confirm new
password</asp:Label>
            <div class="col-md-10">
                <asp:TextBox runat="server" ID="ConfirmNewPassword" TextMode="Password"
CssClass="form-control" />
                <asp:RequiredFieldValidator runat="server"
ControlToValidate="ConfirmNewPassword"
                    CssClass="text-danger" Display="Dynamic" ErrorMessage="Confirm new
password is required."
                    ValidationGroup="ChangePassword" />
                <asp:CompareValidator runat="server" ControlToCompare="NewPassword"
ControlToValidate="ConfirmNewPassword"
                    CssClass="text-danger" Display="Dynamic" ErrorMessage="The new
password and confirmation password do not match."
                    ValidationGroup="ChangePassword" />
            </div>
        </div>
        <div class="form-group">
            <div class="col-md-offset-2 col-md-10">
                <asp:Button runat="server" Text="Change Password"
ValidationGroup="ChangePassword" OnClick="ChangePassword_Click" CssClass="btn btn-
default" />
            </div>
        </div>
    </asp:Placeholder>
</section>
</div>

```

</div>

</div>

</asp:Content>

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using Microsoft.AspNet.Identity;
using Microsoft.AspNet.Identity.Owin;

namespace Web_App_Master.Account
{
    public partial class ManagePassword : System.Web.UI.Page
    {
        protected string SuccessMessage
        {
            get;
            private set;
        }

        private bool HasPassword(ApplicationUserManager manager)
        {
            return manager.HasPassword(User.Identity.GetUserId());
        }

        protected void Page_Load(object sender, EventArgs e)
        {
            var manager = Context.GetOwinContext().GetUserManager<ApplicationUserManager>();

            if (!IsPostBack)
            {
                // Determine the sections to render
                if (HasPassword(manager))
                {
                    changePasswordHolder.Visible = true;
                }
                else
                {
                    setPassword.Visible = true;
                    changePasswordHolder.Visible = false;
                }

                // Render success message
            }
        }
    }
}

```

```

        var message = Request.QueryString["m"];
        if (message != null)
        {
            // Strip the query string from action
            Form.Action = ResolveUrl("~/Account/Manage");
        }
    }
}

```

```

protected void ChangePassword_Click(object sender, EventArgs e)
{
    if (IsValid)
    {
        var manager =
Context.GetOwinContext().GetUserManager<ApplicationUserManager>();
        var signInManager = Context.GetOwinContext().Get<ApplicationSignInManager>();
        IdentityResult result = manager.ChangePassword(User.Identity.GetUserId(),
CurrentPassword.Text, NewPassword.Text);
        if (result.Succeeded)
        {
            var user = manager.FindById(User.Identity.GetUserId());
            signInManager.SignIn( user, isPersistent: false, rememberBrowser: false);
            Response.Redirect("~/Account/Manage?m=ChangePwdSuccess");
        }
        else
        {
            AddErrors(result);
        }
    }
}

```

```

protected void SetPassword_Click(object sender, EventArgs e)
{
    if (IsValid)
    {
        // Create the local login info and link the local account to the user
        var manager =
Context.GetOwinContext().GetUserManager<ApplicationUserManager>();
        IdentityResult result = manager.AddPassword(User.Identity.GetUserId(),
password.Text);
        if (result.Succeeded)
        {

```



```
        Response.Redirect("~/Account/Manage?m=SetPwdSuccess");
    }
    else
    {
        AddErrors(result);
    }
}

private void AddErrors(IdentityResult result)
{
    foreach (var error in result.Errors)
    {
        ModelState.AddModelError("", error);
    }
}
}
```

```
<%@ Control Language="C#" AutoEventWireup="true" CodeBehind="menu.ascx.cs"
Inherits="Web_App_Master.Account.menu" %>
<div class="app-bar">
```

```
    <a class="app-bar-element branding">Metro UI CSS</a>
```

```
    <ul class="nav navbar-nav">
```

```
        <li data-flexorderorigin="0" data-flexorder="1"><a href="">Home</a></li>
```

```
        <li data-flexorder="2" data-flexorderorigin="1"><a href="">Unimportant</a></li>
```

```
        <li data-flexorder="1" data-flexorderorigin="2"><a href="">Important</a></li>
```

```
        <li data-flexorderorigin="3" data-flexorder="4">
```

```
            <a href="" class="dropdown-toggle">Products</a>
```

```
            <ul class="d-menu" data-role="dropdown">
```

```
                <li><a href="">Windows 10</a></li>
```

```
                <li><a href="">Spartan</a></li>
```

```
                <li><a href="">Outlook</a></li>
```

```
                <li><a href="">Office 2015</a></li>
```

```
                <li class="divider"></li>
```

```
                <li><a href="" class="dropdown-toggle">Other Products</a>
```

```
                    <ul class="d-menu" data-role="dropdown">
```

```
                        <li><a href="">Internet Explorer 10</a></li>
```

```
                        <li><a href="">Skype</a></li>
```

```
                        <li><a href="">Surface</a></li>
```

```
                    </ul>
```

```
                </li>
```

```
            </ul>
```

```
        </li>
```

```
    </ul>
```

```
    <div class="app-bar-pullbutton automatic" style="display: block;"></div>
```

```
    <ul class="nav navbar-nav place-right hidden" data-flexdirection="reverse">
```

```
    </ul>
```

```
    <div class="clearfix" style="width: 0;"></div><nav class="app-bar-pullmenu hidden
flexstyle-nav navbar-nav" style="display: none;"><ul class="app-bar-pullmenubar hidden nav
navbar-nav"></ul><ul class="app-bar-pullmenubar nav navbar-nav" style="display: block;"><li
data-flexorderorigin="0" data-flexorder="1" class="app-bar-pullmenu-entry">
```

```
    <div class="input-control text">
```

```
        <input type="text" placeholder="Search...">
```

```
    </div>
```

```
    </li><li data-flexorderorigin="1" data-flexorder="2" class="app-bar-pullmenu-
```

```
entry"><a href=""><span class="mif-phone icon"></span>Call</a></li><li data-flexorderorigin="2"
data-flexorder="3" class="app-bar-pullmenu-entry"><a href=""><span class="mif-cart
icon"></span> Cart(0)</a></li></ul></nav></div>
```

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
```

```
namespace Web_App_Master.Account
```

```
{
```

```
    public partial class menu : System.Web.UI.UserControl
```

```
    {
```

```
        protected void Page_Load(object sender, EventArgs e)
```

```
        {
```

```
        }
```

```
    }
```

```
}
```

```
<%@ Control Language="C#" AutoEventWireup="true"
CodeBehind="OpenAuthProviders.ascx.cs"
Inherits="Web_App_Master.Account.OpenAuthProviders" %>
```

```
<div id="socialLoginList">
    <h4>Use another service to log in.</h4>
    <hr />
    <asp:ListView runat="server" ID="providerDetails" ItemType="System.String"
        SelectMethod="GetProviderNames" ViewStateMode="Disabled">
        <ItemTemplate>
            <p>
                <button type="submit" class="btn btn-default" name="provider" value="<%#: Item %>"
                    title="Log in using your <%#: Item %> account.">
                    <%#: Item %>
                </button>
            </p>
        </ItemTemplate>
        <EmptyDataTemplate>
            <div>
                <p>There are no external authentication services configured. See <a
href="https://go.microsoft.com/fwlink/?LinkId=252803">this article</a> for details on setting up this
ASP.NET application to support logging in via external services.</p>
            </div>
        </EmptyDataTemplate>
    </asp:ListView>
</div>
```

```

using System;
using System.Collections.Generic;
using System.Globalization;
using System.Linq;
using System.Web;
using Microsoft.AspNet.Identity;
using Microsoft.Owin.Security;

namespace Web_App_Master.Account
{
    public partial class OpenAuthProviders : System.Web.UI.UserControl
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            if (IsPostBack)
            {
                var provider = Request.Form["provider"];
                if (provider == null)
                {
                    return;
                }
                // Request a redirect to the external login provider
                string redirectUrl = ResolveUrl(String.Format(CultureInfo.InvariantCulture,
"~/Account/RegisterExternalLogin?{0}={1}&returnUrl={2}", IdentityHelper.ProviderNameKey,
provider, ReturnUrl));
                var properties = new AuthenticationProperties() { RedirectUri = redirectUrl };
                // Add xsrf verification when linking accounts
                if (Context.User.Identity.IsAuthenticated)
                {
                    properties.Dictionary[IdentityHelper.XsrfKey] = Context.User.Identity.GetUserId();
                }
                Context.GetOwinContext().Authentication.Challenge(properties, provider);
                Response.StatusCode = 401;
                Response.End();
            }
        }

        public string ReturnUrl { get; set; }

        public IEnumerable<string> GetProviderNames()
        {
            return Context.GetOwinContext().Authentication.GetExternalAuthenticationTypes().Select(t

```

```
=> t.AuthenticationType);  
    }  
  }  
}
```

```
<%@ Page Title="Out Cart" Language="C#" MasterPageFile="~/Site.Master"
AutoEventWireup="true" CodeBehind="OutCart.aspx.cs"
Inherits="Web_App_Master.Account.Checkout" %>
<asp:Content ID="OutCartContent" ContentPlaceHolderID="MainContent" runat="server">
```

```
    <nav class="navbar navbar-inverse bg-grayDark" style="margin-top:0px !important;
height:auto; width:100%; left:0px!important; border-radius:0px!important; position:fixed !important;
z-index:25000;">
```

```
        <asp:PlaceHolder ID="ApplyChangesButton" runat="server" Visible="true">
            <ul class="nav navbar-nav navbar-right starrag-menu">
                <li>
                    <asp:LinkButton runat="server" OnClick="ContinueToCheckOutBtn_Click"
ID="ContinueToCheckOutBtn" ClientIDMode="Static" style="vertical-align:bottom; text-
decoration:none"><strong>Continue To Shipping&nbsp;&nbsp;</strong><i style="vertical-align:top"
class="glyphicon glyphicon-arrow-right md-glyph"></i></asp:LinkButton>
```

```
                </li>
```

```
            </ul>
```

```
        </asp:PlaceHolder>
```

```
    </nav>
```

```
<div class="bg-metro" style="padding-top:60px !important"></div>
```

```
    <div class="row">
```

```
        <div class="col-md-3">
```

```
            <div class="awp_box rounded bg-sg-title shadow">
```

```
                <div class="awp_box_title bg-sg-title">
```

```
                    <span class="fg-white shadow-metro-black">Options</span></div>
```

```
                <div class="awp_box_content bg-sg-box" style="height:auto !important; vertical-
align:middle; text-align:left !important;">
```

```
                    <h4 class="fg-white shadow-metro-black">Customer</h4>
```

```
                    <asp:PlaceHolder ID="CustomCustomerPlaceHolder" ClientIDMode="Static"
runat="server" Visible="false">
```

```
                        <span class="fg-white shadow-metro-black"><asp:Literal
ID="CustomCustomer" runat="server"></asp:Literal></span>
```

```
                    </asp:PlaceHolder>
```

```
                    <asp:DropDownList Width="100%" ClientIDMode="Static"
```

```
ID='checkout_ShipTo' AppendDataBoundItems="true" runat="server" CssClass="form-control">
```

```
                        <asp:ListItem Text="--Select One--" Value="" />
```

```
                    </asp:DropDownList>
```



```

        <hr />
        <h4 class="fg-white shadow-metro-black">Engineer</h4>
        <asp:PlaceHolder ID ="CustomEngineerPlaceHolder" ClientIDMode="Static"
runat="server" Visible="false">
            <span class=" fg-white shadow-metro-black"><asp:Literal
ID="CustomEngineer" runat="server"></asp:Literal></span>

        </asp:PlaceHolder>
        <asp:DropDownList Width="100%" ClientIDMode="Static"
ID='checkout_ServiceEngineer' AppendDataBoundItems="true" runat="server"  CssClass=" form-
control">
            <asp:ListItem Text="--Select One--" Value="" />
        </asp:DropDownList>
        <hr />
        <h4 class="fg-white shadow-metro-black">User</h4>
        <asp:DropDownList Width="100%" ClientIDMode="Static"
ID='checkout_ShippingPerson' AppendDataBoundItems="true" runat="server"  CssClass=" form-
control">
            <asp:ListItem Text="--Select One--" Value="" />
        </asp:DropDownList>
        <hr />
        <h4 class="fg-white shadow-metro-black">Order Number</h4>

        <div class="">
            <input runat="server" id="checkout_OrderNumber" type="text" class="form-
control" placeholder="Order Number" />
        </div>
    </div>
</div>
</div>
</div>

```

```

        <div class="col-md-9">
        <div class="awp_box rounded bg-sg-title shadow">
            <div class="awp_box_title bg-sg-title">
                <span class="fg-white shadow-metro-black">Items:</span></div>
            <div class="awp_box_content bg-sg-box">
                <div class="row">
                    <asp:UpdatePanel runat="server" ID="OutCartUpdatePanel"

```

```

ClientIDMode="Static" ChildrenAsTriggers="true" UpdateMode="Conditional">
    <ContentTemplate>
        <asp:Repeater runat="server" ID ="FinalCheckoutRepeater"
ClientIDMode="Static" OnItemCommand="FinalCheckoutRepeater_ItemCommand" >
            <ItemTemplate>
                <div class="file " title=" draggable="true"
onclick="BarcodeScanned('<## Eval("AssetNumber")%>', '<## Eval("IsHistoryItem")%>', '<##
Eval("DateShippedTicks")%>');">
                    <div id =" file-color " class="file-color <## Eval("Color")
%>" style="width:100%; height:100%;"></div>

                    <div class="i-icon">
                        <img style="box-shadow: rgba(0, 0, 0, 0.70) 0px 0px 10px;"
src='<## Eval("FirstImage")+ ".r?w=133&h=100" %>' />
                    </div>

                    <div runat="server" class="i-title"><strong class="fg-white
shadow-metro-black"><## Eval("AssetName") %></strong> </div>

                    <div class="i-check">
                        <span id="InOutIndicator" runat="server" class="edge-badge fg-
white shadow-metro-black icon-badge " style="z-index:3 !important;"><strong><##
Eval("AssetNumber")%></strong></span></div>
                        <div runat="server" id="DownloadLink" class="i-download"
style="margin-top:-4px;padding-left:2px" onclick="event.stopPropagation();"> <asp:LinkButton
ID="RemoveFromCheckOutButton" CommandName="Delete" CommandArgument='<##
Eval("AssetNumber")%>' runat="server" > <i title="Add To Task List" id="dl_link" style="font-
size:1.5em;" class="glyphicon glyphicon-remove c-red av-hand-cursor"></i></asp:LinkButton>

                    </div>

                </div>

            </ItemTemplate>
        </asp:Repeater>
    </ContentTemplate>

</asp:UpdatePanel>

</div>
</div>

```

</div>

</div>

</div>

</asp:Content>

```

using Helpers;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace Web_App_Master.Account
{
    public partial class Checkout : System.Web.UI.Page
    {
        //Master Event Wire Up
        protected void Page_PreInit(object sender, EventArgs e)
        {
            this.SiteMaster().OnPanelUpdate += Checkout_OnPanelUpdate;
        }
        private void Checkout_OnPanelUpdate(object sender, UpdateRequestEvent e)
        {
            FinalCheckoutRepeater.DataBind();
            OutCartUpdatePanel.Update();
        }

        public string Customer = null;
        public string Shipper = null;
        public string Engineer = null;
        public string Ordernumber = null;
        public List<Asset> CheckoutItems = null;
        public PendingTransaction Pending = null;
        protected void Page_Load(object sender, EventArgs e)
        {
            this.PreInit += Page_PreInit;
            Button btn = (Button)this.Master.FindControl("UpdateAllCarts");
            AsyncPostBackTrigger Trigger1 = new AsyncPostBackTrigger();
            Trigger1.ControlID = btn.ID;
            Trigger1.EventName = "Click";
            if (OutCartUpdatePanel.Triggers.Count==1)
                OutCartUpdatePanel.Triggers.Add(Trigger1);

            if (!IsPostBack)

```

```

{
    Session["IsPending"] = "false";
    if (!User.Identity.IsAuthenticated)
    {
        Response.Redirect("/Account/Login");
    }
    try
    {
        var TransactionID = Request.QueryString["pend"];
        Session["TransactionID"] = TransactionID;
        if (TransactionID != null)
        {
            Pending= Web_App_Master.Pull.Transaction(TransactionID);
            Session["Pending"] = Pending;
            Session["IsPending"] = "true";
        }
    }
    catch (Exception) { }
    if (Pending != null)
    {
        try
        {
            //build asset list
            var all = AssetController.GetAllAssets();
            List<Asset> list = new List<Asset>();
            foreach (var item in Pending.Assets)
            {
                list.Add((from a in all where a.AssetNumber == item select a).First());
            }
            BindCheckout(list);
            Session["CheckOut"] = list;
            //TODO: FIX THIS SHIT ass find of names in shipping persons
            var user = Page.User.Identity.Name;
            var split = user.Split('@');
            user = split[0];
            split = user.Split('.');
            foreach (var s in split)
            {
                foreach (var shipper in Global.Library.Settings.ShippingPersons)
                {
                    if (shipper.Name.ToLower().Contains(s.ToLower()) ||

```

```

shipper.Email.ToLower().Contains(s.ToLower()))
    {
        Shipper = shipper.Name;
    }
}
}
}
catch { }
checkout_ShippingPerson.Text = Shipper;
if (Pending.Address!=null)
{
    var existing = (from a in Global.Library.Settings.Customers where
Pending.Address.Address == a.Address select a).ToList();
    if (existing.Count>0)
    {
        checkout_ShipTo.Text = existing[0].CompanyName;
    }
    else
    {
        //add to global and save
        Global.Library.Settings.Customers.Add(Pending.Address);
        Push.Globals();
        CustomCustomerPlaceholder.Visible = true;
        CustomCustomer.Text = Pending.Address.CompanyName;
        checkout_ShipTo.Visible = false;
    }
}
checkout_OrderNumber.Value = Pending.OrderNumber;
var engineer = (from a in Global.Library.Settings.ServiceEngineers where a.Email ==
Pending.Name select a).ToList();
if (engineer.Count > 0)
{
    checkout_ServiceEngineer.Text = engineer[0].Name;
}
else
{
    CustomEngineerPlaceholder.Visible = true;
    CustomEngineer.Text = Pending.Name;
    checkout_ServiceEngineer.Visible = false;
}
}

```

```

else
{
    BindCheckout();
    Customer = Session["Customer"] as string;
    Shipper = Session["Shipper"] as string;
    Engineer = Session["Engineer"] as string;
    Ordernumber = Session["Ordernumber"] as string;
    checkout_OrderNumber.Value = Ordernumber;
    checkout_ServiceEngineer.Text = Engineer;

    try
    {
        //TODO: FIX THIS SHIT ass find of names in shipping persons
        string usershipper = Shipper;
        var user = Page.User.Identity.Name;
        var split = user.Split('@');
        user = split[0];
        split = user.Split('.');
        foreach (var s in split)
        {
            foreach (var shipper in Global.Library.Settings.ShippingPersons)
            {
                if (shipper.Name.ToLower().Contains(s.ToLower()) ||
shipper.Email.ToLower().Contains(s.ToLower()))
                {
                    Shipper = shipper.Name;
                }
            }
        }
    }
    catch { }
    checkout_ShippingPerson.Text = Shipper;

    checkout_ShipTo.Text = Customer;
    checkout_ShipTo.Focus();

}
}

}

public void BindCheckout(List<Asset> datasource=null)

```

```

{
    if (datasource==null)
    {
        var checkout = HttpContext.Current.Session["CheckOut"] as List<Asset>;
        if (checkout == null) checkout = new List<Asset>();
        FinalCheckoutRepeater.DataSource = checkout;
        FinalCheckoutRepeater.DataBind();
    }
    else
    {
        FinalCheckoutRepeater.DataSource = datasource;
        FinalCheckoutRepeater.DataBind();
    }

    checkout_ServiceEngineer.DataSource = GetEngineerNames();
    checkout_ServiceEngineer.DataBind();

    checkout_ShippingPerson.DataSource = GetShippingNames();
    checkout_ShippingPerson.DataBind();

    checkout_ShipTo.DataSource = GetShipToNames();
    checkout_ShipTo.DataBind();
}

public List<string> GetShipToNames()
{
    var names = (from ship in Global.Library.Settings.Customers orderby ship.CompanyName
select ship.CompanyName).ToList();
    return names;
}

public List<string> GetEngineerNames()
{
    var names = (from ship in Global.Library.Settings.ServiceEngineers orderby ship.Name
select ship.Name).ToList();
    return names;
}

public List<string> GetShippingNames()
{
    try
    {
        var names = (from ship in Global.Library.Settings.ShippingPersons orderby ship.Name
select ship.Name).ToList();

```



```

        return names;
    }
    catch { return new List<string>(); }
}

```

```

protected void ContinueToCheckOutBtn_Click(object sender, EventArgs e)
{

```

```

    Session["Customer"] = (checkout_ShipTo.Text);
    Session["Shipper"] = (checkout_ShippingPerson.Text);
    Session["Engineer"] = (checkout_ServiceEngineer.Text);
    Session["Ordernumber"] = (checkout_OrderNumber.Value);
    if ((Session["CheckOut"]) == null) return;
    if ((Session["CheckOut"] as List<Asset>).Count == 0) return;

```

```

    //if it is a pending need to remove from sql now

```

```

    if (Session["IsPending"] as string == "true")
    {

```

```

        //AssetController.RemoveTransaction(Pending.TransactionID);

```

```

        Context.Response.Redirect("/Account/Checkout.aspx?tid="+ Session["TransactionID"]
as string);

```

```

    }

```

```

    else

```

```

    {

```

```

        Context.Response.Redirect("/Account/Checkout.aspx");

```

```

    }

```

```

}

```

```

protected void FinalCheckoutRepeater_ItemCommand(object source,
RepeaterCommandEventArgs e)

```

```

{

```

```

    if (e.CommandName.ToLower()=="delete")

```

```

    {

```

```

        Asset rem = null;

```

```

        foreach(var ass in Session["CheckOut"] as List<Asset>)

```

```

        {

```

```

            if (ass.AssetId== e.CommandArgument as string)

```

```

            {

```

```

                rem = ass;

```

```

            }

```

```
    }  
    try  
    {  
        (Session["CheckOut"] as List<Asset>).Remove(rem);  
    }  
    catch { }  
    this.UpdateAll();  
}  
  
}  
  
}
```

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace Asset_Managment
{
    public partial class PackingListForm : Form
    {
        public PackingListForm()
        {
            InitializeComponent();
        }

        private void groupBox1_Enter(object sender, EventArgs e)
        {

        }

        private void tableLayoutPanel2_Paint(object sender, PaintEventArgs e)
        {

        }
    }
}
```

```

namespace Asset_Managment
{
    partial class PackingListForm
    {
        /// <summary>
        /// Required designer variable.
        /// </summary>
        private System.ComponentModel.IContainer components = null;

        /// <summary>
        /// Clean up any resources being used.
        /// </summary>
        /// <param name="disposing">true if managed resources should be disposed; otherwise,
false.</param>
        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
            base.Dispose(disposing);
        }

        #region Windows Form Designer generated code

        /// <summary>
        /// Required method for Designer support - do not modify
        /// the contents of this method with the code editor.
        /// </summary>
        private void InitializeComponent()
        {
            this.tableLayoutPanel1 = new System.Windows.Forms.TableLayoutPanel();
            this.groupBox8 = new System.Windows.Forms.GroupBox();
            this.po = new System.Windows.Forms.Label();
            this.groupBox1 = new System.Windows.Forms.GroupBox();
            this.to = new System.Windows.Forms.Label();
            this.groupBox2 = new System.Windows.Forms.GroupBox();
            this.Weight = new System.Windows.Forms.Label();
            this.OrderNumber = new System.Windows.Forms.Label();
            this.groupBox3 = new System.Windows.Forms.GroupBox();
            this.from = new System.Windows.Forms.Label();
            this.tableLayoutPanel2 = new System.Windows.Forms.TableLayoutPanel();

```

```
this.g12 = new System.Windows.Forms.Label();
this.p12 = new System.Windows.Forms.Label();
this.d12 = new System.Windows.Forms.Label();
this.q11 = new System.Windows.Forms.Label();
this.p11 = new System.Windows.Forms.Label();
this.d11 = new System.Windows.Forms.Label();
this.q10 = new System.Windows.Forms.Label();
this.p10 = new System.Windows.Forms.Label();
this.d10 = new System.Windows.Forms.Label();
this.q9 = new System.Windows.Forms.Label();
this.p9 = new System.Windows.Forms.Label();
this.d9 = new System.Windows.Forms.Label();
this.q8 = new System.Windows.Forms.Label();
this.p8 = new System.Windows.Forms.Label();
this.d8 = new System.Windows.Forms.Label();
this.q7 = new System.Windows.Forms.Label();
this.p7 = new System.Windows.Forms.Label();
this.d7 = new System.Windows.Forms.Label();
this.q6 = new System.Windows.Forms.Label();
this.p6 = new System.Windows.Forms.Label();
this.d6 = new System.Windows.Forms.Label();
this.q5 = new System.Windows.Forms.Label();
this.p5 = new System.Windows.Forms.Label();
this.d5 = new System.Windows.Forms.Label();
this.q4 = new System.Windows.Forms.Label();
this.p4 = new System.Windows.Forms.Label();
this.d4 = new System.Windows.Forms.Label();
this.q3 = new System.Windows.Forms.Label();
this.p3 = new System.Windows.Forms.Label();
this.d3 = new System.Windows.Forms.Label();
this.q2 = new System.Windows.Forms.Label();
this.p2 = new System.Windows.Forms.Label();
this.d2 = new System.Windows.Forms.Label();
this.q1 = new System.Windows.Forms.Label();
this.p1 = new System.Windows.Forms.Label();
this.label1 = new System.Windows.Forms.Label();
this.label2 = new System.Windows.Forms.Label();
this.label3 = new System.Windows.Forms.Label();
this.d1 = new System.Windows.Forms.Label();
this.tableLayoutPanel3 = new System.Windows.Forms.TableLayoutPanel();
this.groupBox5 = new System.Windows.Forms.GroupBox();
this.signature = new System.Windows.Forms.Label();
```

```

this.groupBox6 = new System.Windows.Forms.GroupBox();
this.tracking = new System.Windows.Forms.Label();
this.groupBox7 = new System.Windows.Forms.GroupBox();
this.shippingbottom = new System.Windows.Forms.Label();
this.pictureBox1 = new System.Windows.Forms.PictureBox();
this.groupBox4 = new System.Windows.Forms.GroupBox();
this.attn = new System.Windows.Forms.Label();
this.tableLayoutPanel1.SuspendLayout();
this.groupBox8.SuspendLayout();
this.groupBox1.SuspendLayout();
this.groupBox2.SuspendLayout();
this.groupBox3.SuspendLayout();
this.tableLayoutPanel2.SuspendLayout();
this.tableLayoutPanel3.SuspendLayout();
this.groupBox5.SuspendLayout();
this.groupBox6.SuspendLayout();
this.groupBox7.SuspendLayout();
((System.ComponentModel.ISupportInitialize)(this.pictureBox1)).BeginInit();
this.groupBox4.SuspendLayout();
this.SuspendLayout();
//
// tableLayoutPanel1
//
this.tableLayoutPanel1.BackColor = System.Drawing.Color.White;
this.tableLayoutPanel1.ColumnCount = 3;
this.tableLayoutPanel1.ColumnStyles.Add(new
System.Windows.Forms.ColumnStyle(System.Windows.Forms.SizeType.Percent, 49.93447F));
this.tableLayoutPanel1.ColumnStyles.Add(new
System.Windows.Forms.ColumnStyle(System.Windows.Forms.SizeType.Percent, 25.03276F));
this.tableLayoutPanel1.ColumnStyles.Add(new
System.Windows.Forms.ColumnStyle(System.Windows.Forms.SizeType.Percent, 25.03276F));
this.tableLayoutPanel1.Controls.Add(this.groupBox8, 1, 2);
this.tableLayoutPanel1.Controls.Add(this.groupBox1, 0, 1);
this.tableLayoutPanel1.Controls.Add(this.groupBox2, 0, 0);
this.tableLayoutPanel1.Controls.Add(this.groupBox3, 1, 1);
this.tableLayoutPanel1.Controls.Add(this.tableLayoutPanel2, 0, 3);
this.tableLayoutPanel1.Controls.Add(this.tableLayoutPanel3, 0, 4);
this.tableLayoutPanel1.Controls.Add(this.pictureBox1, 1, 4);
this.tableLayoutPanel1.Controls.Add(this.groupBox4, 0, 2);
this.tableLayoutPanel1.Location = new System.Drawing.Point(128, -322);
this.tableLayoutPanel1.Name = "tableLayoutPanel1";
this.tableLayoutPanel1.RowCount = 5;

```

```

        this.tableLayoutPanel1.RowStyle.Add(new
System.Windows.Forms.RowStyle(System.Windows.Forms.SizeType.Percent, 7.169811F));
        this.tableLayoutPanel1.RowStyle.Add(new
System.Windows.Forms.RowStyle(System.Windows.Forms.SizeType.Percent, 24.33962F));
        this.tableLayoutPanel1.RowStyle.Add(new
System.Windows.Forms.RowStyle(System.Windows.Forms.SizeType.Percent, 6.132075F));
        this.tableLayoutPanel1.RowStyle.Add(new
System.Windows.Forms.RowStyle(System.Windows.Forms.SizeType.Percent, 38.20755F));
        this.tableLayoutPanel1.RowStyle.Add(new
System.Windows.Forms.RowStyle(System.Windows.Forms.SizeType.Percent, 24.0566F));
        this.tableLayoutPanel1.RowStyle.Add(new
System.Windows.Forms.RowStyle(System.Windows.Forms.SizeType.Absolute, 20F));
        this.tableLayoutPanel1.Size = new System.Drawing.Size(891, 1060);
        this.tableLayoutPanel1.TabIndex = 2;
//
// groupBox8
//
this.tableLayoutPanel1.SetColumnSpan(this.groupBox8, 2);
this.groupBox8.Controls.Add(this.po);
this.groupBox8.Dock = System.Windows.Forms.DockStyle.Fill;
this.groupBox8.Location = new System.Drawing.Point(447, 337);
this.groupBox8.Name = "groupBox8";
this.groupBox8.Size = new System.Drawing.Size(441, 59);
this.groupBox8.TabIndex = 10;
this.groupBox8.TabStop = false;
this.groupBox8.Text = "PO:";
//
// po
//
this.po.AutoSize = true;
this.po.Location = new System.Drawing.Point(201, 35);
this.po.Name = "po";
this.po.Size = new System.Drawing.Size(39, 13);
this.po.TabIndex = 3;
this.po.Text = " ";
//
// groupBox1
//
this.groupBox1.BackColor = System.Drawing.Color.White;
this.groupBox1.Controls.Add(this.to);
this.groupBox1.Dock = System.Windows.Forms.DockStyle.Fill;
this.groupBox1.Font = new System.Drawing.Font("Microsoft Sans Serif", 12F,

```

```

System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((byte)(0)));
    this.groupBox1.Location = new System.Drawing.Point(3, 79);
    this.groupBox1.Name = "groupBox1";
    this.groupBox1.Size = new System.Drawing.Size(438, 252);
    this.groupBox1.TabIndex = 2;
    this.groupBox1.TabStop = false;
    this.groupBox1.Text = "TO:";
    this.groupBox1.Enter += new System.EventHandler(this.groupBox1_Enter);
    //
    // to
    //
    this.to.AutoSize = true;
    this.to.Font = new System.Drawing.Font("Microsoft Sans Serif", 15.75F,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((byte)(0)));
    this.to.Location = new System.Drawing.Point(6, 22);
    this.to.Name = "to";
    this.to.Size = new System.Drawing.Size(68, 25);
    this.to.TabIndex = 3;
    this.to.Text = "    ";
    //
    // groupBox2
    //
    this.groupBox2.BackColor = System.Drawing.Color.White;
    this.tableLayoutPanel1.SetColumnSpan(this.groupBox2, 3);
    this.groupBox2.Controls.Add(this.Weight);
    this.groupBox2.Controls.Add(this.OrderNumber);
    this.groupBox2.Dock = System.Windows.Forms.DockStyle.Fill;
    this.groupBox2.Font = new System.Drawing.Font("Microsoft Sans Serif", 15.75F,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((byte)(0)));
    this.groupBox2.Location = new System.Drawing.Point(3, 3);
    this.groupBox2.Name = "groupBox2";
    this.groupBox2.Size = new System.Drawing.Size(885, 70);
    this.groupBox2.TabIndex = 3;
    this.groupBox2.TabStop = false;
    this.groupBox2.Text = "PACKING LIST";
    //
    // Weight
    //
    this.Weight.AutoSize = true;
    this.Weight.BackColor = System.Drawing.Color.White;
    this.Weight.Dock = System.Windows.Forms.DockStyle.Left;
    this.Weight.Location = new System.Drawing.Point(3, 27);

```



```

this.Weight.Name = "Weight";
this.Weight.RightToLeft = System.Windows.Forms.RightToLeft.No;
this.Weight.Size = new System.Drawing.Size(92, 25);
this.Weight.TabIndex = 3;
this.Weight.Text = "Weight:";
this.Weight.TextAlign = System.Drawing.ContentAlignment.MiddleRight;
//
// OrderNumber
//
this.OrderNumber.AutoSize = true;
this.OrderNumber.BackColor = System.Drawing.Color.White;
this.OrderNumber.Dock = System.Windows.Forms.DockStyle.Right;
this.OrderNumber.Location = new System.Drawing.Point(791, 27);
this.OrderNumber.Name = "OrderNumber";
this.OrderNumber.RightToLeft = System.Windows.Forms.RightToLeft.No;
this.OrderNumber.Size = new System.Drawing.Size(91, 25);
this.OrderNumber.TabIndex = 2;
this.OrderNumber.Text = "Order #";
this.OrderNumber.TextAlign = System.Drawing.ContentAlignment.MiddleRight;
//
// groupBox3
//
this.groupBox3.BackColor = System.Drawing.Color.White;
this.tableLayoutPanel1.SetColumnSpan(this.groupBox3, 2);
this.groupBox3.Controls.Add(this.from);
this.groupBox3.Dock = System.Windows.Forms.DockStyle.Fill;
this.groupBox3.Font = new System.Drawing.Font("Microsoft Sans Serif", 12F,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((byte)0));
this.groupBox3.Location = new System.Drawing.Point(447, 79);
this.groupBox3.Name = "groupBox3";
this.groupBox3.Size = new System.Drawing.Size(441, 252);
this.groupBox3.TabIndex = 4;
this.groupBox3.TabStop = false;
this.groupBox3.Text = "FROM:";
//
// from
//
this.from.AutoSize = true;
this.from.Font = new System.Drawing.Font("Microsoft Sans Serif", 15.75F,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((byte)0));
this.from.Location = new System.Drawing.Point(6, 22);
this.from.Name = "from";

```

```

this.from.Size = new System.Drawing.Size(68, 25);
this.from.TabIndex = 3;
this.from.Text = "    ";
//
// tableLayoutPanel2
//
this.tableLayoutPanel2.BackColor = System.Drawing.Color.White;
this.tableLayoutPanel2.CellBorderStyle =
System.Windows.Forms.TableLayoutPanelCellBorderStyle.Single;
this.tableLayoutPanel2.ColumnCount = 3;
this.tableLayoutPanel1.SetColumnSpan(this.tableLayoutPanel2, 3);
this.tableLayoutPanel2.ColumnStyles.Add(new
System.Windows.Forms.ColumnStyle(System.Windows.Forms.SizeType.Percent, 49.86808F));
this.tableLayoutPanel2.ColumnStyles.Add(new
System.Windows.Forms.ColumnStyle(System.Windows.Forms.SizeType.Percent, 25.46174F));
this.tableLayoutPanel2.ColumnStyles.Add(new
System.Windows.Forms.ColumnStyle(System.Windows.Forms.SizeType.Percent, 24.67019F));
this.tableLayoutPanel2.Controls.Add(this.g12, 2, 12);
this.tableLayoutPanel2.Controls.Add(this.p12, 1, 12);
this.tableLayoutPanel2.Controls.Add(this.d12, 0, 12);
this.tableLayoutPanel2.Controls.Add(this.q11, 2, 11);
this.tableLayoutPanel2.Controls.Add(this.p11, 1, 11);
this.tableLayoutPanel2.Controls.Add(this.d11, 0, 11);
this.tableLayoutPanel2.Controls.Add(this.q10, 2, 10);
this.tableLayoutPanel2.Controls.Add(this.p10, 1, 10);
this.tableLayoutPanel2.Controls.Add(this.d10, 0, 10);
this.tableLayoutPanel2.Controls.Add(this.q9, 2, 9);
this.tableLayoutPanel2.Controls.Add(this.p9, 1, 9);
this.tableLayoutPanel2.Controls.Add(this.d9, 0, 9);
this.tableLayoutPanel2.Controls.Add(this.q8, 2, 8);
this.tableLayoutPanel2.Controls.Add(this.p8, 1, 8);
this.tableLayoutPanel2.Controls.Add(this.d8, 0, 8);
this.tableLayoutPanel2.Controls.Add(this.q7, 2, 7);
this.tableLayoutPanel2.Controls.Add(this.p7, 1, 7);
this.tableLayoutPanel2.Controls.Add(this.d7, 0, 7);
this.tableLayoutPanel2.Controls.Add(this.q6, 2, 6);
this.tableLayoutPanel2.Controls.Add(this.p6, 1, 6);
this.tableLayoutPanel2.Controls.Add(this.d6, 0, 6);
this.tableLayoutPanel2.Controls.Add(this.q5, 2, 5);
this.tableLayoutPanel2.Controls.Add(this.p5, 1, 5);
this.tableLayoutPanel2.Controls.Add(this.d5, 0, 5);
this.tableLayoutPanel2.Controls.Add(this.q4, 2, 4);

```

[illegible]

```
this.tableLayoutPanel2.RowStyle.Add(new
System.Windows.Forms.RowStyle(System.Windows.Forms.SizeType.Absolute, 20F));
this.tableLayoutPanel2.RowStyle.Add(new
System.Windows.Forms.RowStyle(System.Windows.Forms.SizeType.Absolute, 20F));
this.tableLayoutPanel2.Size = new System.Drawing.Size(885, 399);
this.tableLayoutPanel2.TabIndex = 6;
this.tableLayoutPanel2.Paint += new
System.Windows.Forms.PaintEventHandler(this.tableLayoutPanel2_Paint);
//
// g12
//
this.g12.AutoSize = true;
this.g12.Location = new System.Drawing.Point(669, 253);
this.g12.Name = "g12";
this.g12.Size = new System.Drawing.Size(49, 20);
this.g12.TabIndex = 36;
this.g12.Text = " ";
//
// p12
//
this.p12.AutoSize = true;
this.p12.Location = new System.Drawing.Point(444, 253);
this.p12.Name = "p12";
this.p12.Size = new System.Drawing.Size(49, 20);
this.p12.TabIndex = 35;
this.p12.Text = " ";
//
// d12
//
this.d12.AutoSize = true;
this.d12.Location = new System.Drawing.Point(4, 253);
this.d12.Name = "d12";
this.d12.Size = new System.Drawing.Size(49, 20);
this.d12.TabIndex = 34;
this.d12.Text = " ";
//
// q11
//
this.q11.AutoSize = true;
this.q11.Location = new System.Drawing.Point(669, 232);
this.q11.Name = "q11";
this.q11.Size = new System.Drawing.Size(49, 20);
```

```
this.q11.TabIndex = 33;
this.q11.Text = "    ";
//
// p11
//
this.p11.AutoSize = true;
this.p11.Location = new System.Drawing.Point(444, 232);
this.p11.Name = "p11";
this.p11.Size = new System.Drawing.Size(49, 20);
this.p11.TabIndex = 32;
this.p11.Text = "    ";
//
// d11
//
this.d11.AutoSize = true;
this.d11.Location = new System.Drawing.Point(4, 232);
this.d11.Name = "d11";
this.d11.Size = new System.Drawing.Size(49, 20);
this.d11.TabIndex = 31;
this.d11.Text = "    ";
//
// q10
//
this.q10.AutoSize = true;
this.q10.Location = new System.Drawing.Point(669, 211);
this.q10.Name = "q10";
this.q10.Size = new System.Drawing.Size(49, 20);
this.q10.TabIndex = 30;
this.q10.Text = "    ";
//
// p10
//
this.p10.AutoSize = true;
this.p10.Location = new System.Drawing.Point(444, 211);
this.p10.Name = "p10";
this.p10.Size = new System.Drawing.Size(49, 20);
this.p10.TabIndex = 29;
this.p10.Text = "    ";
//
// d10
//
this.d10.AutoSize = true;
```

```
this.d10.Location = new System.Drawing.Point(4, 211);
this.d10.Name = "d10";
this.d10.Size = new System.Drawing.Size(49, 20);
this.d10.TabIndex = 28;
this.d10.Text = "    ";
//
// q9
//
this.q9.AutoSize = true;
this.q9.Location = new System.Drawing.Point(669, 190);
this.q9.Name = "q9";
this.q9.Size = new System.Drawing.Size(49, 20);
this.q9.TabIndex = 27;
this.q9.Text = "    ";
//
// p9
//
this.p9.AutoSize = true;
this.p9.Location = new System.Drawing.Point(444, 190);
this.p9.Name = "p9";
this.p9.Size = new System.Drawing.Size(49, 20);
this.p9.TabIndex = 26;
this.p9.Text = "    ";
//
// d9
//
this.d9.AutoSize = true;
this.d9.Location = new System.Drawing.Point(4, 190);
this.d9.Name = "d9";
this.d9.Size = new System.Drawing.Size(49, 20);
this.d9.TabIndex = 25;
this.d9.Text = "    ";
//
// q8
//
this.q8.AutoSize = true;
this.q8.Location = new System.Drawing.Point(669, 169);
this.q8.Name = "q8";
this.q8.Size = new System.Drawing.Size(49, 20);
this.q8.TabIndex = 24;
this.q8.Text = "    ";
//
```

```
// p8
//
this.p8.AutoSize = true;
this.p8.Location = new System.Drawing.Point(444, 169);
this.p8.Name = "p8";
this.p8.Size = new System.Drawing.Size(49, 20);
this.p8.TabIndex = 23;
this.p8.Text = "    ";
//
// d8
//
this.d8.AutoSize = true;
this.d8.Location = new System.Drawing.Point(4, 169);
this.d8.Name = "d8";
this.d8.Size = new System.Drawing.Size(49, 20);
this.d8.TabIndex = 22;
this.d8.Text = "    ";
//
// q7
//
this.q7.AutoSize = true;
this.q7.Location = new System.Drawing.Point(669, 148);
this.q7.Name = "q7";
this.q7.Size = new System.Drawing.Size(49, 20);
this.q7.TabIndex = 21;
this.q7.Text = "    ";
//
// p7
//
this.p7.AutoSize = true;
this.p7.Location = new System.Drawing.Point(444, 148);
this.p7.Name = "p7";
this.p7.Size = new System.Drawing.Size(49, 20);
this.p7.TabIndex = 20;
this.p7.Text = "    ";
//
// d7
//
this.d7.AutoSize = true;
this.d7.Location = new System.Drawing.Point(4, 148);
this.d7.Name = "d7";
this.d7.Size = new System.Drawing.Size(49, 20);
```

```
this.d7.TabIndex = 19;
this.d7.Text = "    ";
//
// q6
//
this.q6.AutoSize = true;
this.q6.Location = new System.Drawing.Point(669, 127);
this.q6.Name = "q6";
this.q6.Size = new System.Drawing.Size(49, 20);
this.q6.TabIndex = 18;
this.q6.Text = "    ";
//
// p6
//
this.p6.AutoSize = true;
this.p6.Location = new System.Drawing.Point(444, 127);
this.p6.Name = "p6";
this.p6.Size = new System.Drawing.Size(49, 20);
this.p6.TabIndex = 17;
this.p6.Text = "    ";
//
// d6
//
this.d6.AutoSize = true;
this.d6.Location = new System.Drawing.Point(4, 127);
this.d6.Name = "d6";
this.d6.Size = new System.Drawing.Size(49, 20);
this.d6.TabIndex = 16;
this.d6.Text = "    ";
//
// q5
//
this.q5.AutoSize = true;
this.q5.Location = new System.Drawing.Point(669, 106);
this.q5.Name = "q5";
this.q5.Size = new System.Drawing.Size(49, 20);
this.q5.TabIndex = 15;
this.q5.Text = "    ";
//
// p5
//
this.p5.AutoSize = true;
```



```
this.p5.Location = new System.Drawing.Point(444, 106);
this.p5.Name = "p5";
this.p5.Size = new System.Drawing.Size(49, 20);
this.p5.TabIndex = 14;
this.p5.Text = "    ";
//
// d5
//
this.d5.AutoSize = true;
this.d5.Location = new System.Drawing.Point(4, 106);
this.d5.Name = "d5";
this.d5.Size = new System.Drawing.Size(49, 20);
this.d5.TabIndex = 13;
this.d5.Text = "    ";
//
// q4
//
this.q4.AutoSize = true;
this.q4.Location = new System.Drawing.Point(669, 85);
this.q4.Name = "q4";
this.q4.Size = new System.Drawing.Size(49, 20);
this.q4.TabIndex = 12;
this.q4.Text = "    ";
//
// p4
//
this.p4.AutoSize = true;
this.p4.Location = new System.Drawing.Point(444, 85);
this.p4.Name = "p4";
this.p4.Size = new System.Drawing.Size(49, 20);
this.p4.TabIndex = 11;
this.p4.Text = "    ";
//
// d4
//
this.d4.AutoSize = true;
this.d4.Location = new System.Drawing.Point(4, 85);
this.d4.Name = "d4";
this.d4.Size = new System.Drawing.Size(49, 20);
this.d4.TabIndex = 10;
this.d4.Text = "    ";
//
```

```
// q3
//
this.q3.AutoSize = true;
this.q3.Location = new System.Drawing.Point(669, 64);
this.q3.Name = "q3";
this.q3.Size = new System.Drawing.Size(49, 20);
this.q3.TabIndex = 9;
this.q3.Text = "    ";
//
// p3
//
this.p3.AutoSize = true;
this.p3.Location = new System.Drawing.Point(444, 64);
this.p3.Name = "p3";
this.p3.Size = new System.Drawing.Size(49, 20);
this.p3.TabIndex = 8;
this.p3.Text = "    ";
//
// d3
//
this.d3.AutoSize = true;
this.d3.Location = new System.Drawing.Point(4, 64);
this.d3.Name = "d3";
this.d3.Size = new System.Drawing.Size(49, 20);
this.d3.TabIndex = 7;
this.d3.Text = "    ";
//
// q2
//
this.q2.AutoSize = true;
this.q2.Location = new System.Drawing.Point(669, 43);
this.q2.Name = "q2";
this.q2.Size = new System.Drawing.Size(49, 20);
this.q2.TabIndex = 6;
this.q2.Text = "    ";
//
// p2
//
this.p2.AutoSize = true;
this.p2.Location = new System.Drawing.Point(444, 43);
this.p2.Name = "p2";
this.p2.Size = new System.Drawing.Size(49, 20);
```

```
this.p2.TabIndex = 5;
this.p2.Text = "    ";
//
// d2
//
this.d2.AutoSize = true;
this.d2.Location = new System.Drawing.Point(4, 43);
this.d2.Name = "d2";
this.d2.Size = new System.Drawing.Size(49, 20);
this.d2.TabIndex = 4;
this.d2.Text = "    ";
//
// q1
//
this.q1.AutoSize = true;
this.q1.Location = new System.Drawing.Point(669, 22);
this.q1.Name = "q1";
this.q1.Size = new System.Drawing.Size(49, 20);
this.q1.TabIndex = 3;
this.q1.Text = "    ";
//
// p1
//
this.p1.AutoSize = true;
this.p1.Location = new System.Drawing.Point(444, 22);
this.p1.Name = "p1";
this.p1.Size = new System.Drawing.Size(49, 20);
this.p1.TabIndex = 2;
this.p1.Text = "    ";
//
// label1
//
this.label1.Anchor = System.Windows.Forms.AnchorStyles.Top;
this.label1.AutoSize = true;
this.label1.Location = new System.Drawing.Point(170, 1);
this.label1.Name = "label1";
this.label1.Size = new System.Drawing.Size(100, 20);
this.label1.TabIndex = 0;
this.label1.Text = "Description";
//
// label2
//
```

```

this.label2.Anchor = System.Windows.Forms.AnchorStyles.Top;
this.label2.AutoSize = true;
this.label2.Location = new System.Drawing.Point(524, 1);
this.label2.Name = "label2";
this.label2.Size = new System.Drawing.Size(57, 20);
this.label2.TabIndex = 0;
this.label2.Text = "Part #";
//
// label3
//
this.label3.Anchor = System.Windows.Forms.AnchorStyles.Top;
this.label3.AutoSize = true;
this.label3.Location = new System.Drawing.Point(737, 1);
this.label3.Name = "label3";
this.label3.Size = new System.Drawing.Size(76, 20);
this.label3.TabIndex = 0;
this.label3.Text = "Quantity";
//
// d1
//
this.d1.AutoSize = true;
this.d1.Location = new System.Drawing.Point(4, 22);
this.d1.Name = "d1";
this.d1.Size = new System.Drawing.Size(49, 20);
this.d1.TabIndex = 1;
this.d1.Text = "    ";
//
// tableLayoutPanel3
//
this.tableLayoutPanel3.ColumnCount = 1;
this.tableLayoutPanel3.ColumnStyles.Add(new
System.Windows.Forms.ColumnStyle(System.Windows.Forms.SizeType.Percent, 50F));
this.tableLayoutPanel3.Controls.Add(this.groupBox5, 0, 0);
this.tableLayoutPanel3.Controls.Add(this.groupBox6, 0, 1);
this.tableLayoutPanel3.Controls.Add(this.groupBox7, 0, 2);
this.tableLayoutPanel3.Dock = System.Windows.Forms.DockStyle.Fill;
this.tableLayoutPanel3.Location = new System.Drawing.Point(3, 807);
this.tableLayoutPanel3.Name = "tableLayoutPanel3";
this.tableLayoutPanel3.RowCount = 3;
this.tableLayoutPanel3.RowStyles.Add(new
System.Windows.Forms.RowStyle(System.Windows.Forms.SizeType.Percent, 50F));
this.tableLayoutPanel3.RowStyles.Add(new

```

```
System.Windows.Forms.RowStyle(System.Windows.Forms.SizeType.Percent, 50F));
    this.tableLayoutPanel3.RowStyles.Add(new
System.Windows.Forms.RowStyle(System.Windows.Forms.SizeType.Absolute, 86F));
    this.tableLayoutPanel3.Size = new System.Drawing.Size(438, 250);
    this.tableLayoutPanel3.TabIndex = 7;
    //
    // groupBox5
    //
    this.groupBox5.Controls.Add(this.signature);
    this.groupBox5.Dock = System.Windows.Forms.DockStyle.Fill;
    this.groupBox5.Location = new System.Drawing.Point(3, 3);
    this.groupBox5.Name = "groupBox5";
    this.groupBox5.Size = new System.Drawing.Size(432, 76);
    this.groupBox5.TabIndex = 0;
    this.groupBox5.TabStop = false;
    this.groupBox5.Text = "Shipper Signature";
    //
    // signature
    //
    this.signature.AutoSize = true;
    this.signature.Location = new System.Drawing.Point(197, 32);
    this.signature.Name = "signature";
    this.signature.Size = new System.Drawing.Size(39, 13);
    this.signature.TabIndex = 2;
    this.signature.Text = "    ";
    //
    // groupBox6
    //
    this.groupBox6.Controls.Add(this.tracking);
    this.groupBox6.Dock = System.Windows.Forms.DockStyle.Fill;
    this.groupBox6.Location = new System.Drawing.Point(3, 85);
    this.groupBox6.Name = "groupBox6";
    this.groupBox6.Size = new System.Drawing.Size(432, 76);
    this.groupBox6.TabIndex = 0;
    this.groupBox6.TabStop = false;
    this.groupBox6.Text = "Tracking Number";
    //
    // tracking
    //
    this.tracking.AutoSize = true;
    this.tracking.Location = new System.Drawing.Point(197, 32);
    this.tracking.Name = "tracking";
```

```
this.tracking.Size = new System.Drawing.Size(39, 13);
this.tracking.TabIndex = 2;
this.tracking.Text = "    ";
//
// groupBox7
//
this.groupBox7.Controls.Add(this.shippingbottom);
this.groupBox7.Dock = System.Windows.Forms.DockStyle.Fill;
this.groupBox7.Location = new System.Drawing.Point(3, 167);
this.groupBox7.Name = "groupBox7";
this.groupBox7.Size = new System.Drawing.Size(432, 80);
this.groupBox7.TabIndex = 0;
this.groupBox7.TabStop = false;
this.groupBox7.Text = "Ship Date";
//
// shippingbottom
//
this.shippingbottom.AutoSize = true;
this.shippingbottom.Location = new System.Drawing.Point(197, 34);
this.shippingbottom.Name = "shippingbottom";
this.shippingbottom.Size = new System.Drawing.Size(39, 13);
this.shippingbottom.TabIndex = 2;
this.shippingbottom.Text = "    ";
//
// pictureBox1
//
this.tableLayoutPanel1.SetColumnSpan(this.pictureBox1, 2);
this.pictureBox1.Dock = System.Windows.Forms.DockStyle.Fill;
this.pictureBox1.Image = global::Asset_Managment.Properties.Resources.STARRAG;
this.pictureBox1.Location = new System.Drawing.Point(447, 807);
this.pictureBox1.Name = "pictureBox1";
this.pictureBox1.Size = new System.Drawing.Size(441, 250);
this.pictureBox1.SizeMode = System.Windows.Forms.PictureBoxSizeMode.Zoom;
this.pictureBox1.TabIndex = 8;
this.pictureBox1.TabStop = false;
//
// groupBox4
//
this.groupBox4.Controls.Add(this.attn);
this.groupBox4.Dock = System.Windows.Forms.DockStyle.Fill;
this.groupBox4.Location = new System.Drawing.Point(3, 337);
this.groupBox4.Name = "groupBox4";
```

```
this.groupBox4.Size = new System.Drawing.Size(438, 59);
this.groupBox4.TabIndex = 9;
this.groupBox4.TabStop = false;
this.groupBox4.Text = "ATTN:";
//
// attn
//
this.attn.AutoSize = true;
this.attn.Font = new System.Drawing.Font("Microsoft Sans Serif", 12F,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((byte)(0)));
this.attn.Location = new System.Drawing.Point(8, 16);
this.attn.Name = "attn";
this.attn.Size = new System.Drawing.Size(49, 20);
this.attn.TabIndex = 3;
this.attn.Text = "      ";
//
// PackingListForm
//
this.AutoScaleDimensions = new System.Drawing.SizeF(7F, 13F);
this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
this.ClientSize = new System.Drawing.Size(1148, 742);
this.Controls.Add(this.tableLayoutPanel1);
this.Font = new System.Drawing.Font("Microsoft Sans Serif", 8.25F,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((byte)(0)));
this.Location = new System.Drawing.Point(6000, 6000);
this.Name = "PackingListForm";
this.StartPosition = System.Windows.Forms.FormStartPosition.Manual;
this.Text = "PackingListForm";
this.tableLayoutPanel1.ResumeLayout(false);
this.groupBox8.ResumeLayout(false);
this.groupBox8.PerformLayout();
this.groupBox1.ResumeLayout(false);
this.groupBox1.PerformLayout();
this.groupBox2.ResumeLayout(false);
this.groupBox2.PerformLayout();
this.groupBox3.ResumeLayout(false);
this.groupBox3.PerformLayout();
this.tableLayoutPanel2.ResumeLayout(false);
this.tableLayoutPanel2.PerformLayout();
this.tableLayoutPanel3.ResumeLayout(false);
this.groupBox5.ResumeLayout(false);
this.groupBox5.PerformLayout();
```

```
this.groupBox6.ResumeLayout(false);
this.groupBox6.PerformLayout();
this.groupBox7.ResumeLayout(false);
this.groupBox7.PerformLayout();
((System.ComponentModel.ISupportInitialize)(this.pictureBox1)).EndInit();
this.groupBox4.ResumeLayout(false);
this.groupBox4.PerformLayout();
this.ResumeLayout(false);

}
```

#endregion

```
public System.Windows.Forms.TableLayoutPanelPanel tableLayoutPanel1;
public System.Windows.Forms.GroupBox groupBox1;
public System.Windows.Forms.GroupBox groupBox2;
public System.Windows.Forms.Label OrderNumber;
public System.Windows.Forms.GroupBox groupBox3;
public System.Windows.Forms.TableLayoutPanelPanel tableLayoutPanel2;
public System.Windows.Forms.TableLayoutPanelPanel tableLayoutPanel3;
public System.Windows.Forms.GroupBox groupBox5;
public System.Windows.Forms.GroupBox groupBox6;
public System.Windows.Forms.GroupBox groupBox7;
public System.Windows.Forms.PictureBox pictureBox1;
public System.Windows.Forms.Label label1;
public System.Windows.Forms.Label label2;
public System.Windows.Forms.Label label3;
public System.Windows.Forms.GroupBox groupBox8;
public System.Windows.Forms.Label po;
public System.Windows.Forms.Label to;
public System.Windows.Forms.Label from;
public System.Windows.Forms.Label g12;
public System.Windows.Forms.Label p12;
public System.Windows.Forms.Label d12;
public System.Windows.Forms.Label q11;
public System.Windows.Forms.Label p11;
public System.Windows.Forms.Label d11;
public System.Windows.Forms.Label q10;
public System.Windows.Forms.Label p10;
public System.Windows.Forms.Label d10;
public System.Windows.Forms.Label q9;
public System.Windows.Forms.Label p9;
```



```
public System.Windows.Forms.Label d9;  
public System.Windows.Forms.Label q8;  
public System.Windows.Forms.Label p8;  
public System.Windows.Forms.Label d8;  
public System.Windows.Forms.Label q7;  
public System.Windows.Forms.Label p7;  
public System.Windows.Forms.Label d7;  
public System.Windows.Forms.Label q6;  
public System.Windows.Forms.Label p6;  
public System.Windows.Forms.Label d6;  
public System.Windows.Forms.Label q5;  
public System.Windows.Forms.Label p5;  
public System.Windows.Forms.Label d5;  
public System.Windows.Forms.Label q4;  
public System.Windows.Forms.Label p4;  
public System.Windows.Forms.Label d4;  
public System.Windows.Forms.Label q3;  
public System.Windows.Forms.Label p3;  
public System.Windows.Forms.Label d3;  
public System.Windows.Forms.Label q2;  
public System.Windows.Forms.Label p2;  
public System.Windows.Forms.Label d2;  
public System.Windows.Forms.Label q1;  
public System.Windows.Forms.Label p1;  
public System.Windows.Forms.Label d1;  
public System.Windows.Forms.Label signature;  
public System.Windows.Forms.Label tracking;  
public System.Windows.Forms.Label shippingbottom;  
public System.Windows.Forms.GroupBox groupBox4;  
public System.Windows.Forms.Label attn;  
public System.Windows.Forms.Label Weight;  
}  
}
```

```
<%@ Page Title="PDF" Language="C#" MasterPageFile="~/Site.Master" AutoEventWireup="true"
CodeBehind="PdfViewer.aspx.cs" Inherits="Web_App_Master.Account.PdfViewer" %>
<asp:Content ID="PdfViewer" ContentPlaceHolderID="MainContent" runat="server">
```

```
<script type="text/javascript">
```

```
$(document).ready(function () {
```

```
function calcIframeDivHeight() {
```

```
    var newHeight = 0;
```

```
    var myViewHeight = jQuery(".main-content").height();
```

```
    newHeight = myViewHeight - 160;
```

```
    $("#IframeDiv").height(newHeight);
```

```
    $("#MainContent_ReportFrame").height(newHeight);
```

```
    $("#MainContent_ReportFrame").width($("#IframeDiv").width());
```

```
}
```

```
// run on page load
```

```
calcIframeDivHeight();
```

```
// run on window resize event
```

```
$(window).resize(function () {
```

```
    calcIframeDivHeight();
```

```
});
```

```
});
```

```
</script>
```

```
<div class="row">
```

```
    <div class="col-md-12">
```

```
        <!-- <span class="awp-save-btn bg-green fg-white shadow mif-ani-flash "
onclick="printpackingslip()"><i title="Print" style="font-size:2em; vertical-align:top" class="mif-
printer av-hand-cursor fg-white shadow-metro-black"></i></span>-->
```

```
        <div class="awp_box rounded bg-metro-dark shadow">
```

```
            <div class="awp_box_title bg-metro-dark">
```

```
                <span class="fg-white shadow-metro-black"><asp:Literal runat="server"
ID="PdfTitle" ></asp:Literal></span>
```

```
            </div>
```

```
            <div class="awp_box_content bg-metro-light">
```

```
                <div id="ReceivingHidden" style="display:none"><asp:Literal ID="ReceivingLink"
runat="server"></asp:Literal> </div>
```

```
<div id="IframeDiv" style="min-height:100px;height:300px;">
  <!-- <iframe frameborder="0" style="position:absolute; min-height:100px;"
id="ReportFrame" runat="server" src="#" ></iframe>-->
    <object id="PdfEmbeddedViewer"
data="/Account/Labels/1Z9A14T70291180501.pdf" type="application/pdf" width="100%"
height="100%">
      <p>This browser does not support PDFs. Please download the PDF to view it:
<a href="/pdf/sample-3pp.pdf">Download PDF</a>.</p>
    </object>
  </div>
</div>
</div>
</div>
</div>

</asp:Content>
```

```

using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace Web_App_Master.Account
{
    public partial class PdfViewer : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            var a = Server.UrlEncode("/Account/Receiving/1-19-2018_5-58-48_PM_11111.pdf");
            if (!IsPostBack)
            {
                try
                {
                    var b = Request["file"];
                    PdfTitle.Text = Path.GetFileName(b);
                    ReportFrame.Src = b;
                }
                catch { }
            }
        }
    }
}

```

```
<%@ Page Title Language="C#" MasterPageFile="~/Site.Master" AutoEventWireup="true"
CodeBehind="PrintBarcodes.aspx.cs" Inherits="Web_App_Master.Account.PrintBarcodes" %>
<asp:Content ID="Content1" ContentPlaceHolderID="MainContent" runat="server">
```

```
<!--menu
```

```
<div class="app-bar shadow-bottom" style="margin-top:0px !important; left:0px!important;
position:fixed !important; z-index:25000;">
```

```
<ul class="app-bar-menu">
```

```
<li><h1></h1> </li>
```

```
<li>
```

```
<a href="#" class="dropdown-toggle">View</a>
```

```
<ul class="d-menu" data-role="dropdown">
```

```
<li><a href="#" onclick="ChangeAssetView('list'); ">Default</a></li>
```

```
<li><a href="#" onclick="ChangeAssetViewListType('list-type-icons');">Icon</a></li>
```

```
<li><a href="#" onclick="ChangeAssetViewListType('list-type-tiles');">Detail</a></li>
```

```
<li><a href="#" onclick="ChangeAssetViewListType('list-type-listing');">Listing</a></li>
```

```
<li><a href="#" onclick="ChangeAssetView('smtile');" >List Tile</a></li>
```

```
<li><a href="#" onclick="ChangeAssetView('mdtile');" >Medium Tile</a></li>
```

```
<li><a href="#" onclick="ChangeAssetView('lgtile');" >large Tile</a></li>
```

```
</ul>
```

```
</li>
```

```
<li>
```

```
<div class="input-control text" data-role="input" style="width:250px !important;">
```

```
<asp:TextBox ID="avSearchString" runat="server" ClientIDMode="Static"
```

```
CssClass="text"></asp:TextBox>
```

```
</div>
```

```
</li>
```

```
<li>
```

```
<asp:LinkButton ID ="AssetSearchBtn" ClientIDMode="Static" runat="server" ><span
class="glyphicon glyphicon-search"></span></asp:LinkButton>
```

```
</li>
```

```
<li>
```

```
<a href="#" style="text-decoration:none" class="dropdown-toggle">&nbsp;<span
title="Search Filter" class="mif-settings-ethernet"></span></a>
```

```
<ul class="d-menu" data-role="dropdown">
```

```
<li><a id ="avAssetSearchBtn" runat="server" onclick="SetSearchType('asset')" >Asset
```

```

Search</a> </li>
    <li><a id ="avHistorySearchBtn" runat="server" onclick="SetSearchType('history')"
>History Search</a> </li>

    </ul>
</li>
<li>
    <a href="#" style="text-decoration:none" class="dropdown-toggle">&nbsp;<span title="View
Filter" class="glyphicon glyphicon-filter"></span></a>
    <ul class="d-menu" data-role="dropdown">
        <li><asp:LinkButton ID ="ViewAll" ClientIDMode="Static" runat="server" >View
All</asp:LinkButton> </li>
        <li><asp:LinkButton ID ="FilterIsOut" ClientIDMode="Static" runat="server" >Checked
Out Only</asp:LinkButton> </li>
        <li><asp:LinkButton ID ="FilterIsIn" ClientIDMode="Static" runat="server" >Checked In
Only</asp:LinkButton> </li>
        <li><asp:LinkButton ID ="FilterIsDamaged" ClientIDMode="Static" runat="server"
>Damaged Only</asp:LinkButton> </li>
        <li><asp:LinkButton ID ="FilterOnHold" ClientIDMode="Static" runat="server" >On Hold
Only</asp:LinkButton> </li>
        <li><asp:LinkButton ID ="FilterCalibrated" ClientIDMode="Static" runat="server"
>Calibrated Only</asp:LinkButton> </li>

    </ul>
</li>

</ul>

</div>
--><!--updatepanel-->
    <div class="row">

        <!--REPEATER-->

<script type="text/javascript">

    $(document).ready(function () {

```

```
function calcIframeDivHeight() {
    var newHeight = 0;
    var myViewHeight = jQuery(".main-content").height();
    newHeight = myViewHeight - 105;
    $("#PrintoutDiv").height(newHeight);
    $("#PrintControlDiv").height(newHeight);
    // $("#MainContent_ReportFrame").height(newHeight);
    // $("#MainContent_ReportFrame").width($("#PrintoutDiv").width());
}
```

```
// run on page load
calcIframeDivHeight();
```

```
// run on window resize event
$(window).resize(function () {
    calcIframeDivHeight();
});
```

```
});
```

```
</script>
```

```
<div class="col-md-3">
```

```
    <!-- <span class="awp-save-btn bg-green fg-white shadow mif-ani-flash "
onclick="printpackingslip()"><i title="Print" style="font-size:2em; vertical-align:top" class="mif-
printer av-hand-cursor fg-white shadow-metro-black"></i></span>-->
```

```
<div class="awp_box rounded bg-sg-title shadow">
```

```
    <span class="awp-bar " style="padding-bottom:-10px!important;">
```

```
        <span class="awp-bar-btn bg-green fg-white shadow">
```

```
            <asp:button runat="server" ClientIDMode="Static" ID="hiddenModeSwitch"
OnClick="hiddenModeSwitch_Click" style="display:none;"></asp:button>
```

```
            <label class="awp-switch" title="Toggle Text/Barcode Label" style="padding-
left:2px!important;">
```

```
                <input id="TestModeSwitch" type="checkbox" checked="checked"
onchange="ChangeLabelDisplay()">
```

```
                <span class="check"></span>
```

```
            </label>
```

```
        </span>
```

```
    </span>
```

```
    <div class="awp_box_title bg-sg-title">
```

```
        <span class="fg-white shadow-metro-black">Label Options</span>
```

```
    </div>
```

```

<div class="awp_box_content bg-sg-box fg-white">
  <div id="PrintControlDiv" style="min-height:500px;height:300px;">
    <span><strong>Start Position</strong></span>
    <asp:TextBox runat="server" ID="StartPos" CssClass="form-control"
TextMode="Number" Text="1" />
    <asp:RequiredFieldValidator runat="server" ControlToValidate="StartPos"
    CssClass="text-danger" ErrorMessage="Must be a number." />
    <div class="awp_box rounded bg-sg-title shadow">
      <span class="awp-bar ">
        <span class="awp-bar-btn bg-green fg-white shadow"><asp:LinkButton
runat="server" ID="SelectAllBtn" OnClick="SelectAllBtn_Click"> <i title="Select All"
class="glyphicon glyphicon-check fg-white"></i></asp:LinkButton></span>
      </span>
      <div class="awp_box_title bg-sg-title">
        <span class="fg-white shadow-metro-black">Assets</span>
      </div>
      <div class="awp_box_content bg-sg-box">
        <div class="ShowScroll" style="height:250px; width:100%;">
          <asp:UpdatePanel ChildrenAsTriggers="true"
UpdateMode="Conditional" ID="AssetChecklistUpdatePanel" runat="server">
            <ContentTemplate>
              <asp:CheckBoxList ID="AssetCheckList" runat="server"
Width="100%" Height="100%" style="text-align:left">
                </asp:CheckBoxList>
            </ContentTemplate>
            <Triggers>
              <asp:AsyncPostBackTrigger ControlID="SelectAllBtn"
EventName="Click" />
            </Triggers>
          </asp:UpdatePanel>
        </div>
      </div>
    </div>
    <asp:Button CssClass="form-control" ID="PreviewBtn"
OnClick="PreviewBtn_Click" Width="100%" Text="Preview" runat="server"/>
  </div>
</div>
</div>
<div class="col-md-9">
  <!-- <span class="awp-save-btn bg-green fg-white shadow mif-ani-flash "

```



```

onclick="printpackingslip()"><i title="Print" style="font-size:2em; vertical-align:top" class="mif-
printer av-hand-cursor fg-white shadow-metro-black"></i></span>-->
    <div class="awp_box rounded bg-sg-title shadow">
        <div class="awp_box_title bg-sg-title">
            <span class="fg-white shadow-metro-black">Labels</span>
        </div>
        <div class="awp_box_content bg-sg-box">
            <asp:UpdatePanel runat="server" ID="BarcodePrintUpdatePanel"
UpdateMode="Conditional" ChildrenAsTriggers="true">
                <ContentTemplate>
                    <div id="ReceivingHidden" style="display:none"><asp:Literal
ID="ReceivingLink" runat="server"></asp:Literal> </div>
                    <div id="PrintoutDiv" style="min-height:100px;height:300px;">
                        <!-- <iframe frameborder="0" style="position:absolute; min-height:100px;"
id="ReportFrame" runat="server" src="#" ></iframe>-->
                        <object id="PdfEmbeddedViewer" data='<%= userid %>'
type="application/pdf" width="100%" height="100%">
                            <p>This browser does not support PDFs. Please download the PDF
to view it: <a href="/pdf/sample-3pp.pdf">Download PDF</a>.</p>
                        </object>s
                    </div>
                </ContentTemplate>
                <Triggers>

                </Triggers>
            </asp:UpdatePanel>

        </div>
    </div>
</div>

<div>
    <asp:TextBox runat="server" ID ="avSelectedSearch" ClientIDMode="Static"
style="display:none"></asp:TextBox>
    <asp:TextBox runat="server" ID ="avSelectedView" ClientIDMode="Static"
style="display:none"></asp:TextBox>
    <asp:Button ID="avChangeView" runat="server" Text="CLICK ME" style="display:none"
ClientIDMode="Static"/>

</asp:Content>

```

```

using Microsoft.AspNet.Identity;
using Microsoft.AspNet.Identity.Owin;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace Web_App_Master.Account
{
    public partial class PrintBarcodes : System.Web.UI.Page
    {
        public string userid { get {
            var r = ".bl"; //barcode handler
            var choice = Session["IsTextLabel"] as string;
            if (choice!=null)
            {
                var check = Convert.ToBoolean(choice);
                if (check)
                {
                    r = ".l"; //labelhandler
                }
            }
            try
            {
                var manager =
Context.GetOwinContext().GetUserManager<ApplicationUserManager>();
                // Require the user to have a confirmed email before they can log on.
                var user = manager.FindByName(User.Identity.Name);
                r = user.Id + r;
            }
            catch { }

            return r; } }
        protected void Page_Load(object sender, EventArgs e)
        {
            if(!IsPostBack)
            {
                AssetCheckList.DataSource = (from a in AssetController.GetAllAssets() orderby
a.AssetNumber select a.AssetNumber).ToList();
                AssetCheckList.DataBind();
            }
        }
    }
}

```

```
    }  
}
```

```
protected void SelectAllBtn_Click(object sender, EventArgs e)
```

```
{  
    foreach (ListItem item in AssetCheckList.Items)  
    {  
        item.Selected = true;  
    }  
  
}
```

```
protected void PreviewBtn_Click(object sender, EventArgs e)
```

```
{  
    var manager = Context.GetOwinContext().GetUserManager<ApplicationUserManager>();  
  
    // Require the user to have a confirmed email before they can log on.  
    var user = manager.FindByName(User.Identity.Name);  
    if (user != null)  
    {  
        var id = user.Id;  
        var list = new List<string>();  
        foreach (ListItem item in AssetCheckList.Items)  
        {  
            if (item.Selected == true)  
            {  
                list.Add(item.Text);  
            }  
        }  
        Application["CurrentPrintList" + id] = list;  
    }  
    else  
    {  
        var id = "nouser";  
        var list = new List<string>();  
        foreach (ListItem item in AssetCheckList.Items)  
        {  
            if (item.Selected == true)  
            {  
                list.Add(item.Text);  
            }  
        }  
    }  
}
```

```
Application["CurrentPrintList" + id] = list;  
}
```

```
}
```

```
protected void hiddenModeSwitch_Click(object sender, EventArgs e)
```

```
{  
    var choice = Session["IsTextLabel"] as string;  
    var check = Convert.ToBoolean(choice);  
    if(check)
```

```
{  
        Session["IsTextLabel"] = "false";  
    }
```

```
else
```

```
{  
        Session["IsTextLabel"] = "true";  
    }
```

```
}
```

```
}
```

```
}
```

```
<%@ Page Title="Register" Language="C#" MasterPageFile="~/Site.Master"
AutoEventWireup="true" CodeBehind="Register.aspx.cs"
Inherits="Web_App_Master.Account.Register" %>
```

```
<asp:Content runat="server" ID="BodyContent" ContentPlaceHolderID="MainContent">
```

```
<div class="bg-shaded"></div>
```

```
<div class="row">
```

```
<div class="col-md-offset-4 col-md-4">
```

```
<h1 class="fg-white shadow-metro-black"><%= Title %></h1>
```

```
<p class="text-danger">
```

```
<asp:Literal runat="server" ID="ErrorMessage" />
```

```
</p>
```

```
<div class="form-horizontal">
```

```
<h4 class="fg-white shadow-metro-black">Create a new account</h4>
```

```
<hr />
```

```
<asp:ValidationSummary runat="server" CssClass="text-danger" />
```

```
<div class="form-group fg-white shadow-metro-black">
```

```
<asp:Label runat="server" AssociatedControlID="Email" CssClass="col-md-2
control-label">Email</asp:Label>
```

```
<div class="col-md-10">
```

```
<asp:TextBox runat="server" ID="Email" CssClass="form-control"
TextMode="Email" />
```

```
<asp:RequiredFieldValidator runat="server" ControlToValidate="Email"
```

```
CssClass="text-danger" ErrorMessage="The email field is required." />
```

```
</div>
```

```
</div>
```

```
<div class="form-group fg-white shadow-metro-black">
```

```
<asp:Label runat="server" AssociatedControlID="Password" CssClass="col-md-
2 control-label">Password</asp:Label>
```

```
<div class="col-md-10">
```

```
<asp:TextBox runat="server" ID="Password" TextMode="Password"
CssClass="form-control" />
```

```
<asp:RequiredFieldValidator runat="server" ControlToValidate="Password"
```

```
CssClass="text-danger" ErrorMessage="The password field is required." />
```

```
</div>
```

```
</div>
```

```
<div class="form-group fg-white shadow-metro-black">
```

```
        <asp:Label runat="server" AssociatedControlID="ConfirmPassword"
CssClass="col-md-2 control-label">Confirm password</asp:Label>
        <div class="col-md-10">
            <asp:TextBox runat="server" ID="ConfirmPassword" TextMode="Password"
CssClass="form-control" />
            <asp:RequiredFieldValidator runat="server"
ControlToValidate="ConfirmPassword"
            CssClass="text-danger" Display="Dynamic" ErrorMessage="The confirm
password field is required." />
            <asp:CompareValidator runat="server" ControlToCompare="Password"
ControlToValidate="ConfirmPassword"
            CssClass="text-danger" Display="Dynamic" ErrorMessage="The password
and confirmation password do not match." />
        </div>
    </div>
    <div class="form-group">
        <div class="col-md-12">
            <asp:Button runat="server" OnClick="CreateUser_Click" Text="Register"
CssClass="btn btn-default" />
        </div>
    </div>
</div>

</asp:Content>
```

```

using System;
using System.Linq;
using System.Web;
using System.Web.UI;
using Microsoft.AspNet.Identity;
using Microsoft.AspNet.Identity.Owin;
using Owin;
using Web_App_Master.Models;
using System.IO;
using System.Threading.Tasks;

namespace Web_App_Master.Account
{
    public partial class Register : Page
    {
        protected void CreateUser_Click(object sender, EventArgs e)
        {
            var manager = Context.GetOwinContext().GetUserManager<ApplicationUserManager>();
            var signInManager = Context.GetOwinContext().Get<ApplicationSignInManager>();
            var user = new ApplicationUser() { UserName = Email.Text, Email = Email.Text };

            IdentityResult result = manager.Create(user, Password.Text);
            if (result.Succeeded)
            {
                // For more information on how to enable account confirmation and password reset
                please visit https://go.microsoft.com/fwlink/?LinkID=320771
                string code = manager.GenerateEmailConfirmationToken(user.Id);
                string callbackUrl = IdentityHelper.GetUserConfirmationRedirectUrl(code, user.Id,
Request);
                //create body and subject

                var a = Task.FromResult( manager.SendEmailAsync(user.Id, "Confirm your account",
                "Please confirm your account by clicking <a href=\"" + callbackUrl + "\">here</a>."));

                //send email
                //EmailHelper.SendEmailAsync(user.Email, "Please confirm your account by clicking <a
                href=\"" + callbackUrl + "\">here</a>.", "Confirm your account");
                //manager.SendEmail(user.Id, "Confirm your account", "Please confirm your account by
                clicking <a href=\"" + callbackUrl + "\">here</a>.");
            }
        }
    }
}

```

```
    if (user.EmailConfirmed)
    {
        signInManager.SignIn( user, isPersistent: false, rememberBrowser: false);
        IdentityHelper.RedirectToReturnUrl(Request.QueryString["ReturnUrl"], Response);
    }
    else
    {
        ErrorMessage.Text = "An email has been sent to your account. Please view the email  
and confirm your account to complete the registration process.";
    }
}
else
{
    ErrorMessage.Text = result.Errors.FirstOrDefault();
}
}
}
```



```
<%@ Page Title="Register an external login" Language="C#" MasterPageFile="~/Site.Master"
AutoEventWireup="true" CodeBehind="RegisterExternalLogin.aspx.cs"
Inherits="Web_App_Master.Account.RegisterExternalLogin" Async="true" %>
```

```
<asp:Content runat="server" ID="BodyContent" ContentPlaceHolderID="MainContent">
<h3>Register with your <%: ProviderName %> account</h3>
```

```
    <asp:Placeholder runat="server">
        <div class="form-horizontal">
            <h4>Association Form</h4>
            <hr />
            <asp:ValidationSummary runat="server" ShowModelStateErrors="true" CssClass="text-
danger" />
            <p class="text-info">
                You've authenticated with <strong><%: ProviderName %></strong>. Please enter an
email below for the current site
                and click the Log in button.
            </p>

            <div class="form-group">
                <asp:Label runat="server" AssociatedControlID="email" CssClass="col-md-2 control-
label">Email</asp:Label>
                <div class="col-md-10">
                    <asp:TextBox runat="server" ID="email" CssClass="form-control" TextMode="Email"
/>

                    <asp:RequiredFieldValidator runat="server" ControlToValidate="email"
                        Display="Dynamic" CssClass="text-danger" ErrorMessage="Email is required" />
                    <asp:ModelErrorMessage runat="server" ModelStateKey="email" CssClass="text-
error" />
                </div>
            </div>

            <div class="form-group">
                <div class="col-md-offset-2 col-md-10">
                    <asp:Button runat="server" Text="Log in" CssClass="btn btn-default"
OnClick="LogIn_Click" />
                </div>
            </div>
        </asp:Placeholder>
    </asp:Content>
```

```

using System;
using System.Web;
using Microsoft.AspNet.Identity;
using Microsoft.AspNet.Identity.Owin;
using Microsoft.Owin.Security;
using Owin;
using Web_App_Master.Models;

namespace Web_App_Master.Account
{
    public partial class RegisterExternalLogin : System.Web.UI.Page
    {
        protected string ProviderName
        {
            get { return (string)ViewState["ProviderName"] ?? String.Empty; }
            private set { ViewState["ProviderName"] = value; }
        }

        protected string ProviderAccountKey
        {
            get { return (string)ViewState["ProviderAccountKey"] ?? String.Empty; }
            private set { ViewState["ProviderAccountKey"] = value; }
        }

        private void RedirectOnFail()
        {
            Response.Redirect((User.Identity.IsAuthenticated) ? "~/Account/Manage" :
"~/Account/Login");
        }

        protected void Page_Load()
        {
            // Process the result from an auth provider in the request
            ProviderName = IdentityHelper.GetProviderNameFromRequest(Request);
            if (String.IsNullOrEmpty(ProviderName))
            {
                RedirectOnFail();
                return;
            }
            if (!IsPostBack)
            {
                var manager =

```

```

Context.GetOwinContext().GetUserManager<ApplicationUserManager>();
    var signInManager = Context.GetOwinContext().Get<ApplicationSignInManager>();
    var loginInfo = Context.GetOwinContext().Authentication.GetExternalLoginInfo();
    if (loginInfo == null)
    {
        RedirectOnFail();
        return;
    }
    var user = manager.Find(loginInfo.Login);
    if (user != null)
    {
        signInManager.SignIn(user, isPersistent: false, rememberBrowser: false);
        IdentityHelper.RedirectToReturnUrl(Request.QueryString["ReturnUrl"], Response);
    }
    else if (User.Identity.IsAuthenticated)
    {
        // Apply Xsrf check when linking
        var verifiedloginInfo =
Context.GetOwinContext().Authentication.GetExternalLoginInfo(IdentityHelper.XsrfKey,
User.Identity.GetUserId());
        if (verifiedloginInfo == null)
        {
            RedirectOnFail();
            return;
        }

        var result = manager.AddLogin(User.Identity.GetUserId(), verifiedloginInfo.Login);
        if (result.Succeeded)
        {
            IdentityHelper.RedirectToReturnUrl(Request.QueryString["ReturnUrl"], Response);
        }
        else
        {
            AddErrors(result);
            return;
        }
    }
    else
    {
        email.Text = loginInfo.Email;
    }
}

```

```

    }

    protected void LogIn_Click(object sender, EventArgs e)
    {
        CreateAndLoginUser();
    }

    private void CreateAndLoginUser()
    {
        if (!IsValid)
        {
            return;
        }
        var manager = Context.GetOwinContext().GetUserManager<ApplicationUserManager>();
        var signInManager =
Context.GetOwinContext().GetUserManager<ApplicationSignInManager>();
        var user = new ApplicationUser() { UserName = email.Text, Email = email.Text };
        IdentityResult result = manager.Create(user);
        if (result.Succeeded)
        {
            var loginInfo = Context.GetOwinContext().Authentication.GetExternalLoginInfo();
            if (loginInfo == null)
            {
                RedirectToFail();
                return;
            }
            result = manager.AddLogin(user.Id, loginInfo.Login);
            if (result.Succeeded)
            {
                signInManager.SignIn(user, isPersistent: false, rememberBrowser: false);

                // For more information on how to enable account confirmation and password reset
                please visit https://go.microsoft.com/fwlink/?LinkID=320771
                // var code = manager.GenerateEmailConfirmationToken(user.Id);
                // Send this link via email: IdentityHelper.GetUserConfirmationRedirectUrl(code,
                user.Id)

                IdentityHelper.RedirectToReturnUrl(Request.QueryString["ReturnUrl"], Response);
                return;
            }
        }
        AddErrors(result);
    }

```

```
}

private void AddErrors(IdentityResult result)
{
    foreach (var error in result.Errors)
    {
        ModelState.AddModelError("", error);
    }
}
}
}
```

```
<%@ Page Title="Reset Password" Language="C#" MasterPageFile="~/Site.Master"
AutoEventWireup="true" CodeBehind="ResetPassword.aspx.cs"
Inherits="Web_App_Master.Account.ResetPassword" Async="true" %>
```

```
<asp:Content runat="server" ID="BodyContent" ContentPlaceHolderID="MainContent">
```

```
    <div class="bg-shaded"></div>
```

```
    <div class="row">
```

```
        <div class="col-md-offset-4 col-md-4">
```

```
        <h2 class="fg-white shadow-metro-black"><%: Title %></h2>
```

```
        <p class="fg-red shadow-metro-black">
```

```
            <asp:Literal runat="server" ID="ErrorMessage" />
```

```
        </p>
```

```
    <div class="form-horizontal ">
```

```
        <h4 class="fg-white shadow-metro-black">Enter your new password</h4>
```

```
        <hr />
```

```
        <asp:ValidationSummary runat="server" CssClass="text-danger" />
```

```
        <div class="form-group">
```

```
            <asp:Label runat="server" AssociatedControlID="Email" CssClass="col-md-2 control-label
fg-white shadow-metro-black">Email</asp:Label>
```

```
            <div class="col-md-10">
```

```
                <asp:TextBox runat="server" ID="Email" CssClass="form-control" TextMode="Email" />
```

```
                <asp:RequiredFieldValidator runat="server" ControlToValidate="Email"
```

```
                    CssClass="fg-red shadow-metro-black" ErrorMessage="The email field is required."
```

```
            />
```

```
        </div>
```

```
    </div>
```

```
    <div class="form-group">
```

```
        <asp:Label runat="server" AssociatedControlID="Password" CssClass="col-md-2 control-
label fg-white shadow-metro-black">Password</asp:Label>
```

```
        <div class="col-md-10">
```

```
            <asp:TextBox runat="server" ID="Password" TextMode="Password" CssClass="form-
control" />
```

```
            <asp:RequiredFieldValidator runat="server" ControlToValidate="Password"
```

```
                CssClass="fg-red shadow-metro-black" ErrorMessage="The password field is
required." />
```

```
        </div>
```

```
    </div>
```

```
    <div class="form-group">
```

```
        <asp:Label runat="server" AssociatedControlID="ConfirmPassword" CssClass="col-md-2
control-label fg-white shadow-metro-black">Confirm password</asp:Label>
```

```
        <div class="col-md-10">
```

```
        <asp:TextBox runat="server" ID="ConfirmPassword" TextMode="Password"
CssClass="form-control" />
        <asp:RequiredFieldValidator runat="server" ControlToValidate="ConfirmPassword"
            CssClass=" fg-red shadow-metro-black" Display="Dynamic" ErrorMessage="The
confirm password field is required." />
        <asp:CompareValidator runat="server" ControlToCompare="Password"
ControlToValidate="ConfirmPassword"
            CssClass="fg-red shadow-metro-black" Display="Dynamic" ErrorMessage="The
password and confirmation password do not match." />
    </div>
</div>
<div class="form-group">
    <div class="col-md-12">
        <asp:Button runat="server" OnClick="Reset_Click" Text="Reset" CssClass="btn btn-
default" />
    </div>
</div>
</div>
</div>
</div>
</div>
</asp:Content>
```

```

using System;
using System.Linq;
using System.Web;
using System.Web.UI;
using Microsoft.AspNet.Identity;
using Microsoft.AspNet.Identity.Owin;
using Owin;
using Web_App_Master.Models;
using System.IO;

namespace Web_App_Master.Account
{
    public partial class ResetPassword : Page
    {
        protected string StatusMessage
        {
            get;
            private set;
        }

        protected void Reset_Click(object sender, EventArgs e)
        {
            string code = IdentityHelper.GetCodeFromRequest(Request);
            if (code != null)
            {
                var manager =
Context.GetOwinContext().GetUserManager<ApplicationUserManager>();

                var user = manager.FindByName(Email.Text);
                if (user == null)
                {
                    ErrorMessage.Text = "No user found";
                    return;
                }

                var result = manager.ResetPassword(user.Id, code, Password.Text);
                if (result.Succeeded)
                {
                    Response.Redirect("~/Account/ResetPasswordConfirmation");
                    return;
                }
                ErrorMessage.Text = result.Errors.FirstOrDefault();
            }
        }
    }
}

```



```
    return;  
}
```

```
    ErrorMessage.Text = "An error has occurred";
```

```
    }  
}  
}
```

```
<%@ Page Title="Password Changed" Language="C#" MasterPageFile="~/Site.Master"
AutoEventWireup="true" CodeBehind="ResetPasswordConfirmation.aspx.cs"
Inherits="Web_App_Master.Account.ResetPasswordConfirmation" Async="true" %>
```

```
<asp:Content runat="server" ID="BodyContent" ContentPlaceHolderID="MainContent">
```

```
  <div class="bg-shaded"></div>
```

```
  <div class="row">
```

```
    <div class="col-md-offset-4 col-md-4">
```

```
      <h2 class="fg-white shadow-metro-black"><%= Title %>.</h2>
```

```
      <div class="fg-white shadow-metro-black">
```

```
        <h4> Your password has been changed. Click <asp:HyperLink ID="login" runat="server"
NavigateUrl="~/Account/Login" CssClass="fg-starrag shadow-metro-
white"><b>here</b></asp:HyperLink> to login </h4>
```

```
      </div>
```

```
    </div>
```

```
  </div>
```

```
</asp:Content>
```

```
using System.Web.UI;
```

```
namespace Web_App_Master.Account
```

```
{
```

```
    public partial class ResetPasswordConfirmation : Page
```

```
    {
```

```
    }
```

```
}
```

```

<%@ Page Title="Checkout" Language="C#" MasterPageFile="~/Site.Master"
AutoEventWireup="true" CodeBehind="ShoppingCart.aspx.cs"
Inherits="Web_App_Master.Account.ShoppingCart" %>
<asp:Content ID="ShopCartContent" ContentPlaceHolderID="MainContent" runat="server">
    <div class="bg-shaded"></div>
    <nav class="navbar navbar-inverse bg-grayDark" style="margin-top:0px !important;
height:auto; width:100%; left:0px!important; border-radius:0px!important; position:fixed !important;
z-index:25000;">

        <asp:PlaceHolder ID="ApplyChangesButton" runat="server" Visible="true">
            <ul class="nav navbar-nav navbar-right starrag-menu">
                <li>
                    <asp:LinkButton runat="server" OnClick="ContinueToCheckOutBtn_Click"
ID="ContinueToCheckOutBtn" ClientIDMode="Static" style="vertical-align:bottom; text-
decoration:none"><strong>Confirm&nbsp;&nbsp;</strong><i style="vertical-align:top"
class="glyphicon glyphicon-arrow-right md-glyph"></i></asp:LinkButton>

                </li>
            </ul>

        </asp:PlaceHolder>
    </nav>
    <asp:PlaceHolder ID="MessagePlaceHolder" ClientIDMode="Static" runat="server"
Visible="false">
        <div class="row">
            <div class="col-md-12">
                <div class="awp_box rounded bg-sg-title shadow">
                    <div class="awp_box_title bg-sg-title">
                        <span class="fg-white shadow-metro-black"><span class="mif-warning mif-ani-
flash mif-ani-slow"></span></span>
                    </div>
                    <div class="awp_box_content bg-red fg-white shadow-metro-black">
                        <asp:Literal ID="ErrorMessage" runat="server"></asp:Literal>
                    </div>
                </div>
            </div>
        </div>
    </div>

</asp:PlaceHolder>
<div class="bg-metro" style="padding-top:60px !important"></div>
    <div class="row">
        <div class="col-md-3" id="ShippingOptionsBox">

```

```

<div class="awp_box rounded bg-sg-title shadow">
  <div class="awp_box_title bg-sg-title">
    <span class="fg-white shadow-metro-black">Shipping Options</span></div>
    <div class="awp_box_content bg-sg-box" style="text-align:left;" >
      <h4 class="fg-white shadow-metro-black"> <%=
Page.User.Identity.GetUserName() %></h4>
      <h4 class="fg-white shadow-metro-black">
        <asp:Label Text="Email" CssClass="fg-white shadow-metro-black"
runat="server" ID="AnonLabel" Visible="false" />
        <asp:TextBox Visible="false" ID="AnonUserInput" CssClass="form-control"
runat="server" />
        <asp:RequiredFieldValidator ID="AnonValidator" CssClass="fg-red
shadow-metro-black" Enabled="false" ErrorMessage="Email Required"
ControlToValidate="AnonUserInput" runat="server" />
      </h4>
      <hr />
      <h4 class="fg-white shadow-metro-black">Ship To</h4>
      <div id ="CustomerAddressInput">
        <asp:DropDownList Width="100%" ClientIDMode="Static"
ID='checkout_ShipTo' AppendDataBoundItems="true" runat="server" CssClass="form-control">
          <asp:ListItem Text="--Select One--" Value="" />
        </asp:DropDownList></div>
        <span><input id="cb_CustomAddress" runat="server" type="checkbox"
name="cb_CustomAddress" value="" onchange="CustomCheckoutAddressChecked()"
style="display:inline-block; vertical-align:middle; margin-right:5px;" /><h5 style="display:inline-
block;" class="fg-white shadow-metro-black">Custom Address</h5>
        </span>
        <div id="CustomAddressInput" style="display:none">
          <input runat="server" id="SprCompany" type="text" class="form-control"
placeholder="CompanyName">
          <input runat="server" id="SprAddr" type="text" class="form-control"
placeholder="Address">
          <input runat="server" id="SprAddr2" type="text" class="form-control"
placeholder="Address Line #2">
          <input runat="server" id="SprCty" type="text" class="form-control"
placeholder="City">
          <input runat="server" id="SprState" type="text" class="form-control"
placeholder="State">
          <input runat="server" id="SprPostal" type="text" class="form-
control" placeholder="Postal">
          <input runat="server" id="SprCountry" type="text" class="form-
control" placeholder="Country">

```

```

        <input runat="server" id="SprName" type="text" class="form-control"
placeholder="Attn">
        <input runat="server" id="SprEmail" type="text" class="form-control"
placeholder="Email">
        <input runat="server" id="SprPhone" type="text" class="form-
control" placeholder="Phone">
    </div>
    <hr />
    <h4 class="fg-white shadow-metro-black">Order # (optional)</h4>
    <div class="">
        <input runat="server" id="checkout_OrderNumber" type="text" class="form-
control" placeholder="Order Number" />
    </div>
</div>
</div>
</div>
<div class="col-md-9">
    <div class="awp_box rounded bg-sg-title shadow">
        <div class="awp_box_title bg-sg-title">
            <span class="fg-white shadow-metro-black">Items:</span></div>
        <div class="awp_box_content bg-sg-box">
            <div class="row">
                <asp:UpdatePanel runat="server" ID="OutCartUpdatePanel"
ClientIDMode="Static" ChildrenAsTriggers="true" UpdateMode="Conditional">
                    <ContentTemplate>
                        <asp:Repeater runat="server" ID
="FinalCheckoutRepeater" ClientIDMode="Static"
OnItemCommand="FinalCheckoutRepeater_ItemCommand" >
                            <ItemTemplate>
                                <div class="file " title=" draggable="true"
onclick="BarcodeScanned('<## Eval("AssetNumber")%>', '<## Eval("IsHistoryItem")%>', '<##
Eval("DateShippedTicks")%>');">
                                    <div id =" file-color " class="file-color <##
Eval("Color") %>" style="width:100%; height:100%;"></div>

                                <div class="i-icon">
                                    <img style="box-shadow: rgba(0, 0, 0, 0.70) 0px
0px 10px;" src='<## Eval("FirstImage")+ ".r?w=133&h=100" %>' />
                                </div>

                                <div runat="server" class="i-title"><strong
class="fg-white shadow-metro-black"><## Eval("AssetName") %></strong> </div>

```

```

                <div class="i-check">
                    <span id="InOutIndicator" runat="server"
class="edge-badge fg-white shadow-metro-black icon-badge " style="z-index:3
!important;"><strong><%# Eval("AssetNumber")%></strong></span></div>
                <div runat="server" id="DownloadLink" class="i-
download" style="margin-top:-4px;padding-left:2px" onclick="event.stopPropagation();">
<asp:LinkButton ID="RemoveFromCheckOutButton" CommandName="Delete"
CommandArgument='<%# Eval("AssetNumber")%>' runat="server" > <i title="Add To Task List"
id="dl_link" style="font-size:1.5em;" class="glyphicon glyphicon-remove c-red av-hand-
cursor"></i></asp:LinkButton>

```

```

            </div>

```

```

        </div>

```

```

    </ItemTemplate>

```

```

</asp:Repeater>

```

```

</ContentTemplate>

```

```

</asp:UpdatePanel>

```

```

    </div>

```

```

</div>

```

```

</div>

```

```

</div>

```

```

</div>

```

```

</asp:Content>

```

```

using Helpers;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace Web_App_Master.Account
{
    public partial class ShoppingCart : System.Web.UI.Page
    {
        protected void Page_PreInit(object sender, EventArgs e)
        {
            this.SiteMaster().OnPanelUpdate += Checkout_OnPanelUpdate;
        }
        private void Checkout_OnPanelUpdate(object sender, UpdateRequestEvent e)
        {
            FinalCheckoutRepeater.DataBind();
            OutCartUpdatePanel.Update();
        }

        public string Customer = null;
        public string Shipper = null;
        public string Engineer = null;
        public string Ordernumber = null;
        public List<Asset> CheckoutItems = null;

        protected void Page_Load(object sender, EventArgs e)
        {
            this.PreInit += Page_PreInit;
            Button btn = (Button)this.Master.FindControl("UpdateAllCarts");
            AsyncPostBackTrigger Trigger1 = new AsyncPostBackTrigger();
            Trigger1.ControlID = btn.ID;
            Trigger1.EventName = "Click";
            if (OutCartUpdatePanel.Triggers.Count == 1)
                OutCartUpdatePanel.Triggers.Add(Trigger1);

            BindCheckout();
            if (!IsPostBack)
            {

```



```
if (!Page.User.Identity.IsAuthenticated)
{
    AnonLabel.Visible = true;
    AnonUserInput.Visible = true;
    AnonValidator.Enabled = true;
}
```

```
Customer = Session["Customer"] as string;
Shipper = Session["Shipper"] as string;
Engineer = Session["Engineer"] as string;
Ordernumber = Session["Ordernumber"] as string;
checkout_OrderNumber.Value = Ordernumber;
//checkout_ServiceEngineer.Text = Engineer;
```

```
try
{
    string usershipper = Shipper;
    var user = Page.User.Identity.Name;
    var split = user.Split('@');
    user = split[0];
    split = user.Split('.');
    foreach (var s in split)
    {
        foreach (var shipper in Global.Library.Settings.ShippingPersons)
        {
            if (shipper.Name.ToLower().Contains(s.ToLower()) ||
shipper.Email.ToLower().Contains(s.ToLower()))
            {
                Shipper = shipper.Name;
            }
        }
    }
}
catch { }
//checkout_ShippingPerson.Text = Shipper;

checkout_ShipTo.Text = Customer;
checkout_ShipTo.Focus();
}
```

```

    }
    public void BindCheckout()
    {
        var checkout = HttpContext.Current.Session["CheckOut"] as List<Asset>;
        if (checkout == null) checkout = new List<Asset>();
        FinalCheckoutRepeater.DataSource = checkout;
        FinalCheckoutRepeater.DataBind();

        //checkout_ServiceEngineer.DataSource = GetEngineerNames();
        //checkout_ServiceEngineer.DataBind();

        //checkout_ShippingPerson.DataSource = GetShippingNames();
        //checkout_ShippingPerson.DataBind();

        checkout_ShipTo.DataSource = GetShipToNames();
        checkout_ShipTo.DataBind();
    }
    public List<string> GetShipToNames()
    {
        var names = (from ship in Global.Library.Settings.Customers orderby ship.CompanyName
select ship.CompanyName).ToList();
        return names;
    }
    public List<string> GetEngineerNames()
    {
        var names = (from ship in Global.Library.Settings.ServiceEngineers orderby ship.Name
select ship.Name).ToList();
        return names;
    }
    public List<string> GetShippingNames()
    {
        try
        {
            var names = (from ship in Global.Library.Settings.ShippingPersons orderby ship.Name
select ship.Name).ToList();
            return names;
        }
        catch { return new List<string>(); }
    }
    private void FillDefaultShipper(Customer cust)
    {
        SprCompany.Value = cust.CompanyName;
    }

```

```

        SprAddr.Value = cust.Address;
        SprAddr2.Value = cust.Address2;
        SprCty.Value = cust.City;
        SprState.Value = cust.State;
        SprPostal.Value = cust.Postal;
        SprCountry.Value = cust.Country;
        SprName.Value = cust.Attn;
        SprEmail.Value = cust.Email;
        SprPhone.Value = cust.Phone;
    }
    private Customer GetCustomAddress()
    {
        Customer cust = new Helpers.Customer();
        cust.CompanyName = SprCompany.Value;
        cust.Address = SprAddr.Value;
        cust.Address2 = SprAddr2.Value;
        cust.City = SprCty.Value;
        cust.State = SprState.Value;
        cust.Postal = SprPostal.Value;
        cust.Country = SprCountry.Value;
        cust.Attn = SprName.Value;
        cust.EmailAddress = new EmailAddress() { Name = SprCompany.Value, Email=
SprEmail.Value} ;
        cust.Phone = SprPhone.Value;
        return cust;
    }

    protected void ShowError(string error)
    {
        try
        {
            MessagePlaceholder.Visible = true;
            ErrorMessage.Text = error;
        }
        catch { }
    }
    protected void ContinueToCheckOutBtn_Click(object sender, EventArgs e)
    {

        if ((Session["CheckOut"]) == null) return;
        if ((Session["CheckOut"] as List<Asset>).Count == 0) return;
        var transaction = new PendingTransaction(Session["CheckOut"] as List<Asset>);

```

```

transaction.OrderNumber = checkout_OrderNumber.Value;
if (transaction.Email.Email=="")
{
    if (!EmailHelper.IsValid(AnonUserInput.Text))
    {
        ShowError("Please enter a Valid email address");
        return;
    }
    else
    {
        transaction.Email = new EmailAddress() { Name = AnonUserInput.Text, Email =
AnonUserInput.Text };
        transaction.Name = AnonUserInput.Text;
    }
}
if (cb_CustomAddress.Checked)
{
    transaction.Address = GetCustomAddress();
}
else
{
    if (checkout_ShipTo.SelectedValue=="")
    {
        ShowError("No customer selected.");
        return;
    }
    transaction.Address = (from a in Global.Library.Settings.Customers where
checkout_ShipTo.SelectedValue == a.CompanyName select a).ToList().FirstOrDefault();
}
Session["CheckOut"] = new List<Asset>();
Session["PendingTransaction"] = transaction;
Push.Transaction(transaction);
//save to log

//add notification with transactionID

```

```

var rurl = "http://starrag-awp.com/Account/UserCheckout.aspx?tid=" +
transaction.TransactionID;
var body = "Thank you for placing a tool request<br>Your confirmation number is: " +
transaction.TransactionID + "<br> you can view your reciept here <a href='" + rurl + "'></a><br>" +
rurl;

```

```

if (!Global.Library.Settings.TESTMODE)
{
    try
    {

        EmailHelper.SendEmail(transaction.Email.Email, body, "AWP - Order Confirmation:");

        var turl = "http://starrag-awp.com/Account/OutCart.aspx?pend=" +
transaction.TransactionID;
        var staticBody = "A new order has been placed<br>Transaction ID: " +
transaction.TransactionID + "<br> you can complete this transaction here <a href=\"" + turl +
"\"></a><br>" + turl;
        var statics = (from a in Global.Library.Settings.StaticEmails select a.Email).ToArray();
        EmailHelper.SendMassEmail(statics, "", "AWP - New Pending Transaction");
    }
    catch (Exception ex)
    {
        Context.AddError(ex);
    }
}
//if logged in add transaction to recent user transactions

//redirect
Context.Response.Redirect("/Account/UserCheckout.aspx?tid=" +
transaction.TransactionID);
}

```

```

protected void FinalCheckoutRepeater_ItemCommand(object source,
RepeaterCommandEventArgs e)
{
    if (e.CommandName.ToLower() == "delete")
    {
        Asset rem = null;
        foreach (var ass in Session["CheckOut"] as List<Asset>)
        {
            if (ass.AssetId == e.CommandArgument as string)
            {
                rem = ass;
            }
        }
        try
        {

```

```
        (Session["CheckOut"] as List<Asset>).Remove(rem);  
    }  
    catch { }  
    this.UpdateAll();  
}  
}  
}
```

```
<%@ Page Title="Pending Transactions " Language="C#" MasterPageFile="~/Site.Master"
AutoEventWireup="true" CodeBehind="Transactions.aspx.cs"
Inherits="Web_App_Master.Account.Transactions" %>
<asp:Content ID="Content1" ContentPlaceHolderID="MainContent" runat="server">
</asp:Content>
```

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace Web_App_Master.Account
{
    public partial class Transactions : System.Web.UI.Page
    {
        public string TransactionID = "";
        protected void Page_Load(object sender, EventArgs e)
        {
            if (!IsPostBack)
            {
                try
                {
                    TransactionID= Request.QueryString["idx"];
                    if (TransactionID!=null)
                    {
                        var transaction = Web_App_Master.Pull.Transaction(TransactionID);
                    }
                }
                catch (Exception)
                {
                    throw;
                }
            }
        }
    }
}

```



```
<%@ Page Title="Two-Factor Authentication" Language="C#" MasterPageFile="~/Site.Master"
AutoEventWireup="true" CodeBehind="TwoFactorAuthenticationSignIn.aspx.cs"
Inherits="Web_App_Master.Account.TwoFactorAuthenticationSignIn" %>
```

```
<asp:Content ID="BodyContent" ContentPlaceHolderID="MainContent" runat="server">
```

```
    <h2><%: Title %>.</h2>
```

```
    <asp:PlaceHolder runat="server" ID="sendcode">
```

```
        <section>
```

```
            <h4>Send verification code</h4>
```

```
            <hr />
```

```
            <div class="row">
```

```
                <div class="col-md-12">
```

```
                    Select Two-Factor Authentication Provider:
```

```
                    <asp:DropDownList runat="server" ID="Providers">
```

```
                </asp:DropDownList>
```

```
                    <asp:Button Text="Submit" ID="ProviderSubmit" OnClick="ProviderSubmit_Click"
```

```
CssClass="btn btn-default" runat="server" />
```

```
                </div>
```

```
            </div>
```

```
        </section>
```

```
    </asp:PlaceHolder>
```

```
    <asp:PlaceHolder runat="server" ID="verifycode" Visible="false">
```

```
        <section>
```

```
            <h4>Enter verification code</h4>
```

```
            <hr />
```

```
            <asp:HiddenField ID="SelectedProvider" runat="server" />
```

```
            <asp:PlaceHolder runat="server" ID="ErrorMessage" Visible="false">
```

```
                <p class="text-danger">
```

```
                    <asp:Literal runat="server" ID="FailureText" />
```

```
                </p>
```

```
            </asp:PlaceHolder>
```

```
            <div class="form-group">
```

```
                <asp:Label Text="Code:" runat="server" AssociatedControlID="Code" CssClass="col-
md-2 control-label" />
```

```
                <div class="col-md-10">
```

```
                    <asp:TextBox runat="server" ID="Code" CssClass="form-control" />
```

```
                </div>
```

```
            </div>
```

```
            <div class="form-group">
```

```
                <div class="col-md-offset-2 col-md-10">
```

```
                    <div class="checkbox">
```

```
                        <asp:Label Text="Remember Browser" runat="server" />
```

```
        <asp:CheckBox Text="" ID="RememberBrowser" runat="server" />
    </div>
</div>
</div>
<div class="form-group">
    <div class="col-md-offset-2 col-md-10">
        <asp:Button Text="Submit" ID="CodeSubmit" OnClick="CodeSubmit_Click"
CssClass="btn btn-default" runat="server" />
    </div>
</div>
</section>
</asp:Placeholder>
</asp:Content>
```

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using Microsoft.AspNet.Identity;
using Microsoft.AspNet.Identity.Owin;
using Web_App_Master.Models;

namespace Web_App_Master.Account
{
    public partial class TwoFactorAuthenticationSignIn : System.Web.UI.Page
    {
        private ApplicationSignInManager signinManager;
        private ApplicationUserManager manager;

        public TwoFactorAuthenticationSignIn()
        {
            manager = Context.GetOwinContext().GetUserManager<ApplicationUserManager>();
            signinManager =
Context.GetOwinContext().GetUserManager<ApplicationSignInManager>();
        }

        protected void Page_Load(object sender, EventArgs e)
        {
            var userId = signinManager.GetVerifiedUserId<ApplicationUser, string>();
            if (userId == null)
            {
                Response.Redirect("/Account/Error", true);
            }
            var userFactors = manager.GetValidTwoFactorProviders(userId);
            Providers.DataSource = userFactors.Select(x => x).ToList();
            Providers.DataBind();
        }

        protected void CodeSubmit_Click(object sender, EventArgs e)
        {
            bool rememberMe = false;
            bool.TryParse(Request.QueryString["RememberMe"], out rememberMe);

```

```

        var result = signinManager.TwoFactorSignIn<ApplicationUser,
string>(SelectedProvider.Value, Code.Text, isPersistent: rememberMe, rememberBrowser:
RememberBrowser.Checked);
        switch (result)
        {
            case SignInStatus.Success:
                IdentityHelper.RedirectToReturnUrl(Request.QueryString["ReturnUrl"], Response);
                break;
            case SignInStatus.LockedOut:
                Response.Redirect("/Account/Lockout");
                break;
            case SignInStatus.Failure:
            default:
                FailureText.Text = "Invalid code";
                ErrorMessage.Visible = true;
                break;
        }
    }
}

```

```

protected void ProviderSubmit_Click(object sender, EventArgs e)
{
    if (!signinManager.SendTwoFactorCode(Providers.SelectedValue))
    {
        Response.Redirect("/Account/Error");
    }
}

```

```

        var user = manager.FindById(signinManager.GetVerifiedUserId<ApplicationUser,
string>());
        if (user != null)
        {
            var code = manager.GenerateTwoFactorToken(user.Id, Providers.SelectedValue);

            SelectedProvider.Value = Providers.SelectedValue;
            sendcode.Visible = false;
            verifycode.Visible = true;
        }
    }
}
}

```

```
<%@ Page Title="" Language="C#" MasterPageFile="~/Site.Master" AutoEventWireup="true"
CodeBehind="UserCheckout.aspx.cs" Inherits="Web_App_Master.Account.UserCheckout" %>
<asp:Content ID="Content1" ContentPlaceHolderID="MainContent" runat="server">
```

```
    <div class="bg-shaded"></div>
```

```
    <div class="row">
```

```
        <div class="col-md-offset-2 col-md-4 fg-white shadow-metro-black" >
```

```
            <h1>Receipt</h1>
```

```
            <div>
```

```
                <span>Confirmation Number: <%= TID %></span>
```

```
            </div>
```

```
            <div>
```

```
        </div>
```

```
        <div>
```

```
            <span>Name: <%= Name %></span>
```

```
        </div>
```

```
        <div>
```

```
            <span>Email: <%= Email %></span>
```

```
        </div>
```

```
        <div>
```

```
            <span>OrderNumber: <%= OrderNumber %></span>
```

```
        </div>
```

```
        <div id="CustomAddressInput" style="display:normal">
```

```
            <input runat="server" id="SprCompany" type="text" readonly="readonly" class="form-
control" placeholder="CompanyName">
```

```
            <input runat="server" id="SprAddr" type="text" readonly="readonly" class="form-control"
placeholder="Address">
```

```
            <input runat="server" id="SprAddr2" type="text" readonly="readonly" class="form-
control" placeholder="Address Line #2">
```

```
            <input runat="server" id="SprCty" type="text" readonly="readonly" class="form-control"
placeholder="City">
```

```
            <input runat="server" id="SprState" type="text" readonly="readonly" class="form-control"
placeholder="State">
```

```
            <input runat="server" id="SprPostal" type="text" readonly="readonly" class="form-
control" placeholder="Postal">
```

```
            <input runat="server" id="SprCountry" type="text" readonly="readonly" class="form-
control" placeholder="Country">
```

```
            <input runat="server" id="SprName" type="text" readonly="readonly" class="form-
control" placeholder="Attn">
```

```
            <input runat="server" id="SprEmail" type="text" readonly="readonly" class="form-
```

```

control" placeholder="Email">
    <input runat="server" id="SprPhone" type="text" readonly="readonly" class="form-
control" placeholder="Phone">
    </div>
</div>
<div class="col-md-4">
    <div>
        <asp:Repeater ID="ReceiptRepeater" runat="server">
            <ItemTemplate>
                <div class="file " title=" draggable="true" onclick="BarcodeScanned('<##
Eval("AssetNumber")%>', '<## Eval("IsHistoryItem")%>', '<## Eval("DateShippedTicks")%>');">
                    <div id = " file-color " class="file-color <## Eval("Color")
%>" style="width:100%; height:100%;"></div>

                    <div class="i-icon">
                        <img style="box-shadow: rgba(0, 0, 0, 0.70) 0px 0px 10px;"
src='<## Eval("FirstImage")+ ".r?w=133&h=100" %>' />
                    </div>

                    <div runat="server" class="i-title"><strong class="fg-white
shadow-metro-black"><## Eval("AssetName") %></strong> </div>

                    <div class="i-check">
                        <span id="InOutIndicator" runat="server" class="edge-badge fg-
white shadow-metro-black icon-badge " style="z-index:3 !important;"><strong><##
Eval("AssetNumber")%></strong></span></div>
                        <div runat="server" id="DownloadLink" class="i-download"
style="margin-top:-4px;padding-left:2px" onclick="event.stopPropagation();" > <asp:LinkButton
ID="RemoveFromCheckOutButton" CommandName="Delete" CommandArgument='<##
Eval("AssetNumber")%>' runat="server" > <i title="Add To Task List" id="dl_link" style="font-
size:1.5em;" class="glyphicon glyphicon-remove c-red av-hand-cursor"></i></asp:LinkButton>

                    </div>

                </div>
            </ItemTemplate>
        </asp:Repeater>
    </div>
</div>
</div>

```

</asp:Content>

```
using Helpers;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
```

```
namespace Web_App_Master.Account
```

```
{
    public partial class UserCheckout : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            if (!IsPostBack)
            {
                TID= Request.QueryString["tid"];
                if (TID != null)
                {
                    var transaction = Pull.Transaction(TID);
                    OrderNumber = transaction.OrderNumber;
                    Email = transaction.Email.Email;
                    Name = transaction.Name;
                    Date = transaction.Date.ToString();
                    Address = transaction.Address;
                    FillDefaultShipper(Address);
                    BindRepeater(transaction);
                }
            }
        }

        private void BindRepeater(PendingTransaction transaction)
        {
            try
            {
                List<Asset> assets = new List<Asset>();
                foreach (string item in transaction.Assets)
                {
                    var ass = (from a in Pull.Assets() where a.AssetNumber == item select a).ToList();
                    assets.Add(ass[0]);
                }
                ReceiptRepeater.DataSource = assets;
            }
            catch { }
        }
    }
}
```



```

        ReceiptRepeater.DataBind();
    }
    catch { }
}

public string TID { get; set; }
public string OrderNumber { get; set; }
public string Email { get; set; }
public string Date { get; set; }
public Customer Address { get; set; }
public string Name { get; set; }
private void FillDefaultShipper(Customer cust)
{
    SprCompany.Value = cust.CompanyName;
    SprAddr.Value = cust.Address;
    SprAddr2.Value = cust.Address2;
    SprCty.Value = cust.City;
    SprState.Value = cust.State;
    SprPostal.Value = cust.Postal;
    SprCountry.Value = cust.Country;
    SprName.Value = cust.Attn;
    SprEmail.Value = cust.Email;
    SprPhone.Value = cust.Phone;
}

}

}

```

```
<%@ Page Title="AWP" Language="C#" MasterPageFile="~/Site.Master"
AutoEventWireup="true" CodeBehind="UserView.aspx.cs"
Inherits="Web_App_Master.Account.UserView" %>
```

```
<asp:Content ID="AWP" ContentPlaceHolderID="MainContent" runat="server">
```

```
    <link href="../../Content/browser.css" rel="stylesheet" />
<%--    <link href="../../Content/jquery-ui.css" rel="stylesheet" />
    <script src="../../Scripts/jquery-ui.js"></script>

    <link href="../../Content/jquery.contextMenu.css" rel="stylesheet" />
<script src="../../Scripts/jquery.contextMenu.js"></script>
<script src="../../Scripts/jquery.ui.position.js"></script>
<script src="../../Scripts/FileBrowserContextMenuActions.js"></script>--%>
    <script type="text/javascript">
        $(document).ready(function () {
            Sys.WebForms.PageRequestManager.getInstance().add_endRequest(InitialiseSettings);
        });
        function InitialiseSettings() {
            HideLoader();
        }
    </script>

    <!--menu-->
<nav class="navbar navbar-inverse navbar-fixed-top shadow-bottom bg-grayDark" style="
padding-right:25px; margin-top:50px !important; text-align:left !important; left:0px!important;
position:fixed !important; z-index:25000; vertical-align:middle">
    <div class="container-fluid">
        <div class="navbar-header">
            <button type="button" class="navbar-toggle" data-toggle="collapse" data-target="#AVmenu"
style="float:left !important">
                <span class="icon-bar"></span>
                <span class="icon-bar"></span>
                <span class="icon-bar"></span>
            </button>

        </div>
        <div id="AVmenu" class="collapse navbar-collapse" >
            <ul class="nav navbar-nav starrag-menu">
                <li class="dropdown">
                    <a class="dropdown-toggle" data-toggle="dropdown" href="#">View
```

```

<span class="caret"></span></a>
<ul class="dropdown-menu starrag-menu">
  <li><a href="#" onclick="ChangeAssetView('list'); ">Default</a></li>
  <li><a href="#" onclick=" ChangeAssetView('icon')">Icon</a></li>
  <li><a href="#" onclick="ChangeAssetViewListType('list-type-tiles'); ">Detail</a></li>
  <li><a href="#" onclick="ChangeAssetViewListType('list-type-listing'); ">Listing</a></li>
  <li><a href="#" onclick="ChangeAssetView('smtile'); ">List Tile</a></li>
  <li><a href="#" onclick="ChangeAssetView('mdtile'); ">Medium Tile</a></li>
<%--      <li><a href="#" onclick="ChangeAssetView('lgtile'); ">large Tile</a></li>--%>
</ul>
</li>
<li style="margin-top:10px;">
  <div class=" text" data-role="input" style="width:250px !important; vertical-
align:middle!important">
    <asp:TextBox ID="avSearchString" runat="server" ClientIDMode="Static"
CssClass="form-control"></asp:TextBox>
  </div>
</li>
<li>
  <asp:LinkButton ID ="AssetSearchBtn" ClientIDMode="Static" runat="server"
OnClick="AssetSearchBtn_Click"><span class="glyphicon glyphicon-
search""></span></asp:LinkButton>
</li>

<li class="dropdown">
  <a class="dropdown-toggle" data-toggle="dropdown" href="#">Filter
  <span class="caret"></span></a>
  <ul class="dropdown-menu starrag-menu">
    <li><asp:LinkButton OnClientClick="ShowLoader()" ID ="ViewAll" ClientIDMode="Static"
runat="server" OnClick="ViewAll_Click">View All</asp:LinkButton> </li>
    <li><asp:LinkButton ID ="FilterIsOut" ClientIDMode="Static" runat="server"
OnClick="FilterIsOut_Click">Checked Out Only</asp:LinkButton> </li>
    <li><asp:LinkButton ID ="FilterIsIn" ClientIDMode="Static" runat="server"
OnClick="FilterIsIn_Click">Checked In Only</asp:LinkButton> </li>
    <li><asp:LinkButton ID ="FilterIsDamaged" ClientIDMode="Static" runat="server"
OnClick="FilterIsDamaged_Click">Damaged Only</asp:LinkButton> </li>
    <li><asp:LinkButton ID ="FilterOnHold" ClientIDMode="Static" runat="server"
OnClick="FilterOnHold_Click">On Hold Only</asp:LinkButton> </li>
    <li><asp:LinkButton ID ="FilterCalibrated" ClientIDMode="Static" runat="server"
OnClick="FilterCalibrated_Click">Calibrated Only</asp:LinkButton> </li>
  </ul>
</li>

```

```

        <li>
            <asp:LinkButton runat="server" OnClientClick="ShowLoader()" ID="RefreshBtn"
OnClick="RefreshBtn_Click" ><span class="mif-loop2"></span> </asp:LinkButton>
        </li>
    </ul>
</div>
</div>
</nav>

```

```

    <div class="bg-shaded"></div>
<!--updatepanel-->
    <asp:UpdatePanel runat="server" ID="CheckoutUpdatePanel"
UpdateMode="Conditional" ChildrenAsTriggers="true">
    <ContentTemplate>
        <!--REPEATER-->
        <div style=" padding-top:50px !important; background-color:black !important">
            <asp:Repeater ID="AssetViewRepeater" runat="server" EnableViewState="false"
OnItemDataBound="AssetViewRepeater_ItemDataBound">
                <HeaderTemplate> </HeaderTemplate>
                <ItemTemplate></ItemTemplate>
                <FooterTemplate> </FooterTemplate>
            </asp:Repeater>
        </div>
    </ContentTemplate>
    <Triggers>
        <asp:AsyncPostBackTrigger ControlID="avChangeView" EventName="Click" />
        <asp:AsyncPostBackTrigger ControlID="AssetSearchBtn" EventName="Click" />
        <asp:AsyncPostBackTrigger ControlID="FilterIsOut" EventName="Click" />
        <asp:AsyncPostBackTrigger ControlID="FilterIsIn" EventName="Click" />
        <asp:AsyncPostBackTrigger ControlID="FilterIsDamaged" EventName="Click" />
        <asp:AsyncPostBackTrigger ControlID="FilterOnHold" EventName="Click" />
        <asp:AsyncPostBackTrigger ControlID="FilterCalibrated" EventName="Click" />
        <asp:AsyncPostBackTrigger ControlID="RefreshBtn" EventName="Click" />
    </Triggers>
</asp:UpdatePanel>

```

```

    <div id="maincontentTiles" style="height:100%;width:100%;min-height:100%;min-
width:100%;background-color:#2d89ef;">
    </div>
    <asp:TextBox runat="server" ID ="avSelectedSearch" ClientIDMode="Static"

```

```
style="display:none"></asp:TextBox>
<asp:TextBox runat="server" ID ="avSelectedView" ClientIDMode="Static"
style="display:none"></asp:TextBox>
<asp:Button OnClientClick="ShowLoader()" ID="avChangeView" runat="server" Text="CLICK ME"
OnClick="ChangeView_Click" style="display:none" ClientIDMode="Static"/>
<asp:Button OnClientClick="ShowLoader()" ID="AjaxRefresherBtn" runat="server" Text="CLICK
ME" OnClick="Refresher_Click" style="display:none" ClientIDMode="Static"/>
```

```
</asp:Content>
```

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
```

```
namespace Web_App_Master.Account
```

```
{
    public partial class UserView : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {

        }
    }
}
```

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Net.Http;
using System.Web.Http;

namespace Web_App_Master.Account
{
    public class ValuesController1 : ApiController
    {
        // GET api/<controller>
        public IEnumerable<string> Get()
        {
            return new string[] { "value1", "value2" };
        }

        // GET api/<controller>/5
        public string Get(int id)
        {
            return "value";
        }

        // POST api/<controller>
        public void Post([FromBody]string value)
        {
        }

        // PUT api/<controller>/5
        public void Put(int id, [FromBody]string value)
        {
        }

        // DELETE api/<controller>/5
        public void Delete(int id)
        {
        }
    }
}
```

```
<%@ Page Title="Verify Phone Number" Language="C#" MasterPageFile="~/Site.Master"
AutoEventWireup="true" CodeBehind="VerifyPhoneNumber.aspx.cs"
Inherits="Web_App_Master.Account.VerifyPhoneNumber" %>
```

```
<asp:Content ID="BodyContent" ContentPlaceHolderID="MainContent" runat="server">
```

```
<h2><%: Title %>.</h2>
```

```
<p class="text-danger">
```

```
<asp:Literal runat="server" ID="ErrorMessage" />
```

```
</p>
```

```
<div class="form-horizontal">
```

```
<h4>Enter verification code</h4>
```

```
<hr />
```

```
<asp:HiddenField runat="server" ID="PhoneNumber" />
```

```
<asp:ValidationSummary runat="server" CssClass="text-danger" />
```

```
<div class="form-group">
```

```
<asp:Label runat="server" AssociatedControlID="Code" CssClass="col-md-2 control-label">Code</asp:Label>
```

```
<div class="col-md-10">
```

```
<asp:TextBox runat="server" ID="Code" CssClass="form-control" />
```

```
<asp:RequiredFieldValidator runat="server" ControlToValidate="Code"
```

```
CssClass="text-danger" ErrorMessage="The Code field is required." />
```

```
</div>
```

```
</div>
```

```
<div class="form-group">
```

```
<div class="col-md-offset-2 col-md-10">
```

```
<asp:Button runat="server" OnClick="Code_Click"
```

```
Text="Submit" CssClass="btn btn-default" />
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</asp:Content>
```



```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using Microsoft.AspNet.Identity;
using Microsoft.AspNet.Identity.Owin;

namespace Web_App_Master.Account
{
    public partial class VerifyPhoneNumber : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            var manager = Context.GetOwinContext().GetUserManager<ApplicationUserManager>();
            var phonenumber = Request.QueryString["PhoneNumber"];
            var code = manager.GenerateChangePhoneNumberToken(User.Identity.GetUserId(),
phonenumber);
            PhoneNumber.Value = phonenumber;
        }

        protected void Code_Click(object sender, EventArgs e)
        {
            if (!ModelState.IsValid)
            {
                ModelState.AddModelError("", "Invalid code");
                return;
            }

            var manager = Context.GetOwinContext().GetUserManager<ApplicationUserManager>();
            var signInManager = Context.GetOwinContext().Get<ApplicationSignInManager>();

            var result = manager.ChangePhoneNumber(User.Identity.GetUserId(),
PhoneNumber.Value, Code.Text);

            if (result.Succeeded)
            {
                var user = manager.FindById(User.Identity.GetUserId());

                if (user != null)
                {

```

```
        signInManager.SignIn(user, isPersistent: false, rememberBrowser: false);  
        Response.Redirect("/Account/Manage?m=AddPhoneNumberSuccess");  
    }  
}
```

```
// If we got this far, something failed, redisplay form  
ModelState.AddModelError("", "Failed to verify phone");
```

```
}  
}  
}
```

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebApp.aspx.cs"
Inherits="Web_App_Master.Account.WebApp" %>
```

```
<!DOCTYPE html>
```

```
<html xmlns="http://www.w3.org/1999/xhtml">
```

```
<head runat="server">
```

```
    <title>Web App</title>
```

```
    <meta name="viewport" content="width=device-width, initial-scale=1.0, minimum-scale=1.0"/>
```

```
<link rel="stylesheet" media="screen, projection" href="~/Content/screen.css"/>
```

```
<link href="~/Content/metro.css" rel="stylesheet" />
```

```
    <script type="text/javascript" src="~/Scripts/jquery-1.11.0.min.js"></script>
```

```
    <script src="~/Scripts/metro.js"></script>
```

```
<script type="text/javascript">
```

```
    $(document).ready(function () {
```

```
        // define our variables
```

```
        var fullHeightMinusHeader, sideScrollHeight = 0;
```

```
        // create function to calculate ideal height values
```

```
        function calcHeights() {
```

```
            // set height of main columns
```

```
            fullHeightMinusHeader = $(window).height() - $(".app-header").outerHeight();
```

```
            $(".main-content, .sidebar-two, .sidebar-one").height(fullHeightMinusHeader);
```

```
            // set height of sidebar scroll content
```

```
            sideScrollHeight = fullHeightMinusHeader - $(".nav-menu").outerHeight() -
```

```
            $(".buttons").outerHeight();
```

```
            $(".side-scroll").height(sideScrollHeight);
```

```
        } // end calcHeights function
```

```
        // run on page load
```

```
        calcHeights();
```

```
        // run on window resize event
```

```
        $(window).resize(function () {
```

```
            calcHeights();
```

```
});
```

```
});
```

```
</script>
```

```
</head>
```

```
<body style="">
```

```
<form id="form1" runat="server" style="height:100%">
```

```
<!-- app-header -->
```

```
<div class="app-header">
```

```
<div class="app-bar">
```

```
    <a class="app-bar-element branding">Metro UI CSS</a>
```

```
    <span class="app-bar-divider"></span>
```

```
    <ul class="nav navbar-nav small-dropdown">
```

```
        <li data-flexorderorigin="0" data-flexorder="1"><a href="">Home</a></li>
```

```
        <li data-flexorderorigin="1" data-flexorder="2">
```

```
            <a href="" class="dropdown-toggle">Products</a>
```

```
            <ul class="d-menu" data-role="dropdown">
```

```
                <li><a href="">Windows 10</a></li>
```

```
                <li><a href="">Spartan</a></li>
```

```
                <li><a href="">Outlook</a></li>
```

```
                <li><a href="">Office 2015</a></li>
```

```
                <li class="divider"></li>
```

```
                <li><a href="" class="dropdown-toggle">Other Products</a>
```

```
                    <ul class="d-menu" data-role="dropdown">
```

```
                        <li><a href="">Internet Explorer 10</a></li>
```

```
                        <li><a href="">Skype</a></li>
```

```
                        <li><a href="">Surface</a></li>
```

```
                    </ul>
```

```
                </li>
```

```
            </ul>
```

```
        </li>
```

```
        <li data-flexorderorigin="2" data-flexorder="3">
```

```
            <a href="" class="dropdown-toggle">Support</a>
```

```
            <ul class="d-menu" data-role="dropdown">
```

```
                <li><a href="">About</a></li>
```

```
                <li><a href="">Contacts</a></li>
```

```
                <li><a href="">Community forum</a></li>
```

```

</li>
  <a href="" class="dropdown-toggle">Support</a>
  <ul class="d-menu" data-role="dropdown">
    <li><a href="" class="dropdown-toggle">About</a></li>
    <li><a href="">Contacts</a></li>
    <li><a href="">Community forum</a></li>
  </ul>
</li>
</ul>
</li>
<li data-flexorderorigin="3" data-flexorder="4"><a href="">Help</a></li>
</ul>
<div class="app-bar-element place-right active-container">
  <a class="dropdown-toggle fg-white"><span class="mif-enter"></span> Enter</a>
  <div class="app-bar-drop-container place-right" data-role="dropdown" data-no-
close="true" style="display: none;">
    <div class="padding20 fg-dark">

      <h4 class="text-light">Login to service...</h4>
      <div class="input-control text">
        <span class="mif-user prepend-icon"></span>
        <input type="text">
      </div>
      <div class="input-control text">
        <span class="mif-lock prepend-icon"></span>
        <input type="password">
      </div>
      <label class="input-control checkbox small-check">
        <input type="checkbox">
        <span class="check"></span>
        <span class="caption">Remember me</span>
      </label>
      <div class="form-actions">
        <button class="button primary">Login</button>
        <button class="button link fg-grayLighter">Cancel</button>
      </div>

    </div>
  </div>
</div>
<div class="app-bar-pullbutton automatic" style="display: none;"></div><div
class="clearfix" style="width: 0;"></div><nav class="app-bar-pullmenu hidden flexstyle-nav

```

```
navbar-nav" style="display: none;"><ul class="app-bar-pullmenubar hidden nav navbar-  
nav"></ul></nav></div>
```

```
</div><!-- /app-header -->
```

```
<!-- sidebar-one -->
```

```
<div class="sidebar-one" style="height: 618px;">
```

```
<!-- nav-menu -->
```

```
<div class="nav-menu">
```

```
<ul>
```

```
<li><a href="https://learnwebcode.com/tutorial-files/full-height-width/#">Menu Item 1</a></li>
```

```
<li><a href="https://learnwebcode.com/tutorial-files/full-height-width/#">Menu Item 2</a></li>
```

```
<li><a href="https://learnwebcode.com/tutorial-files/full-height-width/#">Menu Item 3</a></li>
```

```
<li><a href="https://learnwebcode.com/tutorial-files/full-height-width/#">Menu Item 4</a></li>
```

```
<li><a href="https://learnwebcode.com/tutorial-files/full-height-width/#">Menu Item 5</a></li>
```

```
</ul>
```

```
</div><!-- /nav-menu -->
```

```
<!-- side-scroll -->
```

```
<div class="side-scroll" style="height: 348px;">
```

```
<!-- side-scroll-inner -->
```

```
<div class="side-scroll-inner">
```

```
<h4>Scrollable Content</h4>
```

```
<p>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut  
labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris  
nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit  
esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in  
culpa qui officia deserunt mollit anim id est laborum.</p>
```

```
<p>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut  
labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris  
nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit  
esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in  
culpa qui officia deserunt mollit anim id est laborum.</p>
```

```
</div><!-- /side-scroll-inner -->
```

```
</div><!-- /side-scroll -->
```

```
<!-- buttons -->
```

```
<div class="buttons">
```

```
<!-- buttons-inner -->
```

```
<div class="buttons-inner">
```

```
<a href="https://learnwebcode.com/tutorial-files/full-height-width/#">Button 1</a>
```

```
<a href="https://learnwebcode.com/tutorial-files/full-height-width/#">Button 2</a>
```

```
</div><!-- /buttons-inner -->
```

```
</div><!-- /buttons -->
```

```
</div><!-- /sidebar-one -->
```

```
<!-- main-content -->
```

```
<div class="main-content" style="height: 618px;">
```

```
<!-- main-content-inner -->
```

```
<div class="main-content-inner">
```

```
</div><!-- /main-content-inner -->
```

```
</div><!-- /main-content -->
```

```
<!-- sidebar-two -->
```

```
<div class="sidebar-two" style="height: 618px;">
```

```
<!-- sidebar-two-inner -->
```

```
<div class="sidebar-two-inner">
```

```
<h3>Sidebar</h3>
```

```
<p>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum.</p>
```

```
<p>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum.</p>
```

```
<p>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut
```

labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum.</p>

<p>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum.</p>

</div><!-- /sidebar-two-inner -->

</div><!-- /sidebar-two -->

</form>

</body>

</html>


```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
```

```
namespace Web_App_Master.Account
```

```
{
    public partial class WebApp : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {

        }
    }
}
```

```
<%@ Page Title="Admin Dashboard" Language="C#" MasterPageFile="~/Site.Master"
AutoEventWireup="true" ValidateRequest="false" EnableEventValidation="false"
CodeBehind="AdminDashboard.aspx.cs"
Inherits="Web_App_Master.Account.Admin.AdminDashboard" %>
<asp:Content ID="Content1" ContentPlaceHolderID="MainContent" runat="server">
```

```
    <nav class="navbar navbar-inverse bg-grayDark" style="margin-top:0px !important; height:auto;
width:100%; left:0px!important; border-radius:0px!important; position:fixed !important; z-
index:25000;">
```

```
        <ul class="nav navbar-nav starrag-menu">
            <li class="fg-white"><h2>Administration</h2> </li>
            <li>
                <asp:LinkButton ID ="UsersAndRolesBtn" OnClick="UsersAndRolesBtn_Click"
ToolTip="User Administration" ClientIDMode="Static" runat="server"><span
>Users</span></asp:LinkButton>
            </li>
            <li>
                <asp:LinkButton ID ="AssetAdminBtn" OnClick="AssetAdminBtn_Click" ToolTip="Library
Options" ClientIDMode="Static" runat="server"><span class="mif-books mif-
3x"></span>Settings</asp:LinkButton>
            </li>
            <li>
                <asp:LinkButton ID ="AssetPeopleManager" OnClick="AssetPeopleManager_Click"
ToolTip="Personnel" ClientIDMode="Static" runat="server"><span class="mif-users mif-
3x"></span>Personnel</asp:LinkButton>
            </li>
            <li>
                <asp:LinkButton ID ="CustomerManager" OnClick="CustomerManager_Click"
ToolTip="Customers" ClientIDMode="Static" runat="server"><span class="mif-contacts-mail mif-
3x"></span>Customers</asp:LinkButton>
            </li>
            <li>
                <asp:LinkButton ID ="NotificationManagerBtn" OnClick="NotificationManagerBtn_Click"
ToolTip="Notification Manager" ClientIDMode="Static" runat="server"><span class="mif-bell mif-
3x">Notices</span></asp:LinkButton>
            </li>
            <li>

        </li>
    </ul>
```

```

<asp:Placeholder ID ="ApplyChangesButton" runat="server" Visible="true">
    <ul class="nav navbar-nav navbar-right">
        <li title="Test Mode" style="padding-top:15px;">
            <span style="display:inline-block; padding-right:10px;" class="fg-white shadow-
metro-black">
                TestMode:<asp:Label Text="True" runat="server" ID="TestModeLabel" />

            </span>
            <span style="display:inline-block; padding-right:10px;">
                <asp:Button OnClick="ChangeTestModeBtn_Click" CssClass="btn btn-default"
runat="server" Text="Change" ID="ChangeTestModeBtn" />
            </span>
        </li>
    </ul>

</asp:Placeholder>

</nav>
<%-- <label class="awp-switch">
    <input title="Test Mode" id="TestModeSwitch" type="checkbox"
checked="checked" onchange="TestModeChanged()">
    <span class="check"></span>
</label>    --%>

<div style="padding-top:60px"></div>
<!--Multiview-->
<asp:Placeholder ID ="MessagePlaceHolder" ClientIDMode="Static" runat="server"
Visible="false">
    <div class="row">
        <div class="col-md-12">
            <div class="awp_box rounded bg-sg-title shadow">
                <div class="awp_box_title bg-sg-title">
                    <span class="fg-white shadow-metro-black"><span class="mif-warning mif-ani-
flash mif-ani-slow"></span></span>
                </div>
                <div class="awp_box_content bg-red fg-white shadow-metro-black">

```

```

        <asp:Literal ID="ErrorMsg" runat="server"></asp:Literal>
    </div>
</div>
</div>
</div>

</asp:Placeholder>
<asp:MultiView ID="AdminMultiView" ActiveViewIndex="0" runat="server"
EnableViewState="true" ViewStateMode="Enabled">
    <asp:View ID="RolesAndUsersAdminView" runat="server">

        <div class="row">
            <div class="col-md-12">

                </div>
            </div>
            <!--row end-->
            <div class="row">
                <div class="col-md-8">
                    <div class="awp_box rounded bg-sg-title shadow">

                        <div class="awp_box_title bg-sg-title">
                            <span class="fg-white shadow-metro-black">Roles</span>
                        </div>
                        <div class="awp_box_content bg-sg-box">
                            <asp:UpdatePanel ChildrenAsTriggers="true" ID="RolesUpdatePanel"
ClientIDMode="Static" runat="server" UpdateMode="Conditional">
                                <ContentTemplate>
                                    <asp:Repeater ID="RolesAndUsersRepeater"
OnItemDataBound="RolesAndUsersRepeater_ItemDataBound" ClientIDMode="Static"
runat="server" ViewStateMode="Enabled" EnableViewState="true">
                                        <HeaderTemplate>
                                            <div class="panel-group" id="rolesAccordion">

                                                </HeaderTemplate>
                                                <ItemTemplate>
                                                    <div class="panel panel-default">
                                                        <div class="panel-heading">
                                                            <h4 class="panel-title text-left">
                                                                <a data-toggle="collapse" data-parent="#rolesAccordion"
href='#collapse<%=# Container.ItemIndex + 1%>'>
                                                                    <strong><%=# Eval("RoleName")%></strong>

```

```

        </a>
    </h4>
</div>
<div id='collapse<%# Container.ItemIndex + 1%>' class="panel-collapse
collapse">

    <div class="panel-body text-left">
        <!--Repeaters for users of this role-->
        <asp:Repeater DataSource='<%# Eval("RoleUsers")%>' runat="server">
            <ItemTemplate>
                <div class="row bg-sg-box">
                    <div class="col-sm-12 text-left bg-white" style="width:auto
important">

                        <asp:Button ToolTip="Delete User From Role"
ID="DeleteFromRole" CssClass="btn btn-sm" runat="server" Text="X" Font-Bold="true"
CommandName='<%#Eval("UserId")%>' CommandArgument='<%#Eval("RoleId") %>'
OnCommand="DeleteFromRole_Command" />

                        <asp:DropDownList ClientIDMode="Static"
ID='RoleDropDown' AppendDataBoundItems="true" runat="server"
DataSource='<%#GetRoleNames() %>' CssClass="dropdown-button">
                            <asp:ListItem Text="--Select One--" Value="" />
                        </asp:DropDownList>
                        <asp:Button ToolTip="Copy User To ROle" ID="CopyRole"
CssClass="btn btn-sm" runat="server" Text="Copy" Font-Bold="true"
CommandName='<%#Eval("UserId")%>' CommandArgument='<%#Eval("RoleId") %>'
OnCommand="CopyRole_Command" />
                        <asp:Button ToolTip="Change User Role" ID="ChangeRole"
CssClass="btn btn-sm" runat="server" Text="Change" Font-Bold="true"
CommandName='<%#Eval("UserId")%>' CommandArgument='<%#Eval("RoleId") %>'
OnCommand="ChangeRole_Command" />
                        <strong><%# Eval("UserName")%></strong>
                    </div>
                </div>
            </ItemTemplate>
        </asp:Repeater>
    </div>
</div>

</ItemTemplate>
<FooterTemplate>
</div>

```

```

        </FooterTemplate>
    </asp:Repeater>

    </ContentTemplate>
</asp:UpdatePanel>
</div>
</div>
</div>
    <div class="col-md-4">
<div class="awp_box rounded bg-sg-title shadow">

    <div class="awp_box_title bg-sg-title">
        <span class="fg-white shadow-metro-black">Users</span>
    </div>
    <div class="awp_box_content bg-sg-box">
        <asp:UpdatePanel ID="UserUpdatePanel" ChildrenAsTriggers="true"
ClientIDMode="Static" UpdateMode="Conditional" runat="server">
            <ContentTemplate>
                <asp:Repeater ID="UserRepeater" ClientIDMode="Static" runat="server">
                    <ItemTemplate>
                        <div class="row bg-sg-box-dark " style="text-align:left">

                            <div class="col-sm-12 bg-sg-box fg-white shadow-metro-black"
style="text-align:left">
                                <asp:Button ToolTip="Delete User" ID="DeleteUser" CssClass="btn
btn-sm" runat="server" Text="X" Font-Bold="true" CommandName='<%=#Eval("UserId")%>'
CommandArgument='<%=#Eval("RoleId") %>' OnCommand="DeleteUser_Command" />
                                    <strong><%=# Eval("UserName")%></strong>
                                </div>

                            </div>
                        </ItemTemplate>
                    </asp:Repeater>

                </ContentTemplate>
            </asp:UpdatePanel>
        </div>
    </div>
</div>
</div>

```

```

<!--row end-->
</asp:View>
<asp:View ID="AssetAdminView" runat="server">

<div id="LibraryOptionsBox" class="row">
  <div class="col-md-12">
    <div class="awp_box rounded bg-sg-title shadow">
      <div class="awp_box_title bg-sg-title">
        <span class="fg-white shadow-metro-black">Library Options</span>
      </div>
      <div class="awp_box_content bg-sg-box">
        <div class="row">
          <asp:PlaceHolder runat="server" ID="MessageHolder" Visible="false">
            <div class="col-md-12"><p class="fg-red text-bold" >
              <asp:Literal runat="server" ID="Notification" />
            </p></div>
          </asp:PlaceHolder>
        </div>
        <div class="row">
          <div class="col-md-6">
            <div class="awp_box rounded bg-sg-title shadow">

              <div class="awp_box_title bg-sg-title">
                <span class="fg-white shadow-metro-black">Upload
Library</span>
              </div>
              <div class="awp_box_content bg-sg-box text-center">
                <asp:FileUpload ID="LibraryUploader" runat="server"
Width="250px" />

                <asp:Button OnClientClick="" ID="UploadLibrary"
ClientIDMode="Static" CssClass="btn" runat="server" Text="Upload Asset File"
OnClick="UploadLibrary_Click" />
              </div>
            </div>
          </div>
          <div class="col-md-6">
            <div class="awp_box rounded bg-sg-title shadow">

              <div class="awp_box_title bg-sg-title">
                <span class="fg-white shadow-metro-black">Upload
Notices</span>

```

```

        </div>
        <div class="awp_box_content bg-sg-box text-center">
            <asp:FileUpload ID="NoticeUploader" runat="server"
Width="250px" CssClass="text-center" />
            <asp:Button OnClientClick="" ID="UploadNotices"
ClientIDMode="Static" CssClass="btn" runat="server" Text="Upload Notices File"
OnClick="UploadNotices_Click" />
        </div>
    </div>
</div>
<div class="col-md-6">
    <div class="awp_box rounded bg-sg-title shadow">

        <div class="awp_box_title bg-sg-title">
            <span class="fg-white shadow-metro-black">Upload
History</span>

        </div>
        <div class="awp_box_content bg-sg-box text-center">
            <asp:FileUpload ID="HistoryUploader" runat="server"
Width="250px" />
            <asp:Button OnClientClick="" ID="UploadHistory"
ClientIDMode="Static" CssClass=" btn" runat="server" Text="Upload History File"
OnClick="UploadHistory_Click" />
        </div>
    </div>
</div>
<div class="col-md-6">
    <div class="awp_box rounded bg-sg-title shadow">

        <div class="awp_box_title bg-sg-title">
            <span class="fg-white shadow-metro-black">Export</span>
        </div>
        <div class="awp_box_content bg-sg-box">
            <asp:Button OnClientClick="" ID="ExportLibrary"
ClientIDMode="Static" CssClass=" btn" runat="server" Text="Export Library"
OnClick="ExportLibrary_Click" />
        </div>
    </div>
</div>
</div>

```



```

<div class="row">
    <div class="col-md-12">
        <div class="awp_box rounded bg-sg-title shadow">

            <div class="awp_box_title bg-sg-title">
                <span class="fg-white shadow-metro-black">SQL Options</span>
            </div>
            <div class="awp_box_content bg-sg-box">
                <asp:Button OnClientClick="ShowLoader()" ID="SendSQL"
CssClass="btn" ClientIDMode="Static" runat="server" Text="Asset Library to SQL"
OnClick="SendSQL_Click" />

                <asp:Button OnClientClick="ShowLoader()"
ID="SendSettingsSQL" CssClass="btn" ClientIDMode="Static" runat="server" Text="Settings To
SQL" OnClick="SendSettingsSQL_Click" />

                <asp:Button OnClientClick="ShowLoader()" ID="PullSettings"
CssClass="btn" ClientIDMode="Static" runat="server" Text="SQL To Settings"
OnClick="PullSettings_Click" />

                <asp:Button OnClientClick="ShowLoader()" ID="PullSQL"
CssClass="btn" ClientIDMode="Static" runat="server" Text="SQL to Asset Library"
OnClick="PullSQL_Click" />

                <asp:Button OnClientClick="ShowLoader()" ID="DeleteSQL"
CssClass="btn" ClientIDMode="Static" runat="server" Text="Erase SQL DataBase"
OnClick="DeleteSQL_Click" />

                <asp:Button OnClientClick="ShowLoader()"
ID="DeleteSettingsSQL" CssClass="btn" ClientIDMode="Static" runat="server" Text="Erase
Settings SQL" OnClick="DeleteSettingsSQL_Click" />
            </div>
        </div>
    </div>
</div>

<div id="StaticEmailsBox" class="row">
    <div class="col-md-12">
        <span class="awp-save-btn bg-red fg-white shadow">
            <asp:LinkButton runat="server" ID="AddStatic" CssClass="fg-white"
ClientIDMode="Static" OnClick="AddStatic_Click">
                <i title="Add" style="font-size:1em;" class="glyphicon glyphicon-plus av-hand-
cursor"></i></asp:LinkButton>

```

```

</span>
<div class="awp_box rounded bg-sg-title shadow">
    <div class="awp_box_title bg-sg-title fg-white shadow-metro-black">
        Static Emails
    </div>
    <div class="awp_box_content bg-sg-box">
        <asp:UpdatePanel ID="staticupdatepanel" ChildrenAsTriggers="true"
UpdateMode="Conditional" runat="server">
            <ContentTemplate>
                <asp:Placeholder runat="server" id="AddStaticPlaceHolder"
Visible="false">
                    <div class="form-group">
                        <div>
                            <div><input class="form-control" type="text" name="AddStaticName"
placeholder="Name" value="" runat="server" /></div>
                            <div><input class="form-control" type="text" name="AddStaticEmail"
placeholder="Email" value="" runat="server" /></div>
                            <div><asp:Button Text="Add" CssClass="btn btn-default"
runat="server" ID="AddStaticSubmit" OnClick="AddStaticSubmit_Click" /></div>
                        </div>
                    </div>
                </asp:Placeholder>
                <asp:Repeater ID="StaticEmailRepeater" runat="server">
                    <ItemTemplate>
                        <div class="row bg-sg-box-dark " style="text-align:left">

                            <div class="col-sm-12 bg-sg-box fg-white shadow-metro-black"
style="text-align:left">
                                <asp:Button ToolTip="Delete Email" ID="DeleteStatic"
CssClass="btn btn-sm" runat="server" Text="X" Font-Bold="true"
CommandName='<%#Eval("Name")%>' CommandArgument='<%#Eval("Email") %>'
OnCommand="DeleteStatic_Command" />
                                <strong><%# Eval("Name")%> : <%#
Eval("Email")%></strong>
                            </div>
                        </div>
                    </ItemTemplate>
                </asp:Repeater>
            </ContentTemplate>
            <Triggers>
                <asp:AsyncPostBackTrigger ControlID="AddStatic" EventName="click" />
                <asp:AsyncPostBackTrigger ControlID="AddStaticSubmit"

```

```

EventName="click" />
    </Triggers>
</asp:UpdatePanel>

</div>
</div>
</div>
</div>
<div id="StandardMsgBox" class="row">
<div class="col-md-12">
    <div class="awp_box rounded bg-sg-title shadow">

        <div class="awp_box_title bg-sg-title">
            <span class="fg-white shadow-metro-black">Standard Messages</span>
        </div>
        <div class="awp_box_content bg-sg-box">
            <div class="col-md-12">
                <span class="awp-save-btn bg-red fg-white shadow">
                    <asp:LinkButton runat="server" ID="SaveCheckOutMsgBtn" CssClass="fg-
white" ClientIDMode="Static" OnClick="SaveCheckOutMsgBtn_Click">
                        <i title="Save" style="font-size:1em;" class="glyphicon glyphicon-floppy-disk
av-hand-cursor"></i></asp:LinkButton>
                    </span>
                <div class="awp_box rounded bg-sg-title shadow">
                    <div class="awp_box_title bg-sg-title">
                        <span class="fg-white shadow-metro-black">Check Out Message</span>
                    </div>
                    <div class="awp_box_content bg-sg-box">
                        <div class=" textarea av-input" style="min-height:75px !important;"><textarea
class="form-control" id="checkoutmsgbox" runat="server" ></textarea></div>
                    </div>
                </div>
            </div>
        </div>
    </div>
</div>
<div class="col-md-12">
    <span class="awp-save-btn bg-red fg-white shadow">
        <asp:LinkButton runat="server" ID="SaveCheckInMsgBtn" CssClass="fg-
white" ClientIDMode="Static" OnClick="SaveCheckInMsgBtn_Click">
            <i title="Save" style="font-size:1em;" class="glyphicon glyphicon-floppy-disk
av-hand-cursor"></i></asp:LinkButton>
        </span>
    <div class="awp_box rounded bg-sg-title shadow">
        <div class="awp_box_title bg-sg-title">

```

```

        <span class="fg-white shadow-metro-black">Check In Message</span>
    </div>
    <div class="awp_box_content bg-sg-box">
        <div class=" textarea av-input" style="min-height:75px !important;"><textarea
class="form-control textarea av-input" id="checkinmsgbox" runat="server" ></textarea></div>
    </div>
</div>
<div class="col-md-12">
    <span class="awp-save-btn bg-red fg-white shadow">
        <asp:LinkButton runat="server" ID="SaveNoticMsgBtn" CssClass="fg-white"
ClientIDMode="Static" OnClick="SaveNoticMsgBtn_Click">
            <i title="Save" style="font-size:1em;" class="glyphicon glyphicon-floppy-disk
av-hand-cursor"></i></asp:LinkButton>
        </span>
        <div class="awp_box rounded bg-sg-title shadow">
            <div class="awp_box_title bg-sg-title">
                <span class="fg-white shadow-metro-black">Notification Message</span>
            </div>
            <div class="awp_box_content bg-sg-box">
                <div class=" textarea av-input" style="min-height:75px !important;"><textarea
class="form-control" id="notificationmsgbox" runat="server" ></textarea></div>
            </div>
        </div>
    </div>
    <div class="col-md-12">
        <span class="awp-save-btn bg-red fg-white shadow">
            <asp:LinkButton runat="server" ID="SaveShipperMsgBtn" CssClass="fg-
white" ClientIDMode="Static" OnClick="SaveShipperMsgBtn_Click">
                <i title="Save" style="font-size:1em;" class="glyphicon glyphicon-floppy-disk
av-hand-cursor"></i></asp:LinkButton>
            </span>
            <div class="awp_box rounded bg-sg-title shadow">
                <div class="awp_box_title bg-sg-title">
                    <span class="fg-white shadow-metro-black">Shipper Message</span>
                </div>
                <div class="awp_box_content bg-sg-box">
                    <div class=" textarea av-input" style="min-height:75px !important;"><textarea
class="form-control" id="shipmsgbox" runat="server" ></textarea></div>
                </div>
            </div>
        </div>
    </div>

```

</div>

</div>

</div>

</div>

<div id="UPSbox" class="row">

<div class="col-md-12">

<!--BUTTONS-->

<asp:LinkButton runat="server" ID="SaveUpsAcctBtn" CssClass="fg-white"
ClientIDMode="Static" OnClick="SaveUpsAcctBtn_Click">

<i title="Save " style="font-size:1em;" class="glyphicon glyphicon-floppy-disk av-
hand-cursor"></i></asp:LinkButton>

<!--BOX-->

<div class="awp_box rounded bg-sg-title shadow">

<div class="awp_box_title bg-sg-title">

UPS Account

</div>

<div class="awp_box_content bg-sg-box">

<div class="row">

<div class="col-md-3">

<div class="awp_box rounded bg-sg-title shadow">

<div class="awp_box_title bg-sg-title">

Access License

Number

</div>

<div class="awp_box_content bg-sg-box">

<div class="input-control text">

<input runat="server" id="ups_aln" type="text" class="av-input"
placeholder="Access License #..." />

</div>

</div>

</div>

</div>

<div class="col-md-3">

<div class="awp_box rounded bg-sg-title shadow">

<div class="awp_box_title bg-sg-title">

User ID

</div>

```

        <div class="awp_box_content bg-sg-box">
            <div class="input-control text ">
                <input runat="server" id="ups_userid" type="text" class="av-input"
placeholder="User ID..." />
            </div>
        </div>
    </div>
    <div class="col-md-3">
        <div class="awp_box rounded bg-sg-title shadow">
            <div class="awp_box_title bg-sg-title">
                <span class="fg-white shadow-metro-black">Password</span>
            </div>
            <div class="awp_box_content bg-sg-box">
                <div class="input-control text ">
                    <input runat="server" id="ups_pwd" type="password" class="av-input"
placeholder="Password..." />
                </div>
            </div>
        </div>
    </div>
    <div class="col-md-3">
        <div class="awp_box rounded bg-sg-title shadow">
            <div class="awp_box_title bg-sg-title">
                <span class="fg-white shadow-metro-black">Shipper Number</span>
            </div>
            <div class="awp_box_content bg-sg-box">
                <div class="input-control text ">
                    <input runat="server" id="ups_shippernumber" type="text" class="av-
input" placeholder="Shipper #" />
                </div>
            </div>
        </div>
    </div>
</div>
</div>
</div>
</div>
</div>

```

```

</asp:View>
<asp:View ID ="PersonnelView" runat="server">
    <div id="PersonnelBox" class="row">
        <div class =" col-md-6">
            <div class="awp_box rounded bg-sg-title shadow">
                <span class="awp-bar " style="padding-bottom:-10px!important;">
                    <span class="awp-bar-btn bg-green fg-white shadow"><i
onclick="ShowDiv('CreateEngineerModal')" title="Add To Task List" class="mif-plus av-hand-
cursor"></i></span>
                </span>
                <div class="awp_box_title bg-sg-title">
                    <span class="fg-white shadow-metro-black">Service Engineers</span>
                </div>
                <div class="awp_box_content bg-sg-box">
                    <asp:Repeater ID="EngineerRepeater" runat="server">
                        <ItemTemplate>
                            <div class="row bg-sg-box" style="margin:10px;">

                                <div class="col-sm-12 bg-sg-box" style="width:auto !important;
text-align:left !important">

                                    <asp:Button ToolTip="Delete"
CommandName="DeleteEngineer" ID="DeleteEngineer" CssClass="btn btn-sm" runat="server"
Text="X" Font-Bold="true" CommandArgument='<%# Eval("Name") %>'
OnCommand="DeleteEngineer_Command" />
                                    <strong class="fg-white shadow-metro-black"><%#
Eval("Name")%>-<%# Eval("Email")%></strong></div>
                                </div>

                            </ItemTemplate>
                        </asp:Repeater>
                    </div>
                </div>

            </div>
            <div class =" col-md-6">
                <div class="awp_box rounded bg-sg-title shadow">

```

```

        <span class="awp-bar " style="padding-bottom:-10px!important;">
            <span class="awp-bar-btn bg-green fg-white shadow"><i
onclick="ShowDiv('CreateShipperModal')" title="Add To Task List" class="mif-plus av-hand-
cursor"></i></span>
        </span>
        <div class="awp_box_title bg-sg-title">
            <span class="fg-white shadow-metro-black">Shipping Personel</span>
        </div>
        <div class="awp_box_content bg-sg-box" style="padding:15px;">

            <asp:Repeater ID="ShippingPersonRepeater" ClientIDMode="Static"
runat="server">

                <ItemTemplate>
                    <div class="row bg-sg-box" style="margin:10px;">

                        <div class="col-sm-12 bg-sg-box" style="width:auto !important;
text-align:left !important">

                            <asp:Button ToolTip="Delete"
ID="DeleteShippingPersonBtn" CssClass="btn btn-sm" runat="server" Text="X" Font-Bold="true"
CommandName='<%=#Eval("Email")%>' OnCommand="DeleteShippingPersonBtn_Command" />
                            <strong class="fg-white shadow-metro-black"><%=#
Eval("Name")%>-<%=# Eval("Email")%></strong></div>
                        </div>

                    </ItemTemplate>
                </asp:Repeater>

            </div>
        </div>

    </div>

</div>

<div id="CreateEngineerModal" class="awp-outer-dialog" style="display:none; z-
index:99998" >
    <div class=" awp-inner-dialog" style="opacity:initial !important; top:115px !important;">
        <span class="awp-dialog-close-btn bg-red fg-white shadow "
onclick="HideDiv('CreateEngineerModal')"><i title="Close" style="vertical-align:top" class="mif-
cross av-hand-cursor fg-white shadow-metro-black"></i></span>
        <div class="awp_box rounded bg-sg-box-dark shadow" style="left:50% !important;
top:30%">
            <div class="awp_box_title bg-sg-title">

```



```

        <span class="fg-white shadow-metro-black"><span class="mif-file-upload mif-
2x"></span>Create Personnel</span>
    </div>
    <div class="awp_box_content bg-sg-title fg-white" style="text-align:left">
        <div>
            <span style="font-weight:bold;" ><strong>Name</strong></span>
            <asp:TextBox runat="server" ID="EngineerNameInput" CssClass="form-
control" TextMode="SingleLine" />
        </div>
        <div>
            <span style="font-weight:bold;" ><strong>Email</strong></span>
            <asp:TextBox runat="server" ID="EngineerEmailInput" CssClass="form-
control" TextMode="Email" />
        </div>
        <asp:Button CausesValidation="false" runat="server" ID="CreateEngineerBtn"
ClientIDMode="Static" OnClick="CreateEngineerBtn_Click" Text="Create" />
    </div>
</div>
</div>
<div id="CreateShipperModal" class="awp-outer-dialog" style="display:none; z-
index:99998" >
    <div class=" awp-inner-dialog" style="opacity:initial !important; top:115px !important;">
        <span class="awp-dialog-close-btn bg-red fg-white shadow "
onclick="HideDiv('CreateShipperModal')"><i title="Close" style="vertical-align:top" class="mif-
cross av-hand-cursor fg-white shadow-metro-black"></i></span>
        <div class="awp_box rounded bg-sg-box-dark shadow" style="left:50% !important;
top:30%">
            <div class="awp_box_title bg-sg-title">
                <span class="fg-white shadow-metro-black"><span class="mif-file-upload mif-
2x"></span>Create Personnel</span>
            </div>
            <div class="awp_box_content bg-sg-title fg-white " style="text-align:left">
                <div>
                    <span style="font-weight:bold;" ><strong>Name</strong></span>
                    <asp:TextBox runat="server" ID="ShipperNameInput" CssClass="form-
control" TextMode="SingleLine" />
                </div>
                <div>
                    <span style="font-weight:bold;" ><strong>Email</strong></span>
                    <asp:TextBox runat="server" ID="ShipperEmialInput" CssClass="form-
control" TextMode="Email" />
                </div>
            </div>
        </div>
    </div>

```

```

        </div>
        <asp:Button CausesValidation="false" runat="server" ID="CreateShipperBtn"
ClientIDMode="Static" OnClick="CreateShipper_Click" Text="Create" />
    </div>
</div>
</div>
</div>
</div>

</asp:View>
<asp:View ID="CustomerView" runat="server">
    <div id="CustomersBox" class="row">
        <div class="col-md-12">
            <div class="awp_box rounded bg-sg-title shadow">
                <span class="awp-bar " style="padding-bottom:-10px!important;">
                    <span class="awp-bar-btn bg-green fg-white shadow"><i
onclick="ShowDiv('CreateCustomerModal')" title="Add To Task List" class="mif-plus av-hand-
cursor"></i></span>
                </span>
                <div class="awp_box_title bg-sg-title">
                    <span class="fg-white shadow-metro-black">Customers</span>
                </div>
                <div class="awp_box_content bg-sg-box">
                    <asp:Repeater ID="CustomerRepeater" ClientIDMode="Static"
runat="server" OnItemCommand="CustomerRepeater_ItemCommand">
                        <ItemTemplate>
                            <div class="panel-group">
                                <div class="panel panel-default">
                                    <div class="panel-heading">
                                        <h4 class="panel-title">
                                            <a data-toggle="collapse" href="#collapse<%=
Container.ItemIndex + 1%>'><%= Eval("CompanyName") %></a>
                                        </h4>
                                    </div>
                                    <div id='collapse<%= Container.ItemIndex + 1%>' class="panel-
collapse collapse">
                                        <div class="panel-body">
                                            <div class="row">
                                                <div class="col-md-4">
                                                    <address>
                                                        <strong><%= Eval("CompanyName") %></strong><br>
                                                        <%= Eval("Address") %><br>
                                                        <%= Eval("Address2") %><br>

```

```

        <%# Eval("Address3") %><br>
        <%# Eval("City") %>, <%# Eval("State") %> <%#
Eval("Postal") %><br>

        <%# Eval("Country") %><br>
        <abbr title="Phone">P:</abbr> <%# Eval("Phone") %>
</address>
<address>
    <strong><%# Eval("Attn") %></strong><br>
    <a href='mailto:<%# Eval("Email") %>'><%# Eval("Email")
%></a>

</address>
</div>
</div>
</div>
</div>
<div class="panel-footer">
    <asp:Button CommandName="DeleteCustomer"
ToolTip="Delete" ID="DeleteEngineer" CssClass="btn btn-sm" runat="server" Text="X" Font-
Bold="true" CommandArgument='<%# Eval("CompanyName") + "|" + Eval("Address") %>'/>

</div>
</div>
</div>
</div>
</ItemTemplate>
</asp:Repeater>
</div>
</div>

</div>
</div>
<div id="CreateCustomerModal" class="awp-outer-dialog" style="display:none; z-
index:99998" >
    <div class=" awp-inner-dialog" style="opacity:initial !important; top:10% !important;">
        <span class="awp-dialog-close-btn bg-red fg-white shadow "
onclick="HideDiv('CreateCustomerModal')"><i title="Close" style="vertical-align:top" class="mif-
cross av-hand-cursor fg-white shadow-metro-black"></i></span>
        <div class="awp_box rounded bg-sg-box-dark shadow" style="left:50% !important;
top:30%">
            <div class="awp_box_title bg-sg-title">
                <span class="fg-white shadow-metro-black"><span class="mif-file-upload mif-
2x"></span>Create Customer</span>
            </div>

```

```

<div class="awp_box_content bg-sg-title fg-white">
  <div class="row">
    <div class="col-md-6" style="text-align:left">
      <div>
        <span style="font-weight:bold;" ><strong>Company
Name</strong></span>
        <asp:TextBox runat="server" ID="SprCompany" CssClass="form-
control" TextMode="SingleLine" />
      </div>
      <div>
        <span style="font-weight:bold;" ><strong>Address</strong></span>
        <asp:TextBox runat="server" ID="SprAddr" CssClass="form-control"
TextMode="SingleLine" />
      </div>
      <div>
        <span style="font-weight:bold;" ><strong>Address Line
2</strong></span>
        <asp:TextBox runat="server" ID="SprAddr2" CssClass="form-control"
TextMode="SingleLine" />
      </div>
      <div>
        <span style="font-weight:bold;" ><strong>City</strong></span>
        <asp:TextBox runat="server" ID="SprCty" CssClass="form-control"
TextMode="SingleLine" />
      </div>
      <div>
        <span style="font-weight:bold;" ><strong>State</strong></span>
        <asp:TextBox runat="server" ID="SprState" CssClass="form-control"
TextMode="SingleLine" />
      </div>
    </div>
    <div class="col-md-6" style="text-align:left">
      <div>
        <span style="font-weight:bold;" ><strong>Postal</strong></span>
        <asp:TextBox runat="server" ID="SprPostal" CssClass="form-control"
TextMode="SingleLine" />
      </div>
      <div>
        <span style="font-weight:bold;" ><strong>Country</strong></span>
        <asp:TextBox runat="server" ID="SprCountry" CssClass="form-control"
TextMode="SingleLine" />
      </div>
    </div>
  </div>
</div>

```

```

        </div>
        <div>
            <span style="font-weight:bold;" ><strong>Attn:</strong></span>
            <asp:TextBox runat="server" ID="SprName" CssClass="form-control"
TextMode="SingleLine" />
        </div>
        <div>
            <span style="font-weight:bold;" ><strong>Email</strong></span>
            <asp:TextBox runat="server" ID="SprEmail" CssClass="form-control"
TextMode="Email" />
        </div>
        <div>
            <span style="font-weight:bold;" ><strong>Phone</strong></span>
            <asp:TextBox runat="server" ID="SprPhone" CssClass="form-control"
TextMode="SingleLine" />
        </div>
    </div>
    <div>
        <asp:Button CausesValidation="false" runat="server" ID="CreateCustomerBtn"
ClientIDMode="Static" OnClick="CreateCustomerBtn_Click" Text="Create" />
    </div>
</div>
</div>
</div>
</div>

</asp:View>
<asp:View ID="NotificationsView" runat="server">
    <div id="NotificationsBox" class="row">
        <div class="col-md-12">
            <div class="awp_box rounded bg-sg-title shadow">
                <div class="awp_box_title bg-sg-title">
                    <span class="fg-white shadow-metro-black">Pending Notifications</span>
                </div>
                <div class="awp_box_content bg-sg-box">
                    <!--Repeaters for 30 Day Notices-->
                    <asp:Repeater ID="Notice30DayRepeater" ClientIDMode="Static"
runat="server">
                        <ItemTemplate>
                            <div class="row bg-sg-box" style="margin:10px;">

```

```

padding-left:10px; text-align:left">
                <asp:Button ToolTip="Delete Notice"
ID="DeleteNotice30DayBtn" CssClass="btn btn-sm" runat="server" Text="X" Font-Bold="true"
CommandName='<%=#Eval("NoticeControlNumber")%>'
CommandArgument='<%=#Eval("GUID")%>' OnCommand="DeleteNotice30DayBtn_Command" />
                <asp:Button ToolTip="Send Notice"
ID="SendNotice30DayBtn" CssClass="btn btn-sm" runat="server" Text="Send" Font-Bold="true"
CommandName='<%=#Eval("NoticeControlNumber")%>'
CommandArgument='<%=#Eval("GUID")%>' OnCommand="SendNotice30DayBtn_Command" />
                <strong class="fg-white shadow-metro-black"><%=#
Eval("NoticeControlNumber")%></strong><strong class="fg-white shadow-metro-black"> &nbsp;
&nbsp; Scheduled in&nbsp;<%=# Eval("DaysUntil").ToString()%>&nbsp;Days</strong> &nbsp;
&nbsp;
                <strong class="fg-white shadow-metro-black"><%=#
Eval("Name")%></strong>
            </div>
        </div>
    </div>
    <hr />
    </ItemTemplate>
</asp:Repeater>
</div>
</div>
</div>
</asp:View>
</asp:MultiView>
</asp:Content>

```

```

using Microsoft.AspNet.Identity;
using Microsoft.AspNet.Identity.EntityFramework;
using Microsoft.AspNet.Identity.Owin;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using Web_App_Master.Models;
using Helpers;
using System.Xml;
using System.IO;
using System.IO.Compression;
using System.Web.UI.HtmlControls;
using static Notification.NotificationSystem;

namespace Web_App_Master.Account.Admin
{
    public partial class AdminDashboard : System.Web.UI.Page
    {
        private void UpdateUsersAndRoles()
        {
            {
                var manager = Context.GetOwinContext().GetUserManager<ApplicationUserManager>();
                var roleStore = new RoleStore<IdentityRole>(new ApplicationDbContext());
                var roleManager = new RoleManager<IdentityRole>(roleStore);
                if (!roleManager.RoleExists("NoRole"))
                {
                    roleManager.Create(new IdentityRole("NoRole"));
                }
                var nonRoledUsers = (from user in manager.Users where user.Roles.Count == 0 orderby
user.Email select user).ToList();
                foreach (var user in nonRoledUsers)
                {
                    manager.AddToRole(user.Id, "NoRole");
                }

                var roles = (from role in roleManager.Roles orderby role.Name select
role).ToRoleBindingList();

                RolesAndUsersRepeater.DataSource = roles;
            }
        }
    }
}

```

```
RolesAndUsersRepeater.DataBind();
```

```
var users = (from user in manager.Users orderby user.Email select  
user).ToUserBindingList();
```

```
UserRepeater.DataSource = users;
```

```
UserRepeater.DataBind();
```

```
}
```

```
private void UpdateAssetAdmin()
```

```
{
```

```
try
```

```
{
```

```
ups_pwd.Value = Global.Library.Settings.UpsAccount.P;
```

```
ups_aln.Value = Global.Library.Settings.UpsAccount.A;
```

```
ups_userid.Value = Global.Library.Settings.UpsAccount.I;
```

```
ups_shippernumber.Value = Global.Library.Settings.UpsAccount.N;
```

```
}
```

```
catch { }
```

```
try
```

```
{
```

```
shipmsgbox.Value = Global.Library.Settings.ShipperNotification;
```

```
checkinmsgbox.Value = Global.Library.Settings.CheckInMessage;
```

```
checkoutmsgbox.Value = Global.Library.Settings.CheckOutMessage;
```

```
notificationmsgbox.Value = Global.Library.Settings.NotificationMessage;
```

```
}
```

```
catch
```

```
{
```

```
}
```

```
try
```

```
{
```

```
if (Global.NoticeSystem.Notices != null)
```

```
{
```

```
Notice30DayRepeater.DataSource = Global.NoticeSystem.Notices;
```

```
Notice30DayRepeater.DataBind();
```

```
}
```

```
}
```

```
catch
```

```
{
```

```
PopNotify("Error", "Could Not Bind Notices");
```



```

    }
    try
    {
        if (Global.Library.Settings.ShippingPersons != null)
        {
            var list = (from n in Global.Library.Settings.ShippingPersons orderby n.Name select
n).ToList();

            EmailBindinglist emailList = new EmailBindinglist(list);
            ShippingPersonRepeater.DataSource = emailList;
            ShippingPersonRepeater.DataBind();
        }
    }
    catch
    {
        PopNotify("Error", "Could Not Bind Shipping Personnel");
    }
    try
    {
        if (Global.Library.Settings.ServiceEngineers != null)
        {
            var list = (from n in Global.Library.Settings.ServiceEngineers orderby n.Name select
n).ToList();

            EmailBindinglist emailList = new EmailBindinglist(list);
            EngineerRepeater.DataSource = emailList;
            EngineerRepeater.DataBind();
        }
    }
    catch
    {
        PopNotify("Error", "Could Not Bind Engineers");
    }
    BindCustomers();
    BindStaticEmails();
}
protected void BindCustomers()
{
    try
    {
        if (Global.Library.Settings.Customers != null)
        {
            var list = (from n in Global.Library.Settings.Customers orderby n.CompanyName

```

```

select n).ToList();
        CustomerBindinglist custlist = new CustomerBindinglist(list);
        CustomerRepeater.DataSource = custlist;
        CustomerRepeater.DataBind();
    }
}
catch
{
    PopNotify("Error", "Could Not Bind Customers");
}
}
protected void ShowError(string error)
{
    try
    {
        MessagePlaceholder.Visible = true;
        ErrorMsg.Text = error;
    }
    catch { }
}

protected void Page_Load(object sender, EventArgs e)
{
    if (!IsPostBack)
    {
        UpdateUsersAndRoles();
        UpdateAssetAdmin();
    }
    TestModeLabel.Text = Global.Library.Settings.TESTMODE.ToString();
}

private List<ApplicationUser> GetUsersForRole(string role)
{
    List<ApplicationUser> users = new List<ApplicationUser>();
    var manager = Context.GetOwinContext().GetUserManager<ApplicationUserManager>();
    foreach(var user in manager.Users)
    {
        foreach(var r in user.Roles)
        {
            if (r.RoleId==role)
            {
                users.Add(user);
            }
        }
    }
}

```

```

        if (r.UserId==user.Id)
        {
            users.Add(user);
        }
    }
}
return users;
}
private bool AddUserToRole(string username, string role)
{
    try
    {
        var manager =
Context.GetOwinContext().GetUserManager<ApplicationUserManager>();
        var roleStore = new RoleStore<IdentityRole>(new ApplicationDbContext());
        var roleManager = new RoleManager<IdentityRole>(roleStore);
        var user = manager.AddToRole(username, role);
        return true;
    }
    catch
    {

        return false;
    }
}
private bool RemoveUserFromRole(string userid, string roleid)
{
    try
    {
        var manager =
Context.GetOwinContext().GetUserManager<ApplicationUserManager>();
        var roleStore = new RoleStore<IdentityRole>(new ApplicationDbContext());
        var roleManager = new RoleManager<IdentityRole>(roleStore);
        /*var irole = roleManager.FindById(role);
        IdentityUserRole remItem=null;
        foreach (var u in irole.Users)
        {
            if (u.UserId==userid && u.RoleId==role)
            {
                remItem = u;
            }
        }
    }
}

```

```

        var aa = irole.Users.Remove(remlItem);
        */
        string rolename = "";
        foreach (var r in roleStore.Roles.ToList())
        {
            if (r.Id==roleid)
            { rolename = r.Name; }
        }
        var user = manager.RemoveFromRole(userid, rolename);
        return user.Succeeded;
    }
    catch
    {

        return false;
    }
}
private bool RemoveUser(string userid)
{
    try
    {
        var manager =
Context.GetOwinContext().GetUserManager<ApplicationUserManager>();
        if (userid == null)
        {
            return false;
        }

        var user = manager.FindById(userid);
        var logins = user.Logins;
        var rolesForUser = manager.GetRoles(userid);

        using (var transaction = new ApplicationDbContext().Database.BeginTransaction())
        {
            foreach (var login in logins.ToList())
            {
                manager.RemoveLogin(login.UserId, new UserLoginInfo(login.LoginProvider,
login.ProviderKey));
            }

            if (rolesForUser.Count() > 0)
            {

```

```

        foreach (var item in rolesForUser.ToList())
        {
            // item should be the name of the role
            var result = manager.RemoveFromRole(user.Id, item);
        }
    }

    manager.Delete(user);
    transaction.Commit();
}
return true;
}
catch (Exception)
{

    return false;
}
}
protected void CreateRoleButton_Click(object sender, EventArgs e)
{
    try
    {
        var roleStore = new RoleStore<IdentityRole>(new ApplicationDbContext());
        var roleManager = new RoleManager<IdentityRole>(roleStore);
        //roleManager.Create(new IdentityRole(RoleName.Text));

        UpdateAssetAdmin();
    }
    catch {
        ShowError("Could Not Create Role.");
    }

}

protected void DeleteUser_Command(object sender, CommandEventArgs e)
{
    var a = e.CommandArgument as string;
    var b = e.CommandName as string;
    RemoveUser(b);
    UpdateUsersAndRoles();
}

```

```

protected bool ChangeUserRole(string userid, string role)
{
    var manager = Context.GetOwinContext().GetUserManager<ApplicationUserManager>();
    try
    {
        if (!manager.IsInRole(userid, role))
        {
            manager.AddToRole(userid, role);
            UpdateUsersAndRoles();
        }
        return true;
    }
    catch {
        return false;
    }
}

protected void ChangeRole_Command(object sender, CommandEventArgs e)
{
    var roleId = e.CommandArgument as string;
    var name = e.CommandName as string;
    try
    {
        foreach (RepeaterItem i in RolesAndUsersRepeater.Items)
        {
            Repeater repeater = (Repeater)i.Controls[1];
            foreach (RepeaterItem item in repeater.Items)
            {
                var changebutton = item.FindControl("ChangeRole") as Button;
                var temprole = changebutton.CommandName;
                if (temprole == name)
                {
                    var dropdown = item.FindControl("RoleDropDown") as DropDownList;
                    if (dropdown != null)
                    {
                        var selected = dropdown.SelectedValue;

                        var result = ChangeUserRole(name, selected);
                        result = RemoveUserFromRole(name, roleId);

                        return;
                    }
                }
            }
        }
    }
}

```

```

        }

    }

}
}
catch {}
}
protected List<string> GetRoleNames()
{
    List<string> names = new List<string>();
    var roleStore = new RoleStore<IdentityRole>(new ApplicationDbContext());
    foreach(var role in roleStore.Roles)
    {
        names.Add(role.Name);
    }
    return names;
}

protected void RolesAndUsersRepeater_ItemDataBound(object sender,
RepeaterItemEventArgs e)
{

}

protected void DeleteFromRole_Command(object sender, CommandEventArgs e)
{
    var a = e.CommandArgument as string;
    var b = e.CommandName as string;
    RemoveUserFromRole(b, a);
    UpdateUsersAndRoles();
}

protected void CopyRole_Command(object sender, CommandEventArgs e)
{
    var roleId = e.CommandArgument as string;
    var name = e.CommandName as string;
    try
    {
        foreach (RepeaterItem i in RolesAndUsersRepeater.Items)
        {

```

```

Repeater repeater = (Repeater)i.Controls[1];
foreach (RepeaterItem item in repeater.Items)
{
    var changebutton = item.FindControl("ChangeRole") as Button;
    var temprole = changebutton.CommandName;
    if (temprole == name)
    {
        var dropdown = item.FindControl("RoleDropDown") as DropDownList;
        if (dropdown != null)
        {
            var selected = dropdown.SelectedValue;
            ChangeUserRole(name, selected);
            return;
        }
    }
}

}

}
}
catch {}
UpdateUsersAndRoles();
}

```

```

protected void UsersAndRolesBtn_Click(object sender, EventArgs e)
{
    AdminMultiView.ActiveViewIndex = 0;
    UpdateUsersAndRoles();
}

```

```

protected void AssetAdminBtn_Click(object sender, EventArgs e)
{
    AdminMultiView.ActiveViewIndex = 1;
    UpdateAssetAdmin();
}

```

```

protected void CustomerManager_Click(object sender, EventArgs e)
{
    AdminMultiView.ActiveViewIndex = 3;
    BindCustomers();
}

```



```

protected void UploadLibrary_Click(object sender, EventArgs e)
{
    try
    {

        var name = Server.MapPath("/Account/library.xml");

        LibraryUploader.PostedFile.SaveAs(name);
        var doc = new XmlDocument();
        using (StreamReader reader = new
StreamReader(LibraryUploader.PostedFile.InputStream))
        {
            doc.LoadXml(reader.ReadToEnd());
        }

        Global.Library.Assets = new List<Asset>();
        try
        {
            XmlNodeList elemList = doc.GetElementsByTagName("Asset");
            foreach (XmlElement elem in elemList)
            {
                var a = elem.ToAsset();
                Global.Library.Assets.Add(a);

            }
        }
        catch
        {
            //PopNotify("Error", "");
            Page.SiteMaster().ShowError("Error Adding Assets");

        }
        try
        {
            if (Global.Library.Settings == null) Global.Library.Settings = new Settings();
            if (Global.Library.Settings.Customers == null) Global.Library.Settings.Customers =
new List<Customer>();
            XmlNodeList custElemList = doc.GetElementsByTagName("Customer");
            foreach (XmlElement cust in custElemList)
            {
                //create a customer
                Customer newCustomer = new Customer();
            }
        }
    }
}

```

```

        newCustomer.AccountNumber = cust.Attributes["AccountNumber"].Value;
        newCustomer.AccPostalCd = cust.Attributes["AccPostalCd"].Value;
        newCustomer.Address = cust.Attributes["Address"].Value;
        newCustomer.Address2 = cust.Attributes["Address2"].Value;
        newCustomer.Address3 = cust.Attributes["Address3"].Value;
        newCustomer.Attn = cust.Attributes["Attn"].Value;
        newCustomer.City = cust.Attributes["City"].Value;
        newCustomer.CompanyName = cust.Attributes["CompanyName"].Value;
        newCustomer.ConsInd = cust.Attributes["ConsInd"].Value;
        newCustomer.Country = cust.Attributes["Country"].Value;
        newCustomer.EmailAddress = new EmailAddress() { Email =
cust.Attributes["Email"].Value };
        newCustomer.Fax = cust.Attributes["Fax"].Value;
        newCustomer.LocID = cust.Attributes["LocID"].Value;
        newCustomer.NickName = cust.Attributes["NickName"].Value;
        newCustomer.OrderNumber = cust.Attributes["OrderNumber"].Value;
        newCustomer.PackageWeight = cust.Attributes["PackageWeight"].Value;
        newCustomer.Phone = cust.Attributes["Phone"].Value;
        newCustomer.Postal = cust.Attributes["Postal"].Value;
        newCustomer.ResInd = cust.Attributes["ResInd"].Value;
        newCustomer.State = cust.Attributes["State"].Value;
        newCustomer.USPSPOBoxIND = cust.Attributes["USPSPOBoxIND"].Value;
        Global.Library.Settings.Customers.Add(newCustomer);
    }
}
catch
{
    //PopNotify("Error", "");
    Page.SiteMaster().ShowError("Error adding customers");
}
//get engineers
try
{
    XmlNodeList engElemList = doc.GetElementsByTagName("Engineer");
    foreach (XmlElement eng in engElemList)
    {
        //create a customer
        EmailAddress newCustomer = new EmailAddress();
        newCustomer.Email = eng.Attributes["Email"].Value;
        newCustomer.Name = eng.Attributes["Name"].Value;
    }
}

```

```

        Global.Library.Settings.ServiceEngineers.Add(newCustomer);
    }
}
catch
{
    //PopNotify("Error", "");
    Page.SiteMaster().ShowError("Error adding Engineers");

}
try
{
    if (Global.Library.Settings.ShippingPersons == null)
Global.Library.Settings.ShippingPersons = new List<EmailAddress>();
    //get shipping personel
    XmlNodeList shipElemList = doc.GetElementsByTagName("Shipper");
    foreach (XmlElement eng in shipElemList)
    {
        //create a customer
        EmailAddress newCustomer = new EmailAddress();
        newCustomer.Email = eng.Attributes["Email"].Value;
        newCustomer.Name = eng.Attributes["Name"].Value;

        Global.Library.Settings.ShippingPersons.Add(newCustomer);
    }
}
catch
{
    //PopNotify("Error", "");
    Page.SiteMaster().ShowError("Error adding Shipping Personnel");

}

//get settings

try
{
    if (Global.Library.Settings.StaticEmails == null) Global.Library.Settings.StaticEmails =
new List<EmailAddress>();
    //get shipping personel
    XmlNodeList staticemails = doc.GetElementsByTagName("StaticEmails");
    foreach (XmlElement staticemail in staticemails)
    {

```

```

        try
        {
            XmlNodeList emails = doc.GetElementsByTagName("StaticEmail");
            //create a customer
            EmailAddress newStatic = new EmailAddress();
            newStatic.Email = staticemail.Attributes["Email"].Value;
            newStatic.Name = staticemail.Attributes["Name"].Value;

            Global.Library.Settings.StaticEmails.Add(newStatic);
        }
        catch { }

    }

    string outmsg =
doc.GetElementsByTagName("CheckOutMessage").Item(0).InnerText.Sanitize();
    Global.Library.Settings.CheckOutMessage = outmsg;

    string inmsg =
doc.GetElementsByTagName("CheckInMessage").Item(0).InnerText.Sanitize();
    Global.Library.Settings.CheckInMessage = inmsg;

    string noticemsg =
doc.GetElementsByTagName("NotificationMessage").Item(0).InnerText.Sanitize();
    Global.Library.Settings.NotificationMessage = noticemsg;

    string shippernote =
doc.GetElementsByTagName("ShipperNotification").Item(0).InnerText.Sanitize();
    Global.Library.Settings.ShipperNotification = shippernote;
    }
    catch
    {
        //PopNotify("Error", "");
        Page.SiteMaster().ShowError("Error adding Shipping Personnel");

    }

    UpdateAssetAdmin();
}
catch
{
    Page.SiteMaster().ShowError("Problem uploading library");
}

```

```

    }
    Push.Globals();
    Push.Library();

}

protected void UploadNotices_Click(object sender, EventArgs e)
{
    var name =
Server.MapPath("/App_Data/notice"+DateTime.Now.ToShortDateString().Replace("
", "").Replace("/", "-") + ".xml");
    NoticeUploader.PostedFile.SaveAs(name);
    var doc = new XmlDocument();
    using (StreamReader reader = new
StreamReader(NoticeUploader.PostedFile.InputStream))
    {
        if (reader.EndOfStream) return;
        doc.LoadXml(reader.ReadToEnd());
    }
    //check for nulls
    if (Global.Library.Settings == null) Global.Library.Settings = new Settings();
    if (Global.Library.Settings.Notifications_30_Day == null)
Global.Library.Settings.Notifications_30_Day = new List<AssetNotification>();
    if (Global.Library.Settings.Notifications_15_Day == null)
Global.Library.Settings.Notifications_15_Day = new List<AssetNotification>();

    XmlNodeList elemList = doc.GetElementsByTagName("Notice");
    foreach (XmlElement elem in elemList)
    {
        try
        {
            AssetNotification a = elem.ToNotification();
            EmailNotice notice = new EmailNotice();

            notice.Scheduled = DateTime.Now.AddDays(30);
            notice.NoticeType = NoticeType.Checkout;
            notice.NoticeAction = Global.CheckoutAction;

            notice.Assets.Add(a.AssetNumber);

            notice.NoticeControlNumber = a.EmailsCSV;

```

```

        notice.Body = Global.Library.Settings.NotificationMessage;
        notice.Subject = "Asset Return Reminder";
        notice.Emails.AddRange(a.Emails);
        notice.EmailAddress = a.Emails[0];
        Global.NoticeSystem.Add(notice);

    }
    catch { ShowError("Problem Adding Timed Notice"); }
}
Push.NotificationSystem();
UpdateAssetAdmin();
}

protected void UploadHistory_Click(object sender, EventArgs e)
{
    try
    {
        var name = Server.MapPath("/App_Data/history" +
DateTime.Now.ToShortDateString().Replace(" ", "").Replace("/", "-") + ".xml");
        HistoryUploader.PostedFile.SaveAs(name);
        var doc = new XmlDocument();
        using (StreamReader reader = new
StreamReader(HistoryUploader.PostedFile.InputStream))
        {
            if (reader.EndOfStream) return;
            doc.LoadXml(reader.ReadToEnd());
        }
        //check for nulls
        if (Global.Library.Assets == null)
        {
            //PopNotify("Error", "No Library loaded");
            return;
        }
        if (Global.Library.Assets.Count == 0)
        {
            //PopNotify("Error", "No Assests In Library");
            return;
        }

        XmlNodeList elemList = doc.GetElementsByTagName("HistoryItems");
        foreach (XmlElement elem in elemList)
        {

```

```

try
{
    var assetnumber = elem.GetAttribute("AssetNumber").Sanitize();
    var currentAsset = Global.Library.Assets.FindAssetByNumber(assetnumber);
    foreach (XmlElement subelem in elem.GetElementsByTagName("Asset"))
    {
        try
        {
            Asset a = subelem.ToAsset();
            a.IsHistoryItem = true;
            currentAsset.History.History.Add(a);
        }
        catch
        {
            //PopNotify("Error", "Adding History Item To Library Asset");
            return;
        }
    }
}
catch
{
    //PopNotify("Error", "Loading HistoryItems");
    return;
}

}
try
{
    //PopNotify("File Upload Complete", "History file upload complete");
}
catch { }
}
catch
{
    //PopNotify("Error", "History file upload Error");
}
}
}

```

```

protected void PopNotify(string caption, string content)
{

```

```

var script = @"
<script type='text/javascript'>
function Completed() {
    $.Notify({
        caption: '"+caption + @"',
        content: '" + content + @"'
    });
};
Completed();
</script>
";

```

```

Page.ClientScript.RegisterStartupScript(this.GetType(), "CallMyFunction", script);
Page.ClientScript.RegisterStartupScript(this.GetType(), "CallMyFunction", "Completed();",
true);
}

```

```

protected void SendSQL_Click(object sender, EventArgs e)
{
    try
    {
        SQL_Request req = new SQL_Request().OpenConnection();

        //request all assets
        req.GetAssets(false);

        //merge all assets

        //post merged assets as new master DB
        if (req.Tag != null)
        {
            var cloud = req.Tag as List<Asset>;
            //upload assets
            foreach (Asset a in Global.Library.Assets)
            {
                try
                {
                    var lookup = cloud.FindAssetByNumber(a.AssetNumber);
                    if (lookup == null)
                    {
                        req.AddAsset(a, false);

```



```

        }
        else
        if (lookup.AssetNumber == a.AssetNumber)
        {
            req.UpdateAsset(a, false);

        }
    }
    catch { //PopNotify("Error", "Error Pushing Library To SQL");
    }
}
//upload settings
//upload history
//upload notifications
//upload Calibrations

}
req.CloseConnection();

}
catch
{
    //PopNotify("Error", "Error Pushing Library To SQL");
}
//PopNotify("Complete", "Library Pushed To SQL");
}

```

```

protected void PullSQL_Click(object sender, EventArgs e)
{
    try
    {
        SQL_Request req = new SQL_Request().OpenConnection();

        //request all assets
        req.GetAssets(true);
        if (req.Tag != null)
        {
            //pull assets
            Global.Library.Assets = req.Tag as List<Asset>;
            //pull settings
            //pull notifications

```

```

        //pull calibrations
        //pull history
    }
}
catch
{
    //PopNotify("Error", "Error Pulling Library From SQL");
}
//PopNotify("Complete", "Library Pulled From SQL");
UpdateAssetAdmin();
}

```

```

protected void DeleteSQL_Click(object sender, EventArgs e)
{
    SQL_Request req = new SQL_Request().OpenConnection();
    try
    {

        //request all assets
        req.GetAssets(false);
        if (req.Tag != null)
        {
            var cloud = req.Tag as List<Asset>;

            foreach (var asset in cloud)
            {
                try
                {
                    req.DeleteAsset(asset.AssetId, false);
                }
                catch { //PopNotify("Error", "Error Deleting SQL Library");
                }
            }

            req.CloseConnection();
            Global.Library.Assets = new List<Asset>();
        }
    }
    catch
    {
        //PopNotify("Error", "Error Deleting SQL Library");
    }
}

```

```

    }
    finally
    {
        req.CloseConnection();
        //PopNotify("Complete", "SQL Library Deleted");
    }
}

```

```

}

```

```

protected void SaveUpsAcctBtn_Click(object sender, EventArgs e)

```

```

{
    if (Global.Library.Settings.UpsAccount == null)
    { Global.Library.Settings.UpsAccount = new UPSaccount(); }
    Global.Library.Settings.UpsAccount.P = ups_pwd.Value;
    Global.Library.Settings.UpsAccount.A = ups_aln.Value;
    Global.Library.Settings.UpsAccount.I = ups_userid.Value;
    Global.Library.Settings.UpsAccount.N = ups_shippernumber.Value;
    Push.Globals();
    UpdateAssetAdmin();
}

```

```

protected void SaveAllChangesBtn_Click(object sender, EventArgs e)

```

```

{
    Push.Library();
    Push.Globals();
    UpdateUsersAndRoles();
    UpdateAssetAdmin();
}

```

```

protected void DeleteNotice30DayBtn_Command(object sender, CommandEventArgs e)

```

```

{
    try
    {
        var rem = (from n in Global.NoticeSystem.Notices where n.Guid ==
e.CommandArgument as string select n).FirstOrDefault();
        if (rem != null)
        {
            var idx = Global.NoticeSystem.Notices.IndexOf(rem);
            Global.NoticeSystem.Notices.RemoveAt(idx);
        }
    }
}

```

```

        UpdateAssetAdmin();
    }
    catch {

        try
        {
            var rem = (from n in Global.NoticeSystem.Notices where n.Guid ==
e.CommandArgument as string select n).FirstOrDefault();
            if (rem != null)
            {
                var idx = Global.NoticeSystem.Notices.IndexOf(rem);
                Global.NoticeSystem.Notices.RemoveAt(idx);

            }
            UpdateAssetAdmin();
        }
        catch
        {

            return;
        }
    }
    Push.NotificationSystem();
}

```

```

protected void SendNotice30DayBtn_Command(object sender, CommandEventArgs e)
{
    var rem = (from n in Global.NoticeSystem.Notices where n.Guid == e.CommandArgument
as string select n).FirstOrDefault();
    //send then remove notice
    if (rem != null)
    {
        EmailHelper.SendNotificationSystemNotice(rem);
        if (!Global.Library.Settings.TESTMODE)
        {
            Global.NoticeSystem.Notices.Remove(rem);
        }
    }
    UpdateAssetAdmin();
}

```

```
protected void DeleteNotice15DayBtn_Command(object sender, CommandEventArgs e)
{
    UpdateAssetAdmin();
}
```

```
protected void SendNotice15DayBtn_Command(object sender, CommandEventArgs e)
{
    UpdateAssetAdmin();
}
```

```
protected void DeleteShippingPersonBtn_Command(object sender, CommandEventArgs e)
{
    try
    {
        var email = e.CommandName;
        EmailAddress em = null;
        foreach (var n in Global.Library.Settings.ShippingPersons)
        {
            if (n.Email == email)
            { em = n; }
        }
        Global.Library.Settings.ShippingPersons.Remove(em);
        Push.Globals();
        UpdateAssetAdmin();
    }
    catch { Page.SiteMaster().ShowError("Problem removing Shipping Personnel"); }
}
```

```
protected void DeleteEngineer_Command(object sender, CommandEventArgs e)
{
    if (e.CommandName == "DeleteEngineer")
    {
        try
        {
            var name = e.CommandArgument;
            EmailAddress em = null;
            foreach (var n in Global.Library.Settings.ServiceEngineers)
            {
                if (n.Name == name)
                { em = n; }
            }
            Global.Library.Settings.ServiceEngineers.Remove(em);
        }
    }
}
```

```

        Push.Globals();
        UpdateAssetAdmin();
    }
    catch { Page.SiteMaster().ShowError("Problem removing Shipping Personnel"); }

}
UpdateAssetAdmin();
}

```

```

protected void CustomerRepeater_ItemCommand(object source,
RepeaterCommandEventArgs e)
{
    if (e.CommandName == "DeleteCustomer")
    {
        try
        {
            var split = (e.CommandArgument as string).Split('|');
            var name = split[0];
            var addy = split[1];
            Customer em = null;
            foreach (var n in Global.Library.Settings.Customers)
            {
                if (n.CompanyName == name && n.Address == addy)
                { em = n; }
            }
            Global.Library.Settings.Customers.Remove(em);
            Push.Globals();
            UpdateAssetAdmin();
        }
        catch { Page.SiteMaster().ShowError("Problem removing Shipping Personnel"); }

    }
    UpdateAssetAdmin();
}

```

```

protected void SendSettingsSQL_Click(object sender, EventArgs e)
{
    AssetController.PushSettings(Global.Library.Settings);
    //PopNotify("Success", "Settings Pushed To SQL");
    UpdateAssetAdmin();
}

```

```
protected void PullSettings_Click(object sender, EventArgs e)
{
    Global.Library.Settings = AssetController.GetSettings();
    //PopNotify("Success", "Settings Pulled From SQL");
    UpdateAssetAdmin();
}
```

```
protected void DeleteSettingsSQL_Click(object sender, EventArgs e)
{
    try
    {
        SQL_Request req = new SQL_Request().OpenConnection();
        req.SettingsDelete();
        Global.Library.Settings = new Settings();
    }
    catch { }
}
```

```
protected void AssetPeopleManager_Click(object sender, EventArgs e)
{
    AdminMultiView.ActiveViewIndex = 2;
    UpdateAssetAdmin();
}
```

```
protected void NotificationManagerBtn_Click(object sender, EventArgs e)
{
    AdminMultiView.ActiveViewIndex = 4;
    UpdateAssetAdmin();
}
```

```
protected void SaveCheckOutMsgBtn_Click(object sender, EventArgs e)
{
    var a = checkoutmsgbox.Value;
    Global.Library.Settings.CheckOutMessage = a;
    Push.Globals();
}
```

```
protected void SaveCheckInMsgBtn_Click(object sender, EventArgs e)
{
    Global.Library.Settings.CheckInMessage = checkinmsgbox.Value;
    Push.Globals();
}
```

```

}

protected void SaveNoticMsgBtn_Click(object sender, EventArgs e)
{
    Global.Library.Settings.NotificationMessage = notificationmsgbox.Value;
    Push.Globals();
}

protected void SaveShipperMsgBtn_Click(object sender, EventArgs e)
{
    Global.Library.Settings.ShipperNotification = shipmsgbox.Value;
    Push.Globals();
}

private string AppendDir(string dir, string file)
{
    return dir + file;
}

protected void ExportLibrary_Click(object sender, EventArgs e)
{
    try
    {
        GC.Collect();
        var libraryname = Server.MapPath("/Zip/lib.xml");
        var noticename = Server.MapPath("/Zip/notice.xml");
        var settingname = Server.MapPath("/Zip/settings.xml");
        Global.Library.Settings.SerializeToXmlFile(Global.Library.Settings, settingname);
        Global.Library.SerializeToXmlFile(Global.Library, libraryname);
        Global.NoticeSystem.SerializeToXmlFile(Global.NoticeSystem, noticename);
        var temp = "/Zip/" + Guid.NewGuid().ToString() + ".zip";
        temp = Server.MapPath(temp);
        var files = new List<string>() { libraryname, noticename, settingname };

        var images = (from i in Directory.GetFiles(Server.MapPath("/Account/Images")) select
i).ToList();
        var labels = (from i in Directory.GetFiles(Server.MapPath("/Account/Labels")) select
i).ToList();
        var packing = (from i in Directory.GetFiles(Server.MapPath("/Account/PackingLists"))
select i).ToList();
        var PdfFiles = (from i in Directory.GetFiles(Server.MapPath("/Account/PdfFiles")) select
i).ToList();
        var Receiving = (from i in Directory.GetFiles(Server.MapPath("/Account/Receiving"))
select i).ToList();

```



```

        var templates = (from i in Directory.GetFiles(Server.MapPath("/Account/Templates"))
select i).ToList();

        bool result = false;
        try
        {
            using (var memoryStream = new MemoryStream())
            {
                using (var archive = new ZipArchive(memoryStream, ZipArchiveMode.Create,
true))
                {
                    foreach (var file in files)
                    {
                        var name = file;
                        archive.CreateEntryFromFile(file, Path.GetFileName(name));
                    }
                    foreach (var file in images)
                    {
                        var name = file;
                        archive.CreateEntryFromFile(file, "Images/" + Path.GetFileName(name));
                    }
                    foreach (var file in labels)
                    {
                        var name = file;
                        archive.CreateEntryFromFile(file, "Labels/" + Path.GetFileName(name));
                    }
                    foreach (var file in packing)
                    {
                        var name = file;
                        archive.CreateEntryFromFile(file, "PackingSlips/" +
Path.GetFileName(name));
                    }
                    foreach (var file in PdfFiles )
                    {
                        var name = file;
                        archive.CreateEntryFromFile(file, "PdfFiles/" + Path.GetFileName(name));
                    }
                    foreach (var file in Receiving)
                    {
                        var name = file;
                        archive.CreateEntryFromFile(file, "Receiving/" + Path.GetFileName(name));
                    }
                }
            }
        }
    }
}

```

```

        foreach (var file in templates)
        {
            var name = file;
            archive.CreateEntryFromFile(file, "Templates/" + Path.GetFileName(name));
        }

    }

    using (var fileStream = new FileStream(temp, FileMode.Create))
    {
        memoryStream.Seek(0, SeekOrigin.Begin);
        memoryStream.CopyTo(fileStream);
    }
}
result = true;
}
catch { result = false; }
if (!result)
{
    GC.Collect();
    GC.Collect();
    Page.SiteMaster().ShowError("Problem exporting Library file...");
}
else
{
    GC.Collect();
    GC.Collect();
    string fileName = String.Empty;

    string filePath = String.Empty;

    filePath = temp;

    fileName = System.IO.Path.GetFileName(filePath);
    try
    {
        HttpContext.Current.Response.ClearContent();

        HttpContext.Current.Response.ClearHeaders();

        HttpContext.Current.Response.AddHeader("Content-Disposition", "inline;
filename=" + fileName);
    }
}

```

```

        HttpContext.Current.Response.ContentType = "application/zip";

        HttpContext.Current.Response.WriteFile(filePath);
        GC.Collect();
        HttpContext.Current.Response.End();
    }
    catch { GC.Collect(); }
}
catch
{
    Page.SiteMaster().ShowError("Problem exporting Library file...");
    GC.Collect();
}
}

```

```

protected void CreateShipper_Click(object sender, EventArgs e)
{
    EmailAddress email = new EmailAddress();
    email.Name = ShipperNameInput.Text;
    email.Email = ShipperEmailInput.Text;
    Global.Library.Settings.ShippingPersons.Add(email);
    Push.Globals();
}

```

```

protected void CreateEngineerBtn_Click(object sender, EventArgs e)
{
    EmailAddress email = new EmailAddress();
    email.Name = EngineerNameInput.Text;
    email.Email = EngineerEmailInput.Text;
    Global.Library.Settings.ServiceEngineers.Add(email);
    Push.Globals();
}

```

```

protected void CreateCustomerBtn_Click(object sender, EventArgs e)
{
    Customer cust = new Customer();
    cust.CompanyName = SprCompany.Text;
    cust.Address = SprAddr.Text;
    cust.Address2 = SprAddr2.Text;
    cust.Attn = SprName.Text;
}

```

```

    cust.City = SprCompany.Text;
    cust.State = SprState.Text;
    cust.Postal = SprPostal.Text;
    cust.Country = SprCountry.Text;
    cust.EmailAddress = new EmailAddress() { Email = SprEmail.Text, Name= cust.Attn };
    cust.Phone = SprPhone.Text;
    Global.Library.Settings.Customers.Add(cust);
    Push.Globals();
}

```

```

protected void AddStatic_Click(object sender, EventArgs e)
{
    AddStaticPlaceholder.Visible = true;
    staticupdatepanel.Update();
}

```

```

protected void DeleteStatic_Command(object sender, CommandEventArgs e)
{
    try
    {
        var idx = Global.Library.Settings.StaticEmails.FindIndex(x => x.Email ==
(string)e.CommandArgument);
        Global.Library.Settings.StaticEmails.RemoveAt(idx);
    }
    catch (Exception)
    {

    }
}

public void BindStaticEmails()
{
    StaticEmailRepeater.DataSource = Global.Library.Settings.StaticEmails;
    StaticEmailRepeater.DataBind();
}

protected void AddStaticSubmit_Click(object sender, EventArgs e)
{
    var controls = staticupdatepanel.Controls[0].Controls[1].Controls;
    var a = 0;
    var name = (controls[1] as HtmlInputText).Value;
    var email = (controls[3] as HtmlInputText).Value;
}

```

```
EmailAddress stat = new EmailAddress() { Name=name, Email=email };
```

```
if (EmailHelper.IsValid(stat))
```

```
{
```

```
    Global.Library.Settings.StaticEmails.Add(stat);
```

```
    Push.Globals();
```

```
    BindStaticEmails();
```

```
    staticupdatepanel.Update();
```

```
}
```

```
else
```

```
{
```

```
    ShowError("Email is not valid");
```

```
}
```

```
}
```

```
protected void ChangeTestModeBtn_Click(object sender, EventArgs e)
```

```
{
```

```
    Global.Library.Settings.TESTMODE = !Global.Library.Settings.TESTMODE;
```

```
    Push.Globals();
```

```
    TestModeLabel.Text = Global.Library.Settings.TESTMODE.ToString();
```

```
}
```

```
}
```

```
}
```

[illegible]

[illegible]

CtrNjgvd0NxJiN4QTt1S3FVeUxaeHpXc1Jaa2RyWWM1V2FWeDIZbDIKdmpjc3hvQIZIUk5QbFFZc
XEvNGZzUDhBZnQ1LzBuWG4vVIhGWGY0ZnNQOEFmdDUvJiN4QTswblhuL1ZYRIhmNGZzUD
hBZnQ1LzBuWG4vVIhGWGY0ZnNQOEFmdDUvMG5Ybi9WWEZYZjRmc1A4QWZ0NS8wblhuL1
ZYRIhmNGZzUDhBJiN4QTtmdDUvMG5Ybi9WWEZYZjRmc1A4QWZ0NS8wblhuL1ZYRIhmNGZz
UDhBZnQ1LzBuWG4vVIhGVmtGdWw1YIJ3U2xsU09XNCTLTm1qYytsJiN4QTtLMFFQcUtRNEpC
cXhVZ2sreEILcS93RHcvWWY3OXZQK2s2OC82cTRxNy9EOWgvdjl4LzZUcnovcXJpcnY4UDJIKy
9iei9wT3ZQK3F1JiN4QTtLdS93L1ImNzI2UCtrNjgvNnE0cTcvRDloL3YyOC82VHJ6L3FyaXJ2OFA
ySCsvYnovcE92UCtxdUt1L3cvWWY3OXZQK2s2OC82cTRxJiN4QTswL2w2eVpHVVMzUUxLUU
M5MWN5cnVLYnh5U09qai9KWINEM0dLci9MMY9IQTb6L21FZy81TnJpcVIZcTdGWFIxN0ZYWX
E3RIhZcWxXJiN4QTtwZjM4bit0WWY5Ulp4Vk5jVmRpcnNWZGlyc1ZkaXJzVmRpcVhhWDluL25yZ
WY5Ukp4Vk1jVmRpcnNWZGlyc1ZkaXJzVmRpcVgrWHYrJiN4QTtPQnBuL01KQi93QW0xeFZNT
VZkaXJzVmRpcnNWZGlyc1ZTclV2NytUL1dzUCtvczRxbXVLdXhWMkt1eFYyS3V4VjJLdXhWTHR
MK3ovJiN4QTtBTtliei9xSk9LcGppcnNWZGlyc1ZkaXJzVmRpcnNWU0xTWXRhZIM3TjdTNNrvYI
Y0STJnaGx0cEplU05rQIZDNnp4aHVJMnJ4R0tvJiN4QTt2NnY1ay81YjdQOEE2UTVmK3lyRIhmVi
9Nbi9MZlovOUljdI9BR1ZZcTc2djVrLzViN1AvQUtRNWYreXJGWGZWL01uL0FDMzJmL1NIJiN4Q
TtMLzJWWXE3NnY1ay81YjdQL3BEbC93Q3lyRIhmVi9Nbi9MZlovD0RTSEwvMIZZcTc2djVrL3dD
Vyt6LzZRNWYreXJGVkdaWjBpdUk3JiN4QTt5UlpidG50dUU4U2NFWG5LRmlLb3pPZmdrcXhET
WErTk5ncXJmVi9Nbi9MZFOvOEFTSEwvQU5sV0t1K3lrWIARvYt6L0FPa09YL3NxxJiN4QTt4VjMxZ
npKL3kzMmYvU0hML3dCbFdLdStyK1pQK1crei93Q2tPWC9zcXhWMzFmekovd0F0OW4vMGh5L
zlsV0t1K3lrWIARvYt6LzZRJiN4QTs1ZjhBc3F4VjMxZnpKL3kzMmY4QTB0eS85bFdLdStyK1pQOE
FsdXMvK2tPWC9zcXhWU3QwdW5zNFV0SkZpdVVsbkVrOGk4MvIwS3I5JiN4QTtNWXdWcjZqL0
FCY1F3cDQ3VUtxcjlYOHImOHQ5bi8waHkvOEFAvmlydnEvbVQvbHZzLzhBcERsLzdLc1ZkOVg4
eWY4QUxmWi85SWN2JiN4QTsvWIZpcnZxL21UL2x2cy8ra09YL0FMS3NWZDIYOHImOHQ5bi9B
TkIjdI9aVmlydnEvbVQvQUpiN1AvcERsLzdLc1ZkOVg4eWY4dDluJiN4QTsvd0JJY3Y4QTJWWXF
0a3RmTWJveXRmV3BCQkJFZHRKRTU5bGtNOHZBL3dDVndOUERGVR5OS94d05NLzVoSVAr
VGE0cW1HS3V4VjJLJiN4QTt1eFYyS3V4VjJLcFzXWDkvSi9yV0gvVvdjVIRYRIhZcTdGWFIxN0Z
YWXE3RIhZcWwybC9aL3dDZXQ1LzFFbkZVeHhWMkt1eFYyS3V4JiN4QTtWMkt1eFYyS3BmNW
Uvd0NPQnBuL0FEQ1FmOG0xeFZNTVZkaXJzVmRpcnNWZGlyc1ZTclV2NytUL0FGckQvcUxPS
3ByaXJzVmRpcnNWJiN4QTtkaXJzVmRpcnNWUzdTL3MvODliei9xSk9LcGppcnNWZGlyc1ZkaXJ
zVmRpcnNWU0xTWmRhVFM3TxxTMXRwclZJSTFnbWx1WkxzU5VJiN4QTtBVnlpd1NCZVEzc
HIPS292Nng1ay93Q1dDei82VEpmK3IYRIhmV1BNbi9MQlovOEFTWkwvQU5rdUt1K3NIWIARV0N
6L0FPa3IYL3NsJiN4QTt4VjMxanpKL3l3V2YvU1pML3dCa3VLdStzZVpQK1dDei93Q2t5WC9zbHh
WMzFqekovd0FzRm4vMG1TLzlrUdUt1K3NIWIARV0N6LzZUJiN4QTtKZjhBc2x4VIJtYWQ0cmITOGp
XSzdWN2JoQkUvTkc0U2hvZ3JzcUg0NUtxU3lpbmhUY3FxmZfQekoveXcyZjhBMG1TLzlrUdUt1K3
NIJiN4QTtaUDhBbGdzLytreVgvc2x4VjMxanpKL3l3V2YvQUVtUy93RFPmaXJ2ckhtVC9sZ3Mvd0R
wTWwvN0pjVmQ5WTh5ZjhzRm4vMG1TLzhBJiN4QTtaTGlydnJlbVQvbGdzLzhBcE1sLzdKY1Zk
OVk4eWY4QUxCWi85Smt2L1pMaXJ2ckhtVC9saHMvK2t5WC9BTEpjVIVyZDdsTE9GN1NOJiN4
QTtaYmw1Wnk4RWpjRIV2S3pTS1pBR3A2Yi9BQThncHI0YjFDCXI5WTh5ZjhzRm4vMG1TLzhBW
kxpcnZySG1UL2xncy84QXBNbC83SmNWJiN4QTtkOVk4eWY4QUxCWi85Smt2L1pMaXJ2ckhtV
C9sZ3MvK2t5WC9BTEpjVmQ5WTh5ZjhzRm4vQU5Ka3YvWkxpcnZySG1UL0FKWUxQL3BNJiN4
QTtsLzdKY1ZkOVk4eWY4c0ZuL3dCSmt2OEEyUzRxdGt1dk1hSXpOWTJvQUJKTWR6Sks0OTF
qTUVYTS93Q1R6RmZIRIZUeTkveHdOTS81JiN4QTtoSVArVGE0cW1HS3V4VjJLdXhWMkt1eFY

yS3BWcVg5L0ovcldIL1VXY1ZUWEZYWXE3RIhZcTdGWFlxN0ZYWXXFsMmwvWi93Q2V0NS8xJi
N4QTtFbkZVeHhWMkt1eFYyS3V4VjJLdXhWMktwZjVIL3dDT0Jwbi9BRENRZjhtMXhWTU1WZGly
c1ZkaXJzVmRpcnNWU3JVdjcrVC9BRnJEJiN4QTsvUxPS3ByaXJzVmRpcnNWZGlyc1ZkaXJzVI
M3Uy9zLzg5YnovcUpPS3BqaXJzVmRpcnNWZGlyc1ZkaXJzVINQU3RXdExUUzdLMWtTJiN4QT
s1ZVNDQ0tOMml0Ym1XTWxVQUptUkkyUmw4Q3BvY1ZSWCtJTEQvZiY1L3dCSU41LzFTEFYz
K0IMRC9mVjUvd0JJTjUvMVN4VjMrSUxEJiN4QTsvZiY1L3dCSU41LzFTEFYzK0IMRC9mVjUvd0J
JTjUvMVN4VjMrSUxEL2ZWNS93QKlONS8xU3hWMytJTEQvZiY1L3dCSU41LzFTEFYzJiN4QTsr
SUxEL2ZWNS93QKlONS8xU3hWU21kYnIPYTzPREtpTmJlaEtyUk9mcTh2cXQ4RGhXRIFhTHIBc
jhxSEZWWC9FRmgvdnE4LzZRYnovJiN4QTtBS3BZcTcvRUZoL3ZxOC82UWJ6L0FLcFlxNy9FR
mgvdnE4LzZRYnovQUtwWXE3L0VGaC92cTgvNIFiei9BS3BZcTcvRUZoL3ZxOC82JiN4QTtRYno
vQUtwWXE3L0VGaC92cTgvNIFiei9BS3BZcTcvRUZoL3ZxOC82UWJ6L0FLcFlxNy9FRmgvdnE4L
zZRYnovQUtwWXXfZ3VFcZdhJiN4QTtPZVvNeVNTM0h3eHEwamoxWldsQTIOUVhKQUZHQ2dr
SDJCSVZYLzRnc1A4QWZWNS8wZzNuL1ZMRlhmNGdzUDhBZiY1LzBnM24vVxkGJiN4QTtYzJR
nc1A4QWZWNS8wZzNuL1ZMRlhmNGdzUDhBZiY1LzBnM24vVxkGWGY0Z3NQOEFmVjUvMGc
zbi9WTEZYzJRnc1A4QWZWNS8wZzNuJiN4QTsvVxkGWGY0Z3NQOEFmVjUvMGczbi9WTEZXb
jh3MINvekNLNkpWU1FIdGJtSmRoWGVtU05FUWY1VE1BTzV4VmY1ZS80NEdtZjh3a0gvJiN4Q
TtBQ2JYRIV3eFYyS3V4VjJLdXhWMkt1eFZLdFQvQUw5Lzlldy82aXppcWE0cTdGWFlxN0ZYWX
E3RIhZcTdGVXUwdjdQL0FEMXZQK29rJiN4QTs0cW1PS3V4VjJLdXhWMkt1eFYyS3V4VkwvTD
MvQUJ3Tk0vNWhJUDhBazJ1S3BoaXJzVmRpcnNWZGlyc1ZkaXFWYwWvZnlmNjFoLzFGJiN4Q
TtuRIUxeFYyS3V4VjJLdXhWMkt1eFYyS3BkcGYyZitldDUvMUvURIV4eFYyS3V4VjJLdXhWMkt1e
FYyS3BkNWRaVG9kZ2dJTHd3UIJUJiN4QTtLRHVraUIBeU1PektkaURpcVk0cTdGWFlxN0ZYWX
E3RIhZcWxOODZTdfBMR3I0RkZKYUxLNEk0cVliam5MeVBiz3BxM2hpcWFvNINJJiN4QTtyb3dk
SEFaV1UxQklzQkJHS3Q0cTdGWFlxN0ZYWXE3RItdWIVvc3hDcW9xekhZQUR1Y1ZTeXduaGhn
am1sY0pGTE5jaU9RbWi0NnR3JiN4QTt6UmtlcFJ4OWs5L3B4Vk5NVmRpcnNWZGlyc1ZkaXJzVI
U3aTRodDRXbW1jSkd0S3NmYzBBQUc1Sk93QTZuRIVuMGJSdEh1ZEhzYm01JiN4QTtzYmVhN
G10NHBKcHBJa1ozZGtCWm1ZZ2trazFKT0tvei9EMmdmOVd5MC81RVlvOEFOT0t1L3dBUGFC
LzFiTFQvQUpFUi93RE5PS3UvJiN4QTt3OW9IL1ZzdFArUkVmL05PS3UvdzlvSC9BRmJMVC9rU
kgvelRpcnY4UGFCLzFiTFQva1JIL3dBMDRxNy9BQTIvSC9Wc3RQOEFrUkgvJiN4QTtBTtA0cTc
vRDJnZjIXeTAvNUVSLzgwNHfocm0zdDdWWmJXMmlTRzJaclRsQWlxcUgxcmpoSjhJRIBpVVVP
S29uL0QyZ2Y5V3kwLzVFJiN4QTtSLzhBTk9LdS93QVBhQi8xYkxUL0FKRVIvd0ROT0t1L3c5b0g
vVnN0UCtSRWYvTk9LdS93OW9IL0FGYkxUL2tSSC96VGlydjhQYUlvJiN4QTsxYkxUL2tSSC93Q
TA0cTcvQUE5b0gvVnN0UDhBa1JIL0FNMDRxNy9EMmdmOVd5MC81RVlvODA0cTcvRDJnZjhB
VnN0UCtSRWYvTk9LJiN4QTtxVm5iVzF6YXJiWEVTeTI2UzNJV0Z3Q2xJNTJSQnhPMUZYWUR
0aXFyL2g3UVArcLphZjhBSWIQL0FKcHhWMytldEEvNnRscC95SWovJiN4QTtBT2FjVmQvaDdRU
CtyWmFmOGIJLythY1ZkL2g3UVA4QXEyV24vSWIQL21uRIhmNGUwRC9xMlduL0FDSWovd0Nh
Y1ZkL2g3UVArcLphJiN4QTtmOGIJL3dEbW5GWGY0ZTBEL3EYV24vSWIQL21uRld4NWywSUdx
NmRiSXc2T2tTS3dQaUdVQWcrNHhWcnk5L3h3Tk0vNWhJUCtUYTRxJiN4QTtR0t1eFYyS3V4Vj
JLdXhWMktwVnFYOS9KL3JXSC9VV2NWVFhGWFlxN0ZYWXE3RIhZcTdGWFlxbDJsL1ovd0Nld
DUvMUvURIV4eFYyJiN4QTtLdXhWMkt1eFYyS3V4VjJLcGY1ZS93Q09CcG4vQURDUWY4bTF4
Vk1NVmRpcnNWZGlyc1ZkaXJzVINyVWgrL2YvQUZyRC9xTE9LcHJpJiN4QTtyc1ZkaXJzVmRpcn
NWZGlyc1ZTN1Mvcy84OWJ6L3FKT0twamlyc1ZkaXJzVmRpcnNWZGlyc1ZTUFN0SnRMdlM3Sz
ZrZTVTU2VDS1lxJiN4QTtpdXJtS01Ga0JJU05KRIJWOEFvb01WUigrSDdEL2Z0NS93QkoxNS8x

VnhWMytIN0QvZnQ1L3dCSjE1LzFWeFYzK0g3RC9mdDUvd0JKJiN4QTsxNS8xVnhWMytIN0Qv
ZnQ1L3dCSjE1LzFWeFYzK0g3RC9mdDUvd0JKMTUvMVZ4VjMrSDdEL2Z0NS93QkoxNS8xVnh
WMytIN0QvZnQ1JiN4QTsvd0JKMTUvMVZ4VINtUmJPT2ExaUxNanRiRG5LeIN1UHJFdnBOOGJ
sbU5BS3J5SnA4cURGVlgvRDloL3YyOC82VHJ6L0FLcTRxNy9EJiN4QTs5aC92MjgvNIRyei9BS3
E0cTcvRDloL3YyOC82VHJ6L0FLcTRxNy9EOWgvdjI4LzZUcnovQUtxNHE3L0Q5aC92MjgvNIRye
i9BS3E0JiN4QTtxNy9EOWgvdjI4LzZUcnovQUtxNHE3L0Q5aC92MjgvNIRyei9BS3E0cTcvRDloL3
YyOC82VHJ6L0FLcTRxc2d0MHZMYU9DVXNxUnkzJiN4QTtleFJzMGJuMHBXaUI5UINIQklOV0tr
RW4ySkJWWC80ZnNQOEFmdDUvMG5Ybi9WWEZYZjRmc1A4QWZ0NS8wblhuL1ZYRIhmNGZ
zUDhBjN4QTtmdDUvMG5Ybi9WWEZYZjRmc1A4QWZ0NS8wblhuL1ZYRIhmNGZzUDhBZnQ1L
zBuWG4vVihGWGY0ZnNQOEFmdDUvMG5Ybi9WWEZYZjRmJiN4QTtzUDhBZnQ1LzBuWG4vVI
hGV244dldUSXlpVzZCWINBWHVybVZkeFRIT1NSMGNmNUxLUWU0eFZmNWUvNDRHbWY4d2
tlL0FDYIhGVXd4JiN4QTtWMkt1eFYyS3V4VjJLdXhWS3RTL3Y1UDlhdY82aXppcWE0cTdGWFIx
N0ZYWXE3RIhZcTdGVXUwdjdQOEF6MXZQK29rNHftT0t1eFYyJiN4QTtLdXhWMkt1eFYyS3V4
VkwvQUM5L3h3Tk0vd0NZU0QvazJ1S3BoaXJzVmRpcnNWZGlyc1ZkaXFWYwWvZnlmNjFoL3dC
Ulp4Vk5jVmRpJiN4QTtyc1ZkaXJzVmRpcnNWZGlxWGFYOW4vbnJIZjhBVVNjVIRIRihZcTdGWFI
xN0ZYWXE3RIhZcWtXa3hhMCtsMmlybDFiUTJyd1J0QkRMJiN4QTtiU1NP a2JJQ3FGMW5qRGN
SdFhpTVZSZjFmekoveTMyZi9TSEwvd0JsV0t1K3lrWIArVyt6L3dDa09YL3NxeFYzMWZ6Si93QX
Q5bi8wJiN4QTtoeS85bFdLdStyK1pQK1crei82UTVmOEFzcXhWMzFmekoveTMyZjhBMGh5LzlsV
0t1K3lrWIA4QWx2cy8ra09YL3NxeFYzMWZ6Si95JiN4QTszMmYvQUVoeS93RFpWaXFqTXM2UI
hFZDVJc3QyejIzQ2VKT0NMemxDeEZVWm5Qd1NWWWhtTmZHbXdWVnZxL21UL2x1cy93RHB
EbC83JiN4QTtLc1ZkOVg4eWY4dDluLzBoeS84QVpWaXJ2cS9tVC9sdnMvOEFwRGwvN0tzVmQ
5WDh5ZjhBTGZaLzIJY3YvWIZpcnZxL21UL2x2cy8rJiN4QTtrT1gvQUxLc1ZkOVg4eWY4dDluL0F
OSWN2L1pWaXJ2cS9tVC9BSml3UC9wRGwvN0tzVmQ5WDh5Zjh0MW4vd0JJY3Y4QTJWWXF
wVzZYJiN4QTtUMmNLV2tpeFhLU3ppU2VSZWFzVWxaWkdNWUsxOVIvaTRoaFR4Mm9WVlg2d
jVrLzViN1AvQUtRNWYreXJGWGZWL01uL0FDMzJmL1NIJiN4QTtMLzJWWXE3NnY1ay81YjdQL
3BEbC93Q3lyRIhmVi9Nbi9MZlovd0RTSEwvMIZZcTc2djVrL3dDVyt6LzZRNWYreXJGWGZWL01u
L0xmJiN4QTtaLzhBU0hML0FObFdLdStyK1pQK1crei9BT2tPWC9zcXhWYkphK1kzUmxhK3RTQ0
NDSTdhU0p6N0xJWjVIQI95dUJwNFlxcWVYdjhBjN4QTtqZ2FaL3dBd2tlL0p0Y1ZUREZYWXE3R
lhZcTdGWFIxN0ZVcTFMKy9rL3dCYXcvNml6aXFhNHE3RIhZcTdGWFIxN0ZYWXE3RIV1MHY3Ji
N4QTtQL1BXOC82aVRpcVk0cTdGWFIxN0ZYWXE3RIhZcTdGVXY4dmY4Y0RUUCtZU0QvazJ1
S3BoaXJzVmRpcnNWZGlyc1ZkaXFWYwWvZnlmJiN4QTs2MWgvMUZuRIUxeFYyS3V4VjJLdXh
WMkt1eFYyS3BkcGYyZitldDUvMUVuRIV4eFYyS3V4VjJLdXhWMkt1eFYyS3BGcE11dEpwVm1sJ
iN4QTtwYTIwMXFrRWF3VFMzTWtUdkdxQUs1UVFTQmVRM3B5Tk1WUmYxanpKL3dBc0ZuLzB
tUy85a3VLdStzZVpQK1dDei82VEpmOEFzbHhWJiN4QTszMWp6Si95d1dmOEEwbVMvOWt1S3
Urc2VaUDhBbGdzLytrevgvc2x4VjMxanpKL3l3V2YvQUVtUy93RFpMaXJ2ckhtVC9sZ3Mvd0RwJi
N4QTtNbC83SmNWZDIZOHImOHNGbi8wbVMvOEFA TGlxak0wN3hYRWw1R3NWMnlyM0NDSit
hTndsRFJCWFpVUHH5VIVsbEZQQ201VIZ2ckhtJiN4QTtUL2xocy84QXBNbC83SmNWZDIZOHIm
OEFMQlovOUprdi9aTGlydnJlbVQvbGdzLytrevgVQUxKY1ZkOVk4eWY4c0ZuL0FOSmt2L1pMJiN
4QTtpcnZySG1UL0FKWUxQL3BNbC83SmNWZDIZOHImOHNGbi93QkprdjhBMIM0cTc2eDVRlZV
ZTFA4QTZUSmYreVhGWGZXUE1uL0xEWi85JiN4QTtKa3YvQUdTNHFwVzcsS1djTDJrYXkzTHI
6bDRKRzRLcGVWbWtVeUFOVDaZK0hrRk5mRGVvVIZmckhtVC9BSIIMUC9wTWwvN0pjVmQ5Ji
N4QTtZOHIImOHNGbi93QkprdjhBMIM0cTc2eDVRlZVZTFA4QTZUSmYreVhGWGZXUE1uL0xCW

i85Smt2L0FHUzRxNzZ4NWsvNVIMUC9BS1RKJIN4QTtmK3lYRIhmV1BNbi9BQ3dXZI9TWkwvMI
M0cTc2eDVRlZVZTFAvceE1sL3dDeVhGVnNsMTVqUkdackcxQUFKSmp1WkpYSHVzWmdpNW4v
JiN4QTtBQ2VZcjQ0cXFIWHYrT0Jwbi9NSklveWJYRIV3eFYyS3V4VjJLdXhWMkt1eFZLdFMvdjVQ
OWF3LzZpemlxYTRxN0ZYWXE3RIhZcTdGJiN4QTtYWXE3RIV1MHY3UC9BRDF2UCtvazRxbU9
LdXhWMkt1eFYyS3V4VjJLdXhWTC9MMY9BQndOTS81aEIQOEFrMnVLcGhpcnNWZGlyc1ZkJiN
4QTtpcnNWZGLxVmFsL2Z5ZjYxaC8xRm5GVTF4VjJLdXhWMkt1eFYyS3V4VjJLcGRwZjJmK2V0
NS8xRW5GVXh4VjJLdXhWMkt1eFYyS3V4JiN4QTtWMktwSHBXcldscHBkbGF5SmN2SkJCRCk3
Uld0ekxHU3FBRXBja2JJJeStCVTBPS29yLOVGaC92cTgvNIFieI9xbGlydjHrv0grK3J6JiN4QTsvcE
J2UCtxV0t1L3hCWWY3NnZQK2tHOC82cFlxNy9FRmgvdnE4LzZRYnovcWxpcnY4UVdlKytyei9w
QnZQK3FXS3UveEJZZjc2dlArJiN4QTtrRzgvNnBZcTcvRUZoL3ZXOC82UWJ6L3FsaXFSTTYza2
MxMUHVHvkVhMIBDVldPy9WNWZWYjRIQ3NLZzBYa0JYNVVPS3F2OEFpQ3cvJiN4QTszMWV
mOUIONS93QIVzVmQvaUN3LzMxZWY5SU41L3dCVXNWZC9pQ3cvMzfIZjIJTjUvd0JVc1ZkL2ID
dy8zMWVmOUIONS93QIVzVmQvJiN4QTtpQ3cvMzfIZjIJTjUvd0JVc1ZkL2IDdy8zMWVmOUIONS
93QIVzVmQvaUN3LzMxZWY5SU41L3dCVXNWZC9pQ3cvMzfIZjIJTjUvd0JVJiN4QTtzVldRWEN
XZHRIUEtHWkpKYmo0WTFhUng2c3JTZ2Vtb0xrZ0NqQIFTRDDBa0tyLzhRV0grK3J6L0FLUWJ6L
3FsaXJ2OFFXSCsrcnovJiN4QTtBS1FieI9xbGlydjHrv0grK3J6L0FLUWJ6L3FsaXJ2OFFXSCsrcn
ovQUtRYnovcWxpcnY4UVdlKytyei9BS1FieI9xbGlydjHrv0grJiN4QTsrcnovQUtRYnovcWxpcnY4U
VdlKytyei9BS1FieI9xbGlyVCtZYkpVWmhGEVxcEIEEMnR6RXV3cnZKSskdpSVA4cG1BSGM0cXY
4dmY4JiN4QTtjRFRQK1ITRC9rMnVLcGhpcnNWZGlyc1ZkaXJzVmRpcVZhbi9BSDcvQU92WWY
5UlP4Vk5jVmRpcnNWZGlyc1ZkaXJzVmRpcVhhWDluJiN4QTsvbnJIJzISSnhWTWNWZGlyc1ZkaX
JzVmRpcnNWZGLxWCtYditPQnBuL01KQi93QW0xeFZNtVZkaXJzVmRpcnNWZGlyc1ZTclV2Nyt
UJiN4QTsvV3NQK29zNHftUt1eFYyS3V4VjJLdXhWMkt1eFZMdEwrei9BTtliei9xSk9LcGppcnN
WZGlyc1ZkaXJzVmRpcnNWUzd5NnlUTdCJiN4QTtBUVhoZ2lpbVVIZEpFUUJrWWRtVTdFSEZV
eHhWMkt1eFYyS3V4VjJLdXhWS2I1MGxhZVdObGFLS1MwV1Z3UnhVdzNIT1hrZTNCFVF2JiN4
QTtERIUXujBrUIhSZzZPQXIzcHFDRHVDQ01WYnhWMkt1eFYyS3V4VjJLdE15b3BaaUZWUIztT3
dBSGM0cWxsafBEREJITks0U0tXYTVfJiN4QTtjaE5GYjFiaG1qSVBTamo3SjcvGlXyVIxN0ZYW
XE3RIhZcTdGWFlxcDNgeERid3ROTTTrTTmFWWSs1b0FBTnlTZGdCMU9Ldi8yUT09PC94bXBH
SW1nOmItYWdlPgogICAgICAglCAglCAglCA8L3JkZjpsaT4KICAgICAglCAglCAgPC9yZGY6QWx
0PgogICAgICAglCA8L3htcDpUaHVtYm5haWxzPgogICAgICAglCA8cGRmeDpQREbiholwMDlwU
HJvZHVjZXi+QXZlcncgUHJvZHVjdHMgQ29ycC48L3BkZng6UERG4oaCMDAYMFByb2R1Y2VyP
gogICAgICAglCA8cGRmeDPHVFNfUERGWfZlcnNpb24+UERGL1gtMzoyMDAzPC9wZGZ4Okd
UU19QREZYvmVyc2lvbj4KICAgICAglCAgPHBkZng6R1RTX1BERlhDb25mb3JtYW5jZT5QREYv
WC0zOjlwMDM8L3BkZng6R1RTX1BERlhDb25mb3JtYW5jZT4KICAgICAglCAgPHBkZjpQcm9kd
WNlcj5odHRwOi8vYmZvLnNvbS9wcm9kdWN0cy9wZGY/dmVyc2lvbj1udWxsPC9wZGY6UHJvZ
HVjZXi+CIAglCAglCAglDXwZGY6VHJhcHBIZD5GYWxzZTwvcGRmOlRyYXBwZWQ+CIAglCAglC
AglDxkyzpmB3JtYXQ+YXBwbGljYXRpb24vcGRmPC9kyzpmB3JtYXQ+CIAglCAglCAglDxkyzpj
mVhdG9yPgogICAgICAglCAglCA8cmRmOINlcT4KICAgICAglCAglCAglCAgPHJkZjpsaT5BdmVy
eTwvcmlRmOmXPgogICAgICAglCAglCA8L3JkZjptZXE+CIAglCAglCAglDwwZGM6Y3JlYXRvcj4
KICAgICAglCAgPGRjOmRlc2NyaXB0aW9uPgogICAgICAglCAglCA8cmRmOkFsdD4KICAgICAgl
CAglCAglCAgPHJkZjpsaSB4bWw6bGFuZz0ieC1kZWZhdWx0lj5XZWlgUHJpbnRpbmc8L3JkZjps
aT4KICAgICAglCAglCAgPC9yZGY6QWx0PgogICAgICAglCA8L2RjOmRlc2NyaXB0aW9uPgogIC
AglCAglCA8ZGM6dGI0bGU+CIAglCAglCAglCAglDxyZGY6QWx0PgogICAgICAglCAglCAglCA8c

mRmOmxplHhtbDpsYW5nPSJ4LWRIZmF1bHQIPkF2ZXJ5IERlc2lnbiBhbmQgUHJpbmQgT25saW5lLCB2NiBEb2N1bWVudDwvcmlRmOmxpPggogICAgICAgICAgICA8L3JkZjpBbHQ+CiAgICAgICAgIDwvZGM6dGI0bGU+CiAgICAgICAgICAgIDx4bXBuUGc6TIBhZ2VzPjE8L3htcFRQZzpOUGFmZXM+CiAgICAgICAgICAgIDx4bXBuUGc6SGFzVmIzaWJsZVRyYW5zcGFyZW5jeT5GYWxzZTWveG1wVFBNOkhhc1Zpc2libGVUcmFuc3BhcnVuY3k+CiAgICAgICAgICAgIDx4bXBuUGc6SGFzVmIzaWJsZU92ZXJwcmludD5GYWxzZTWveG1wVFBNOkhhc1Zpc2libGVPdmVycHJpbmQ+CiAgICAgICAgICAgIDx4bXBuUGc6TWF4UGFnZVNpemUgcmlRmOnBhcnNIVHlwZT0iUmVzb3VyY2UiPgogICAgICAgICAgICA8c3REaW06dz44LjUwMDAwMDwvc3REaW06dz4KICAgICAgICAgICAgPHN0RGltOmg+MTEuMDAwMDAwPC9zdERpbTpoPgogICAgICAgICAgICAgICA8c3REaW06dW5pdD5JbmNoZXNM8L3N0RGltOnVuaXQ+CiAgICAgICAgICAgIDwveG1wVFBNOk1heFBhZ2VTaXplPgogICAgICAgICAgICA8eG1wVFBNOlBsYXRITmFtZXM+CiAgICAgICAgICAgICAgIDxyZGY6U2VxPgogICAgICAgICAgICAgICAgICA8cmRmOmxpPkN5YW48L3JkZjpsaT4KICAgICAgICAgICAgICAgICAgPHJkZjpsaT5NYWdlbnRhPC9yZGY6bGk+CiAgICAgICAgICAgICAgICAgIDxyZGY6bGk+WWVsbG93PC9yZGY6bGk+CiAgICAgICAgICAgICAgICAgIDxyZGY6bGk+QmxhY2s8L3JkZjpsaT4KICAgICAgICAgICAgICAgICAgPC9yZGY6U2VxPgogICAgICAgICAgICA8L3htcFRQZzpQbGF0ZU5hbWVzPgogICAgICAgICAgICA8eG1wVFBNOlN3YXRjaEdyb3Vwcz4KICAgICAgICAgICAgPHJkZjptZXE+CiAgICAgICAgICAgICAgICAgIDxyZGY6bGkgcmRmOnBhcnNIVHlwZT0iUmVzb3VyY2UiPgogICAgICAgICAgICAgICAgICAgICAgICA8eG1wRzpnncm91cE5hbWU+RGVmYXVs dCBTd2F0Y2ggR3JvdXA8L3htcEc6Z3JvdXBOYW1lPgogICAgICAgICAgICAgICAgICAgICAgICA8eG1wRzpnncm91cFR5cGU+MDwveG1wRzpnncm91cFR5cGU+CiAgICAgICAgICAgICAgICAgIDwvcmlRmOmxpPggogICAgICAgICAgICAgICA8L3JkZjpTZXE+CiAgICAgICAgICAgIDwveG1wVFBNOlN3YXRjaEdyb3Vwcz4KICAgICAgICAgICAgPHhtcE1NOlJlbmRpdGlvbksNsYXNzPnByb29mOnBkJzwveG1wTU06UmVuzGloaW9uQ2xhc3M+CiAgICAgICAgICAgIDx4bXBNTTpWZXJzaW9uSUQ+MTwveG1wTU06VmVyc2lvbkIEPgogICAgICAgICAgICA8eG1wTU06RG9jdW1lbnRJRD54bXAuZGllckJkZmZhYTdlLWYxMjAtNDdjYS1iYjM4LWE2ZDhkYmJiOWFiMTwveG1wTU06RG9jdW1lbnRJRD4KICAgICAgICAgICAgPHhtcE1NOKluc3RhbmNISUQ+dXVpZDoxMmJjMTM4OS1IN2UxLWU2NGltYmRiYS02ODQ3YjFINTIlOTAsL3htcE1NOKluc3RhbmNISUQ+CiAgICAgICAgICAgIDx4bXBNTTpPcmInaW5hbERvY3VtZW50SUQ+dXVpZDo4ZDM0MWQzMm1mYjY3LTU5NGltOTliZS04Mjg1MTFIODcxNDI8L3htcE1NOK9yaWdpbmFsRG9jdW1lbnRJRD4KICAgICAgICAgICAgPHhtcE1NOKRIcmI2ZWVGcm9tIHJkZjpwYXJzZVR5cGU9IlJlc291cmNilj4KICAgICAgICAgICAgPHN0UmVmOmduc3RhbmNISUQ+dXVpZDpmNTc3YTcwMC1iODkzLTUyZNGUtYjk4My1iY2EyNW00NTYwNTk8L3N0UmVmOmduc3RhbmNISUQ+CiAgICAgICAgICAgICAgIDxdZjJlZjpkb2N1bWVudEIEPnV1aWQ6OGQZNDFkMzltZml2Ny02OTRiLTk5YmUtODI4NTEwZTg3MTQyPC9zdFJlZjpkb2N1bWVudEIEPgogICAgICAgICAgICAgICA8c3RSZWY6b3JpZ2luYWxEb2N1bWVudEIEPnV1aWQ6OGQZNDFkMzltZml2Ny02OTRiLTk5YmUtODI4NTEwZTg3MTQyPC9zdFJlZjpvcmInaW5hbERvY3VtZW50SUQ+CiAgICAgICAgICAgICAgIDxdZjJlZjpwYXJzZVR5cGU9IlJlc291cmNilj4KICAgICAgICAgICAgPHN0RXZ0OmFjdGlubj5zYXZlZDwvc3RFdnQ6YWN0aW9uPgogICAgICAgICAgICAgICAgICAgICAgICA8c3RFdnQ6aW5zdGFuY2VJRd54bXAuaWlkOjAzZmZhYTdlLWYxMjAtNDdjYS1iYjM4LWE2ZDhkYmJiOWFiMTwvc3RFdnQ6aW5zdGFuY2VJRd4KICAgICAgICAgICAgICAgICAgICAgICAgICAgPHN0RXZ0OndoZW4+MjAxNy0wNy0xM1QyMzozMD01OS0wNzowMDwvc3RFdnQ6d2hlbj4KICAgICAgICAgICAgICAgICAgICAgICAgICAgPHN0R

[illegible]

2lkc1s2IDAgUI0vVHlwZS9QYWdlcz4+DWVuZG9iag02IDAgb2JqDTw8L0JsZWVhZm94WzAuMC
AwLjAgNjEyLjAgNzkyLjBdL0NvbnRlbnRzIDEyIDAgUi9NZWRpYUJveFswLjAgMC4wIDYxMi4wIDc
5Mi4wXS9QYXJlbnQgMyAwIFlvUmVzb3VyY2VzPDwvRXh0R1N0YXRIPDwvR1MwIDEzIDAgUj4
+L1Byb3BlcnRpZXM8PC9NQzAgMTQgMCBSPj4+Pi9UcmItQm94WzAuMCAwLjAgNjEyLjAgNzky
LjBdL1R5cGUvUGFnZT4+DWVuZG9iag0xMiAwLjAgMTQgMCBSPj4+Pi9UcmItQm94WzAuMCAwLjAgNjEyLjAgNzky
vTGvUz3RoIdlXOTY+PnN0cmVhbQ0KSIncl2O5MgRhP1+in6B660ii3/urQQZ0hnCGbIFQpCMXg
EnWXp7VWVG5l8jgwMQUGOAmYwddmYVMya+/fanv/7nb/96fvle3n+/Lvz8dvj6m8tue2rM8fj7q
+tqmJqPNr6cX02jaKtr/25/kl4q1iPeprMrFvr6k3Qa91b0+ZMlrzYwV/+lhe62rieK3zisen175aI0oZMjo
E+abkeSh5cLSVG+m8UZ6PIN6Pfzz+8vzno7z2/dmPt9b+fT3m/v1Yp+cfH9/+8Gt5/v3fj9+e/Tf9q8q
FSpn0Tj8e4x9/PH6aX8e29fKnfu7nLGeXcpNzS7k2OaXWuxwRH9um/sltPFi31gePur+WbX8ds4i6
T6+68INUTVeiatMtqNADsEE/tbYt/fFx3nHrXx9/7l/dAXU3C0y1mQemWoMJoOCCqN5QthNV3B86
ymplEqxg9fsxH7N7YT62vjT9VSSleoESXkjtCldUeL8bDtuhnnqhSCueGFc40YzbF81w/a/zdDGHY
7M0OpkZmjICGaAghmiekP5Uo5gBnaU3cokNYPX78fSP2lmWMYfKsywHHs0AyXMkOSbkoeixPn
ZdtwM89QMQVwwg1zjg8xgcW1sGAI5X5wN67Y6G1QQB/UIbKhLYkOdwlY6GRvqFNgAATbl45r+
aGTySGzATEqcx7hSEhvkRs6iM4PpOhvGnT6FDWqBnOQ1mAAKLojKaMCdQC2ZDWO1OkmtY
LWzAQJs6lrpj14mj8QGzKXEOYwrJbFh3Cyw6Mxg+glb7jHDPWxQM6QkR8aXwAaalSqjgS3ICGa
wjmO3OknM4LWzAWIxmZD90cvkkdiAuZQ4IHGIJDYocqqbIYHpC2z4DDNYXDsbuTcC3tgQ2mB
DaU5G5a2ORuWNkc2LK0oG0YBN0zS2EChla6PS76zkcktsOezCQOcx+Se2TBu5Cw6M5gus0Hu
9DFsEAvkJK/BBFQtsOgvXZTtBGpObJDVyiRYwWpjA4WG+FDId/YyuUU2cC5hgEOZ3DMb+s0Ci
84MputsuMkMN7FBzJCSXDMel5OqRTbYez+CGajmxAbZrUxSM3htbKBYzAzId/YyuUU2cC5hgE
OZ3DMbBDnVzZDAdJ0NH2IGi2tjwxCa8/PqbFjmydmgAjho2+JsaFuNbGjrrmwYBdgwSmMDhYa4
Pi75zkYml8gGziQqcB6Ta2KD3MhZdGYwXWad3OIT2KAWyElegwmopsAge+2ibCdQNbFBViuT
YAWrjQ0UGuJDId/Zy+QS2cC5RAUOZXJNbBg3Cyw6M5ius+EmM9zDBjVDSnLNeLxMqimwwd/7
EcXAVRMbZLcySc3gtbGBYjEzIN/Zy+QS2cC5RAUOZXJNbFDkVDdDaTn1NnyIGSyujQ1DaM73C
caGthZngwrioMzOhrnHcWDD3KfKhFGADaM0NIBoiOvjku9sRBjIEGcDZILiPCZbYoPcyFI0ZjBdZo
Pc6VPYOBbISV6DCahKYIO9dlXciSjbHzrKamUSrGC1sYFCQ3wo5Dt7EQbwQpIGAx7KZEtsGDcL
LDozmK6z4SYz3MMGNUNKcs14vEyqEtjg7/0IZIBIC0RH2a1MUjN4bWygWMwMyHf2lgxghiQNBjy
UyZbYoMipboYEputs+BAzWFwbG4ZAzK/OhnnfnQ0qilO5BjZMW2KDpPuY0AuyoZfOBgiwwWHA
RpQyxNjAmZQ4j3FISmyQGzmLzgyM62wYd/oUNqgFcpLXYAlouCAqowF3oor7Y0epd7eC1c4GC
LDBYcBelPBCKgYDHsq4MiU2jJsFFp0ZTF9gwz1mulcNaooU5Mj4KbCBZojKaGBLOYIZrKPUq5n
Ba2cDxGJmlBvQixJmSNJgwEMZV6bEBkVOdTMkMH2BDZ9hBotrZ8P4qOZ8CWzov3U2iAAOpU
VwNkzLEtkwLZOyYRRgwyiNDRQa4vq45DsbnTwGziTMMB5jCsIs2EcyFI0ZjBdZoPc6WPYIBbIS
V6DCaDggqhlA9sJ1JLYIKuVSbCC1cYGCg3xoZDv7GXyiGzgXMIahzKulMwGCbvZvZDAdJ0NN5
nhJjalGVKSI+NLZAPMEBVp4Es5ghnYUXYrk9QMXhsbKBYzA/KdvUwekQ2cSxjgUMaVktkgfaql
YHpOhs+xAwW18aGITTn2+5smFpzNqgADuq+ORvqPkc21L0oG0YBN0zS2EChla6PS76zkcktsO
EziQqcx+Se2CA3chadGUyX2SB3+hQ2qAvyktgdAqoW2GCvXZTtBGpObJDVyiRYwWpjA4WG+F
DId/YyuUU2cC5RgUOZ3BMbxs0Ci84MputsuMkM97BBzZCSXDMel5OqBTb4ez+CGajmxAbZrU
xSM3htbKBYzAzId/YyuUU2cC5RgUOZ3BMbFDnVzZDAdJ0NH2IGi2tjwxCa89vqbKjb5GxQQRzU
JbChB0NkQ9nBhrlbG8oe2AABNsijmv5oZHJJbMBMSpzH5JrYIDdyFp0ZTNfZMO70KWxQC+Qkr
8EEVFNgg712VdwJVM1sGKvVSWoFq50NEGBDV0x/9DK5JDZgLiUOZXJNbBg3Cyw6M5i+wIz7
zHAPG9QMKck14/EyqabABn/vRzADVc1sGLvVSWIGr50NEluZgemPXiaXxAbMpcShTK6JDYqc6
mZIYPoCGz7DDBbXzoYuNOdLC2wYf2rGBhHAQZsdDfMRyTD+yz7a95/gQq8MC6g1vuVRCXa0

gJLmhgSMlgJwCpMtE2Hcwwl0ZhxdJsK4zMcAQfae47uGzVOVCAS+dVHciAiuDv3GTmWM7p+ls
QC1Zva8MvrRBQoGiMpzH0cx2TIG+n0Cds7MoOsYuMcBN1FAHJAYW9Mc75KqRArYaz/cASq4O
/QbS5Ux4gArDQCoFzoAEY4uUHBAVB72OlrJlrNfkFLdAQk817P//98Bv//l+7P/+K8AAwBJRrVDD
WVvUZHNOcmVhbQ1lbnRvYmoNMTQgMBCvYmoNPDwvQ29sb3JbMjAyMjQgMzI3NjggNjU1MzV
dLORpbW1lZCBmYWxzZS9FZGI0YWJsZSB0cnVIL1ByZXZpZXcgcGdHJ1ZS9QcmlludGVkIHRYdWU
vVGI0bGUoTGF5ZXIgcG9yYmlzaWJsZSB0cnVlPj4NZW5kb2JqDTEzIDAgb2JqDTw8L0FJUyBm
YWxzZS9CTS9Ob3JtYWwvQ0EgMS4wL09QIGZhbHNIL09QTSAXL1NBIHRYdWUvU01hc2svTm9
uZS9UeXBIL0V4dEdTdGF0ZS9jYSAxLjAvb3AgZmFsc2U+Pg1lbnRvYmoNMTEgMBCvYmoNPD
wvTGVuZ3R0IDY1NDQ1Ni9OIDQ+PnN0cmVhbQ0KAAAn8eEFEQkUCEAAAcHJ0ckNNWUtMYWI
gB9kABgAaAAAFgATYWNzcEFQUEwAAAAAQUURCRQAAAAAAAAAAAAAAAAAAAAAAAAAPbW
AAEAAAAA0y1BREJFAAA
AAAAAAAAAAAAAAAAAOZGVzYwAAASwAAACAY3BydAAAAawAAACFd3RwdAAAAjQAAAAUdGFy
ZwAAAKgAAAAAddGVjaAAAAmgAAAAmdnVIZAAAAAnQAAABedmllldwAAAtQAAAAkQTJCMAAAA
vgAAV9mQTJCMgAAAv9mQTJCMQABYmAAAV9mQjJBMAACwgcAAji0QjJBMQAE+nwAA
ji0QjJBMgAHMzAAAji0Z2FtdAAJa+QAAJCRZGVzYwAAAAAAAAAAmQ29hdGVkIEdSQUNvTCAY
MDA2lChJU08gMTI2NDctMjoyMDA0KQAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AA
AB0ZXh0AAAAAENvcHlyaWdodCAyMDA3LTlwMDkgQWRvYmUgU3lzdGVtcyBJbmNvcnBvcnF0
ZWQuIEZvciBsaWNlbnNpbmcgdGVybXMgYW5kIGNvbmlldwAAAtQAAAvsV0ZXh0A
AAAAEIDQ0hEQVQgQ0dBFVMgVFllgMDA2AAAAAHNPZyAAAAAAAb2Zmc2Rlc2MAAAAAAAAAAB
EQ1MAAA
AAAHZpZXcAAAAAAmXUKwJ
8nqkCDSXMAHrECQB/Uu8AaQePAAAAAW1mdDIAAAAAABAMLAAABAAAAAAAAAAAAAAAAAAAA
AAAAQAAAAAAAAAAAAAAAAAAAAAAEAAAEAAAIAAAELAhYDIwQxBUAGUAdhCHEJgAqOC5w
Mqg25DskP2RDpEfktChQaFSOWOhdJGFYZYRpsG3ccgh2NHpcfoSCqlbMiuyPCJMGlcCbOJ9Ao
0SnQKs4rzCzJLcYuxC/CMMAXvjK6M7c0tjW3Nrs3wDjGOc462DviPO09+T8FQBBBGkIIQy9EOE
VCRktHU0hcSWRKbUt1TH1NhE6CT4FQf1F+UnxTe1R6VXIWeFd4WHhZd1p3W3dcdl1zXnBfbW
BqYWhiZWNjZGFIX2ZeZ1xoW2laallrWGxTbUpuQm86cDjXKnJcxt0FXUOdgh3A3f+ePI59Hrwe+1
86n3ofuZ/5IDkgeOC5lPlhOaF6lbrh+6l8on3ivyMAY0Hjg6PCJACKPyR95Lzk/CU7ZXqlumX6Jjome
ma65vtnPCd9Z76oAChB6lOoxekIKUqpjWnQahOqVyqa6t7rlutna6vr8Ow17HsswK0GbUxtkC3Q7
hHuUy6UrtZvGG9ar50v3/Ai8GZwqfDt8TlxdvG7sgByRXKKss/zE/NWs5l2Z/QeNGB0onTj9SV1ZrW
ntei2KTZptqn26jcp92m3qXfo+Ch4Z7imuOW5JHleeZd50LoJukK6e7q0eu07Jfteu5c7z7wIPEB8eL
yw/Oj9IP1YvZB9yD3//jf+b/6oPuB/GX9Sf4v/xb//wAAAO4B3ALLA7oEpwWVBoEHcAhfCVAkQws2
DCKNHA4QDwQP+RDvEeYS3hPXFNEVyx/F7QYqRmfGpUbixyBHXgebx9mIF0hUyJlIzWkMSUI
JhknDigCKPg7irlK90s1i3MLr8vtDCsMaYyojOhNKE1pDaoN604szm7Or87wzzHPcw+0D/UQNdB
20LfQ+JE5kXqRu5H7EjnSeNK4EvcTNpN2E7WT9VQ1FHUUtVT1ITXVddW11fWWNdZ2FrZW9tc
3V3gXuRf52DsYfBi9mP7ZP1l+2b6Z/lo+Gn5avpr+2z9bgBvA3AHcQTYEXMXdB51JXYudz4QXIM
elh7ZXxzYfYJ+kn+jgLWBylLdg/KFCIYghzGIO4lGilOLYlxujX6Ojo+fkLKRxZLZk++VBpYdlzaYUJlrm
oSbnJy1nc6e6aAFoSkiQKNfpH+loabDp+apCqovq1WsfK2krs2v8LEHsh6zNRrPtWm2g7eeuLq51
7r0vBO9Mr5Tv3XAmMG9wuPEC8U0xl/HiMiqyc3K8MwTzTbOWM960JrRudLW0/HVCdYg1zXYR
9lY2mfbC9x/3YjekN+W4Jvhnuka43rkWuU45hXm8OfL6KTpfepU6yrr/+zT7afuee9K8Brw6vG48obz

U/Qg9Ov1tvaB90r4FPjd+ab6cPs5/AP8zf2Y/mT/Mf//AAAA6QHTAr0DqASSBXsGZQdOCDkJJAoP
CvsL5wzTDb8Org+eEJARhhJ+E3gUcxVvFmsXaBhkGV4aWRtSHEodQh46HzEgJyEdlhMjCSP/J
PUI6ybhJ9cozSnDKrgrrSyhLZQuhy95MGkxWDJHMzQ0ITUNNfk25TfROL05qDqUO4E8bj1bPko/
OUApQRhCCEL4Q+hE2UXLRr1HsEijSZdKi0t/THRNaU5eT1JQRIE7UjFTJ1QdVRRWC1cDV/xY9
FnuWuhb4lzeXdpe11/UYNJh0WLQY9Bk0GXQZtFn02jVadhq22vfbONT6G7tb/Nw+XH/cwZ0DXUU
dhx3JHgteTV6PntHfFF9WH5ff2aAbIFzgnuDgoSKhZGGmYehiKqJsoq6i8OMzl3WjuCP6ZDvkfaS/
ZQFIQ6WF5chmCyZOJpFm1OcYZ1xnoGfkqCkobeiy6PgpPamCKcaqC6pQqpYq26sha2errev0rD
usgqzKLRHtWi2ireuuNS5+7slvFC9fr6qv9jBCMI6w27EosXYxw7IRMI5yq7L4s0TzkPPb9CZ0cDS5
dQH1SjWRtdi2HzZldqs273cu9233rLfreCm4Z/iluON5ILld+Zq513oT+IA6jDrH+wN7Prt5u7S773wp/
GQ8nnzYfRJ9S/2Fvb79+H4xvms+pH7d/xd/UP+K/8U//8AAADhAcICpQOIBGwFUQY1BxkH+wjdC
b8KoQuDDGcNSw4vDxQP+BDcEb8SohOFFGgVShYsFw8X8RjUGbYamRt8HF8dQh4lHwgf6yD
OlbEilCN3JFolPSYgJwMn5SjlKakqiysLE0tLS4MLusvyTCnMYQyYjM/NBw0+TXWNrl3jzhsOUk6J
zsEO+l8vz2dPns/WEA2QRRB8ULPQ6xEikVnRkRHIUf+SNtJuEqVS3NMUE0tTgtO6U/HUKZRhF
JjU0NUI1UDVeRWxVenWlpZbVpRWzZcG10BXehez1+2YJ9hh2JwY1pkRGUuZhlNBGfvaNtpxmq
ya55siW11bmFvTXA4cSRyD3L6c+V00HW7dqZ3kXh8eWh6VHtAfCx9Gn4HfvZ/5YDVGcaCulOrh
KCFIYaMh4Wlf4l6iniLd4x3jXmOfI+BkleRj5KXk6GUrJW4lsWX1JjimfKbA5wUnSWeN59KoFyhb6K
Co5WkqKW6ps2n36jyqgSrF6wprTuuTa9fsHGxg7KV s6e0urXOtUO3+bkQuiq7RbxjvYO+p7/OwPjC
JcNVxlfFvMb0yC7Jasqny+XNJM5jz6LQ4dlg01/Un9Xf1x/YX9mg2uLcJN1n3qvf7+E14nzjxOUN5ljn
pOjy6kPrlezq7kHvnPD48lfztvUW9nb31fkx+or74v08/pr///8AgACAAOluf8x/2NPPf5x/tb4df4x/s6hUf
3x/s5Jof2p/uHxEf2J/zWXDf2l/8U6Xf3+AGTX7f8eAQRk7gFSAqvyzfwqJ5Oc+ft+l19G+frWHz7w1fq
eG+aaafp6GLJDkfp2FbXr2fqOExmSsfrKEOU24fsuDszVUfw2DQBjgf12DI/qlfhGTzuVKfe2R28/mf
cyP7bqAfcuOQaUPfdGMpY+lfd+LH3nMffKJumOzfggleUzxfieHRDS/fmSGOhiQfnyFc/jYfTedw+O
OfRSa6M48fPGYFLjzfPOVlqOlP+TL45LfRiQ6Hi8fTuOvmLUfWmMuUxDfZWKyTQ8fcyJLxhJfbG
HdfckfHWnweHwfFekC8y0fDigWbeGfDudB6JZfEuZ040mfG2WxHfCfJ2T02lGfNmRCUujfQ6OYD
PLfT6MOhgffO+JTvWre9uxzuB5e8etO8tCe6+or7Yue7OkjKEfe8WgiowPe+ucsHbSfB+Y+mFBfF+
VdEsNfJOSGzNvfLaPcRgnfC6LFvQ7e3O76t8ee1e2jsn8ezixM7T+ez2sOaAJe1GnYYsZe3qis3X+
e7GeMmCSe/KZ60qJfCSV4DMffD6Suhgxe4KMpPLSey3GGd3GewW/8ci1etu5x7PPet2z+571eu+
uUYolexio0XUre0+jhl/me5GegEoGe8GZxDLPe9KWFBg7eu6N+vF/evnQYtxteszJWMdaepvCUrK
Nepe7w53QeqS1VokjesevFHROevqpCV8wezqjSUI8e2ad4DJ7e3KZFxbCenCPGvAaetLaztsYeq
DTB8YXemnLQRfJel/D7pzCemW8uYg0eoG1rXOBerCu216leu2oW0j9exWiQjluex2brhhJegSQD+
6rerXlc9nben3dLcUHej7U2LBoei/MrZveei/Em4dreka8sHLVenG1BF37eqyttkiSetCmmTHtetad1xh
OeaqQb/Deh0h+btX0hnd+csgDhad+fLOBhPl+pZ7nhFd+y4ong8Z+73Ukgz5/J1+4gsh/c0mcgmt/w
DHygmGABBYxgueAne7zhkOH/9qmhXyHJsZQhLWGU7HkhA2Frp1qg3aFDojYgvOEdHQDgnmD
+F7Dgg6DmEjUgbaDQDFagaWC8xXYge2DAO0jhVSRftjvhJSPycSzg9OOHLBogzSMr5wTgqeLS
4ergjKJ83L/gcWlWf3ogWSHtEgJgROGtjDSgPuF3hWlgQqFGOt7hlGa/ddXg8KYbcMrgwKV5q7/g
miTtZrNgeGRkoaLgXOPgXIjgRSNIF0fgMaLzUeHgICKHzBbgGOlwxVDgD6G6+nbG9CkhNXPgx
Khl8G9gIKdy62ugbqa05mbgTeX7oV9gM+VIHEigHqSdlxkgDiP9Ub3f/mNmS/0f9OLwRUbf3Wlmu
hog0muDdRggo+p48BWgdKlwqxigTqiBphugLeeXoR0gFGa0HA/f/6Xa1urf72UOkZsf3qROy+hf0W
O8BUnfqOKPecCguK3udMNgiSyzb8XgWKt5as9gMmpXZdkgEek64Olf+KglG92f5GccFsHf1CYi0
XyfwqU5y9YfseSNBU2feqLreWegpjBgdG7gdK7y73XgQi2GqoWgG2wypZZf+irklKdf4KmdG6tfy+h
klpifu6c+kV4fqSYtS8PflaVUhVDfUqM6ORGgmTLWtBigZjEz7yBgMe+TKjagCW4Q5U8f5uyUYGif
y6sfW3Xftim51m1fpaho0T1fkecwS7AffKYCxVOfMSN8uLmgjrVWc8SgWrOG7tDgJPG4qe1f+rAH

ZQzf1m5boC6fuWy3W0QfousjFkTfkeml0R8ffShFy53fZqaYxVXfFOObuGLghTfks3ggUHX2bo3gG
fQGKa+f7jIhpNUfyDBBn/2fqe5o2xpfkmyiFiLfgOr10QWfa6IDC46fU+cVxVfe/SOc+L/jrV9Es+6jUN9
P7xwi9R9cKkPiph9rJWaiWt96oIEiFJ+LG4hh0R+hVnlhkp+8kS4hXV/Yi35hRV/xBM0hYqAh+FdjZ2
GTc40jDiFpbsDitWFAKe0iaSEd5RailSD94Drh3iDg20shniDLVj0hYuC90QJhL6CyS1whFeCpBLfhl
iCtt/RjK+Pb8y+i1CN9LmkifOMfaZyiMqLNM4h7OJ+H/shrKlzmXOhb6Hy1g3hN6G8ENthBiGJSz3
g6uFgRKVg5uEoN5Wi+2Yg8tUio6WOLhMiTCT86U0iAmR75IWhvWP/X7rhfqOJGtzhRGMb1eGh
ECK5ELkg4OJcyyMgxGIWRJUgseGS9zji0uhoMn1ieqejbCilubf6QCh2aYvpEAhISWFH31hVuTiG
qihHmRIFbeg7OO5UJkgvqM0ywwgn6LTBlxgfmH1tuOis+qvMigiXCm6rWxiBGjHaLJhuufoY/khdm
cPnz7hOCY/GnNg/6V5VYygzeTBkHlgniQXSvlgemOdRJCgSCJWdo7imuz98dgiQqvblSEh6iq46
GzhoGmpY7nhW6ignwchHSegmkPg5Kat1WagsmXMEF2ggST8iukgWSRnxJVgGCKrNjliiC9TsYb
iLm4A7NRh1GyuaCXhietvo3nhRGo33s6hBSkJmhQgzCfqVUAgmabfEEEgZuXqytigO6UcBJlf7yL
z9eaiezGt8TMiH/AorlChw+6kp9jhd+06YzQhMKvXHpHg7+p9meBgtek0IRYggugBUCIgTybqSsYgl
WW5xJyfzKMedZCibzQR8ODiEzJiLDJhtjCzp5DhaK8c4vNhH62NXlkg3SwHmbBgomqTFO9gbqk
3kAUgOef1yrTgCmZBHj+fryMgNTRIYvaE8JUiB3S3q/AhqrLoZ1QhW7EhoryhEW9hHilgzW2qmYfg
kWwHVM6gXWqBj+zgJ+jayqZf9uayxKHflqMh9VYlKx8JMM2IDF8ebEQkxh8zp7HkE59Goxujpd9a
Hn3jPx9umcsi2x+GVPjifN+hz/eiKh+/SoGh+5/fhBxh+qAbtP+ITqE9MH0kyyEfK/ikSWEBZ2mj2OD
mlthjbeDMnkJjCaC12ZciqCCk1MwiTGCaD9lh+qCTCmNhyuCURAlhtmCb9KsIFWNqMC1kk2MZ6
64kEuLJ5yWjpGKBYptjO2I7ngxi2SH5mWgieiG/1KPiIGGPD7Dh0CFjykhnuFlg/iheaEMNFZk5qW
Vb93kZKUTa2Pj4+SSpuCjdeQdolxDWOs3dTiq+NBWTjiTuLdlH4h+KKCz5OhqilxCjFhd6H7g+oh
RKft9AWkwGfBL5FkPacPqxtjvGZfZp1jTqW94h8i5mUhXZ6ihOSL2QsiKSP91Flh1KN6D3ehhuM
Cih0hUaK2A+LhESHI87kkoOnrr0QkH2kN6s6jnqgwplYjMKdi4d5iyCaa3WXiZmXaWNriCmUjVDJh
tSR5D1shZSPfSgyhKqN/Q+fg2aliM2wkh6wd7vtkBesUKonjhOokphajFqkO4aTiregZnTMiS6cs2K+
h7yZLIA/hmWV6z0HhRyS/Sf6hB6Q4w+0gqGJwcx0kdO5XLq+j8m0f6kHjb+voZdQjAOrAIWjil2men
P6iNCiGWIMh1ud70+whgCaEzyfhLCWpCe9g6KTbg/HgfmKq8s4kaDCVrl8j5G8uafEjYG3lZYmi7+
x24SWihGss3MPiH2nsGFGhwKi608RhaSefzwphE6alyd5gzSVqQ/WgWuKtcnskXbLebg8j2PFPK
aSJU+/ApUNi4a5FoOZidGzR3lziDStoGCMhrSoO057hVKjPTu7g/aeeCc4gtOXIQ/kgPKKvsiekVDU
27cWjzvOK6WQjSTHdJQgi1bA0oLDiZq6TXF5h/az8V/whnGt4k39hQ2oRDtfg6yhtScCgoGZMA/vg
l6KxsgFng57hbbwm2F73aXamLh8NpSmlk58koNmlAJ88HIMkdp9UmBXj8B9t04ajcB+IzsZi/p+nyY
fiul/Qg3wigSAZ8b1nQaD5bXymmSDfKTsl8aDF5PAIWiCxYKNkyaCenFKkQaCN1+tjvKCBE2ljPm
B4zqeizWB2SW5ih+CDA2viO+CP8XVnCaMKLTomYmLAqP3lvCJ4JLjJqI5lHHkl+H8nCakEWHD
V8SjjiGQU0EjEWFIDoxioSFBiVeiWmE0w12h/2D28S3m26Ua7PcmNGSicL/ljiQsJH7k+SPDIDzka
qNeG/fj5CL9l51jYmKi0yHi6GJQTnRieallyUOiMeHlw1EhyqFQ8OgmtWcqbLVmDeaGKIGlZyXjZE
Uk0mVQYAgkQ+TCG8ljvSQ513YjPCO3kwLiw6M+TlziVOLUSTKiCmKfA0thlqGkcKVmlik5LHEI76
hraDzlSWee5AVktKbiH85kJeYq25ajnmV6V0sjHOTRkuAioyQ1DkPiMeOsCSTh4ONdw1GhXaH2c
GBmfOtPrC+l1ipYJ/9IL6lhY8xkmqh5n5rkC6eX22ljg6a9lySjASXt0sFihmUtzi1iEqSGyRkhu+QEw1g
hK2l+cBdmaS1tK+mlwixKJ7ylGysn446khWoV32Mj9WkKmjzjBcGlFvui6KcSEqCibKYvjhWh9mVs
SQwhmySXg11hACJGr8zmWm+Oa5zls+4/J27IDGzxo0bkDWu5HyKj4yqH2wCjV6lgfsyi0qhGUnri
VSdDDfoh3OZiyPxfhiUYQ2lg26JJr33mTzG6q1AlqLBGZySIAO7TowKkaC10HuUj0+wb2stjRirOFp
/ivymQEiciQGhsjeAhxmdDCO1hZKWHg2XgvOJMby8mRvP4awmln/JopuXk9zDX4sjkXK9N3rFjxu
3Lmp6jNuxUFnoirqrkjkilumDdcphsyf9yODhTyXkQ2lgoyJObstpgV7RKsropR7upsnny98Llr1nDF
8jHq3mVV88GpilqN9WlmlAl9yEhkkZp+QDZOj3d+zilXjj1/nAtqjGGA97p8pPWDSap1oZeDBppsnk
OCwoo+m0+CfnoLmHuCQmnJlc+CEVklkzmB70fykM2B4jXwjqeB9iHPjWOCZgtMiz6CormfpBSL

MamioL6KOZmmnXKJQomNmoWlW3ltl7eHgGk8lQ+Gtliskn6GBEElkBSFdTWeje2FECGPjJ+FM
Qsxij+EGbito1mTBKjBoAeRYJjVnLuPvojJmdCOO3i6lwOMx2iglFmLaVgqkc2KI0cnj2yJATVVjUiIG
yFXi/CH+gsaiWKFX7fFosGa0Kfjn22YiZgBnCCWRlgEmTaUl3gHlmiSFmgDk7yQl1enkTGOSEbCj
tSMmDUNjK+LNIeoi0aKxQsailiGirbnokaimab6nvWfuZcSm6mc2ocmmL6aHHc+le2XdGdTkz+U6l
cSkLCSgEZNjk6QTTS8jByOfSEGipONWwtGh5mHn7X6od2qhQYYno2nBpY8m0GjhoZgmFWgJn
aLIYOc3ma2ktGZuVaOkD+Wv0XkjdeUCzRzi5iR0SDpifKPnAtwhsWHu7TyoyeykKUdnjmuaJVNm
uyqQIWEI/2mQnXEISeiYGYJknCeolX8j9qbHEVxjWyX7DQjyKVUiDFiWSRIwuThhCH1LPfoUO6
o6QBnfe11ZQvmqyx4R7l7asfHTWINmoDGU6khqjxVVPj32fvETojQmcFjPBiraYzyCUiOaTVAuyh
XeH6LK5oQzC5KLkncO9iZMbmne4MYN+l3uzEHPylJauEGR1kc6pPVSqjyqkr0RljLGgmjNkilWb2
CBjiHiU1AvLhPWH+bGVon/LbKHenZfFoplymkq/1lKol0m6FnMyIF20emPOkY6vD1QfjuWp+EP3j
Gek0zMwigSeYSA7iBuVyvwvhlmICK6vriF7PJ+nqhJ7rZCfpg98HlFuonB8gHlynvd862Lgm6x9X1
MmmlD92ELQIZ1+XDGHkxV/AB4hkbR//gkwjneBda4+rRKC458sqRaCp5AbpSKCa4DqoYuCNXG
1nhmCCWJzmtKB6ILHI6+B20KBIMWB5DFikjeCGB35kMSC0AkWjUiC+q2crDGKbp6PqDyJh4+Hp
E+lo4BmoL+H03FBnVKHEGINmg6GYFJwlu6Fy0l5IASFWjEskXCFIx3Wj+6FogkxjD+ET6zbq3eR
1p3bp4OQUI7fo5WOzn/KoAeNa3CznJmMGmGRmVOK4VILlJaJwEHwk1SlYTD8kMGIHh23jzCIW
wkxi1eFeKwfqtuZNP0npuaXGY4zovaU/38pn2mTCnAgm/mRKWERmLKPZVGilZWNukGikrWMPz
DKk6LJx2fjnaK0AlFinSGRKtsqlugkZxmpmid541ponqbPn5unuuYtW93m3qWRGB9mC+T8lEnlQ
6Rw0FDkiiP0jCNj4GOWh2UjbGNDAmBiXuGbaqje6oEJunpf2kz4y0og6hjn3Gnn6ebW7gmwubZ
1/5l7yYhFC5lJiV0EDukauTazBWjvWRmh2KjQKO/wm6iJ+Gk6m8qZOvrJrLpaKryovjobOn6n0FniG
kNW4xmQmgN19il1adMVA6lC2Z/UCLkTqXKDAVjneU6R13jGaQtQnqh+KGtKjCqUi3T5nJpVqy0or
doWquW3wUndKqH21YmlSmBV6mlvmiF0+dk8qebEAQkNCbMS/AjgOX7h1Ti92SNQoSh0OGz6
e0qQm/HpjBpRy6HYnfoS21InsqnY6wXmyHmgqrV13wlqinUU8Fk3KjLz+ZkHOfdC9ujZyakB0vi2W
TgQo1hruG56akqNTHMZfPpOnBxlkloPm8WHplnVW3BGvWmcux1l1Xlmos306FkyioQD81kCOjH
S8pjUScwR0QiwCTpgpShkuG+qKltmN7dJRjsdB7woZErUF8FXgPqPp8fGnRpNI87VuAoOx9a0y+
nTN97j1Vmcp+gC0Hlth/PBo0lVSAcgc5kE6B5al7tV+CuZQYsNSCaYX6rE2CIHfBqAyB9mmHo/KB
2VtAoAiBzUylnFCB0T0rmOOB8SsrleaCSRowlEiDUQdVjXSDSKHntIGJ4pOwr/il+IWAq3KlFHdup
ziHV2k+oyOGrFsCnzyGF0xVm4SFnj0EmBSFTizSIQ+FSBosk1qGFQdtjgGEf6FEs8WQ55MtrzmP
Z4UdqrCN8Hb6pneMrGjWomGLfVqonniKZ0wPmsOJbTzXl1mloiy8lFGlNhookoelfgeBjRKFE6Czs
yKX35KirpSV04SZqgmTz3Z+pdGSA2hmobqQTlpHnc6OuEvCmhqNPTyilrCL+yyik6GLMhoqkbqK
pgenjCaFLaAlspie0Jllrg6cQoP2qYSZunXnpUuXZGfdoTKVJ1nRnUOTDkthmYqRGjxalhmPbSx6k
viOUxo2kOCMIgfyiySFYJ+CsiAl4pFtrZuixONkqRCfsHVgpNecy2djoLuaBFImnMmXZUsJmQyU+D
wZlZSS5CxWkmKReBpBkB+ORQg3ikCFjp69scatDZCxrTqpWoKxqK2lr3S6pHGiqmbOoFke9Vjln
Fub1kqemJqY9TvlIRuWgCwlkdqUWxo/j3SPvwhyiX2Ftp3hsXW0Ql/LrOqv/oHFqFqryHPhpBmn3W
YLn/SkF1g8m/eggkoRmDCdNjtblKqaavdkV2W9Roqjt2RCgikiNiF2JzqsTC7nY7arkW24YDdqBOy
MXMMo8ytyGVLn6KpiFeWm5+lfoMHI9KhyjrwIEaeNCuWkO6ZOhoSjlqRoQjOiEyF9JvssPfDOo37
rGu+GIAZp9e4/XJY04y0EGSqN1yvTlclm1Sqy0kTI4KmeDqWk/ChXCtakJGbhRn+jeyRkwjyh9eG
DJaLvsN8B4lLuXV8PHwStDN8eG6+r2F81GFlqrZ9P1P9pkB9vEYlogp+RTelnjJ+5Cg1muZ/thYYm
UKBlwV3kfWCSZZrvcGC4ok0uH2CjnwAs0OCQW6lrmOCFWFMqcuB+IPrpVeB9kYboR2CCjeknT
qCQyhBmduCwBY+mBKD7QWqkLGDjpY1vOKJpokAt56lx3vQsmaH7m5/rZ2HOGEvqPmGmVP
WpIWGF0YOoEiFtzeHnF6FjShLmO+FvBZelwWGxwXXj5WD85XkvCCQS4irttiO43t5sZuNg24vrN
OMUGDoqC+LNVOco7mKPEXnn32JZjeTm5WlZShVmB+lqRZ6lhlGAX/jp6EDpV/u3eW4lhHti6U+
HsYsO2TF23WrCWRZWczp4CPz1NZoweOXkW3nsqNEzd6muCMEChVI2CLoBaZITOKYgY0jay

EMpUVuuWda4fStaGbCnqbsGOYsG1iq5qWfWAxpvKUaVMAonWSf0VwnjOQxzdKmj+PZShFlqi
OrhbAIECMCwaNjKKEbpSTummkDodZtSihLHoqr+yeTmz6qyCbl1/TpnaZA1KwofWWn0Uvna6Ue
TcembKSvyg1lgaRaRbik2iNfAbdi7eEpJPvugOqyoa8tMGnXHmWr4Oj9mxxqrWgv19YpgadsFJGo
YKa1kTZnTaYSDbfmTGWPSgVIXOT1RbzkquOvQciiu2E0pMxua2xklX0tGutnnjHryqptmuzqlamC
16upaGijFGyoRifR0RenMWcWzaCmLmZ1SfblO2WABbvkgPaPgQdcikOE+pJWuWO4g4UftCC0H
Xf5rtyvwWr3qgKrol4FpUins1EfoLmkBkPknGGgvTYkmE2c/CeflHaX4hbkkXaPegeNibOFG5FvuS
O/t4RWs+C66ndMrpq2JmpYqbuxjF12pP2tKFCjoGmpEEN9nA2kvTXWl/OfnSdtlBKZcxbbkP2PdA
e2iTufN4rYx0h80X5xwU1853lQu2d9BmWRtgx9UVkSsNh9sEyJq9h+Kz+VpyN+uTH5ott/ZyNinz
OAUhlLnUqBxwPsk2eColr2xkaDSX6XwFWC6XI6unSCkGWxtRiCW1kur+SCPkyqquSCQD+6pia
CYzlmoCqCuSOBngaDYhJbm/WEpWQ1khuC2lr/xV6Jq36bv26lznI8uY+H+GW7tDqHRFk/rwqGrE
y/qgmGOT/WpUOF9TJLoNmF9SPNnP2GZBKgmseGagR0kPeDA4rmxl+P9H5zvpYom3lJuLiNS
mWPs2SMllkbrjOLGUynqS+KOD/RpGmJiDjen/2JJyP5nBaJWBLdmb+ITASsj/qDKYquw92WKX4
4veeUXXHMuACSI2VYsqyQ+litrXmPfkYEqHKOMD+/o6mNFzJjnzmMWiQXm0KMQhMYmMCJ8w
TwjwGDVoprW0KcUn3qvVKaF3F2t26X4WUJsheVy1inrOCT20xKp9aSHz+UoweQpDJNnoqPlyQg
mniO4xNal7OLXwVUjfcDmooMwr6ijH2UvNKf2XEotvCdK2TBsZWam1hnrFyYNUwVp06WCj9son
mULjl4nfGS2CQmmcaRKxOUIsSMmgWujP+D14mlwk6o230XvGOlqHC0toCifGRXsSKfdlgIq+Scn
0vDptOaCT8rofiX0zIMnWaWICQXmSaTLRO5lfONVgX7jDCEC4jnwfkVnN3xtvAWriXAEth2n5WO2
sLkdFd4q3ahNktGpmCeQT7EoYCbUDG/nOWZGCPsmJWU9BPFITyNXgY8i4KEN4gowaC1wnu
0u7Gxpm9TtctelGMUsFypt1bnqxSmFErHpfqiWj5aoRSfozFvnHkbqyO9mBWWfBPJlJ2NYQZ0iu6
EXYdbwVa8lHsEu2m4Dm69tX6zk2KMsA2vRFZtqsGrOUpcpaOnVD4BoLmi7DEsnBGd0SOVI6iX
wxPMlBiNYwaiinKEfH7uz9t9+XM5ycJ9yWeQw6l9q1vxvbF94VBYt9x+NES9sjd+qzi5rOl/QSwNqB
mAAx5WpAWBEw5GoPWCVQKBILuBsX+BzuOEMnO7yMSDjWgDwqSC91xVvKiCtIC0ts+CIUUZ
sSmCnTkXq86C1CxwpuKDTR7FoqeEMg67n4OEggLdk2eB8H/czfCKNnQFx82JH2hAwaqIGFyYu
7SHZFD7teCG00VhsDmGcTljqtkGTcZepdGGgR8moXWHRA8gnj2GaQMukjyCJn/3zRGQlnQNx
ueOnGg4wL2NJ1yWusiMCVEBtPOLEUVxr0iKSzmFqd+JyCz/pNiJqx96oGuKHA95nSCIEQN0KT
mCVn/pzFCV8HP7xiGUBmgkv/GSLVylufyQoId5tCWPQkVyrneOGjmWqQqNOy0mo/yM0h+6n3q
Mmw/NnA6JgQPfkdQcJX/jy7ibwXPixYSZa2f8v06XJFxiuVWVK1Das3uTYUVbrcqR1jmLqFaQoS0
tozqP+h/gnpeOyBAImu6KsgQ0jyOC13+vyzKhmXO1xPuez2fWvsCcFVxAuMSZrFC8sueXd0VFrT
SVijl/p7qUAY0yopCTBB//ndCQqhBymfCLHgSXjiyDGn9Hyrune3NaxlGkOmeHvkKhClv4uEGeNIB
8smCbm0UPrKuZUTIXpyyXfi0cofaVtSAFnR+SThComQ+LQwTsjViDVH68ylOtaHLNxBWptmb6vd
CmGFt4t8ii4lAKseOf60SrrCydTTkEpqibEyzgoWmYFR/qnICTuRDEmEiLVgU0jKaDhX4Tyfazh3lv
w7Wva2ZnvWurZlr0t12nzU+TsXSke0RCq7qhhjitpjKeOyygoOyalx/lm/SU8RDWI5uLYgVxjA+Drn1
gyaS58HGZw2K1Z2XsvRaw+FqFtwOs+E8wsRapPkPqq1qlRzhjpc6g3CxqolKb2R+sm36VahDIlw
qLbAWki5CD0PQlFmd6dN91fNZ6zMq8fOl7KrXufRI7n6ELfUd8ElwFfXJ8e3a9fap88GEPfe99akq1fj
x94DLgfqJ+QhcbfyV+o/lDe1qEXN1we3+DzsjTe6iDRbQje/KC4Z9mfDiCg4qTfHmCK3V/fMSB5W
AFfReBrknffW2BeDI8fc+BRBaffhCBj/AHeiSOMduQeleMvscQeo6LUBKCeueKHJ3uez+l8olle5SH
1nRje/CG018XfFKF6UkhfLGFBjGrfQ6EQRZlFRKdJ+44eROYANnZeVCVqcVweY+TWbECefKRvP
yRelWPZYgXereNinNfeySLx15Ce5uKHkh6fAeliDErfGCHNXX8fDCFqOyGeDOhztg+eHOeocPueL
Wbeq+deSCYpptOeYuV64b8efaTTnJtenGQyF18eveOYUfge22MGDC9e7+KRBXPe1+Hler8d4Or
mNa8d8innMJ4eA+jpK5GeH2gBpoZeO2cgoXseV+ZH3GEeeCV2ly9emqSvEdQeuCPyTBleyeNfB
XZepOJb+l/dvi1hdVYdzqwvsEsd36r+a0Xd++niJkleGKjNIT7eNmFA3C2eV+a+FwUee6XIEbRemO
TgzAYeqCQxBXmeeGLDugJdo+/jNP4dsq59b/jdwe0Xqvqd3mvHpf5d+2p/IQNeGalAG/qeO6gMVtt

eX6boEZSefKXWS/MeieUGxXweUiMcuajdkHJndKWdnjDKb6KdrC8uqqudx62u5bdd4+w24MUeA
SrlW8Velqllq+eRqgUUXOeYubZS98ebuXGhX6eMeNnuU4dgTTzdFAdjBmN71KdmjFdKmKdtO+s
5XXd0C4D4lud7CxxG5QeDOrRFoceMKIREVTTeTCfsC8yeV2ZrhYBeFmOnePUddTeLtAldf/Wfrw7
divOwqiUdpPHLZT6dv2/soFsd2i4Wm2qd+ixO1mUeHaqdUTseOKj8i70eQyb1BYld/6O5earg7p5O
9MUgzd5s792grh6L6vAglp6wJf4ggJ7TIQQgbN70W/dgXF8Y1s2gUF8/kXfgSR9IS7vgUB+DxP7gc
9+mOTigliCsNFngemCTb3jgX2B76pAgTWBtZaWgPKBfYLZgLiBSG7PgImBKFpPgGmBG0UggF
WBDy5bgGyA/ROcgLaBDOMogSWMCs/FgMOKzrxbgGWJmajbgC2lnJVvf/yHpYHAf9OGtG3bf7
WF4VmAf6SFLER3f5qEfy3Xf6uD5xNKf76DOuFzgCuVZc4of9KTUbrXf3qRRad6f0ePhZQYfx2N0
YCpfv2MLGzufu6Ko1jAfvCJOEPifvGH4S1ifv6GyxMCfuaFH9/Tf2Gexcyifwib5LlsfrCZC6YtfoGWhZ
Ltlf2UEX+jfkSRs2wQfkCPcVgMfk6NUkNYfleLUyz/flujxxLZfhmG4N5YfruoJssxfmakf7glfhGg4KTM
feedmJHFfcaaZX6ffbGXs2syfbGUVFdZfcKRiULTfcmO7Cyvfb6M9hLofUalldzxfjOxqcnffdytQbbMf
Yao3aPEfV+kzZC/fUGg0324fTCc9mpufTOZQ1a6fUaVyUJdfUmSjSxqfTGQOBL6f12KFNUncu7Q
siNfW+2E7WNfRSw56KffO6sE4+4fNCnV3zSfL+iuWmqfMOeTVYcfNeaJkHnfNeWTiwlFLOTVRMJ
e+6LXNovfX/E48cwfSG+4rQ0fMG46qFlfJazYo6ffHSt8nvdfF6ooWjafF+jhlVzfHKet0FqfG6aSivafE
OWDBMVe2iMcNjGfUbOpMXafOPH8rLzfH7BSKBBfE27CY2afCa04Xr5fAqu12gZfAipB1TWfBmjj
ED1fBGeiiuUe+GYYxMgevaM7tdifRjYlsShfK/RaLHkfETKNJ9LfA/DLYy+e+O8Ono5e8K1ZWd3e7
2u0IRSe8yookCTe8OicStZe46aVhMpepeM9Nltitl4J8bpicB4urRfiK15UKG2h8557Y8BhvV6inw3hi
N7KwKWhWF71IV2hLV8j0EfhCh9QysOg/h91xEohEp+ptfpiXmBM8V/iHSA9rMOh3WAvKB6hqiA
ml3hheKAe3s5hSOAaGg3hHSAbFSyg9qAhkB4g1eAoSqliKAsxDTgzKA6NZ4iE2KIMQkh1aJFr
HJhmOIEJ9ShaSHMYzYhOyGXXpRhD2FmGdsg56E9FQCgxSEbz/kgpuD9CoSgmCDixCJgj2C5
NTvh2GTB8KxhnCRM7BuhYKPZJ4XhMiNzlu9hBaMRXIXg22K1WaZgtuJg1NZgmGIUz9igfKHNY
mqgbKGXRBjGWeEndNxbqKb/MFMhbCZY68jhMGWzpzohAyUe4qsg16SP3hngqrQIGXOgjCOH
1K3gcCMRT7ogVeKjCIRgQ2JSxAngJqGONIUhguk+b/0hRqhoq3UhCyeT5uyg3mbRImTgs2YUnd
xgiyVgWT+gaSS1VIQgTSQWD5ugMaODikLgGmMcxA6f8KH9y9DBHyQuEr61hJiqAqypg6ml9Jqdg
viiK4iZgk+efXaWga6a8mREgSiXlIF8gLiUdT4DgESRmSjOf9aPnRBPfwSJLM9rhSO3P71whDCyc
qt2gz+tpmDgo2pl4eageKkvXW3gUCgfGOJgLmccldmgEiYsD2Vf8+VRyiPf1OSbxBhfmCKW84Z
hNnAd7wbg+S646ojgu+1VJhNgjiwJoaFgYirFnTGgOGmLmK+gFWhgFBEf+OdJD0ff2SZNSHJft6U
5xBwfdaLHcy4hJ/J0LrLg6fDIKjkgq69XZcqqfK3gIWCgTuxwnPlgl2sLWIBf/6m10+tf4ih3jyfwfadUig
HfniXBxB9fWKLJstZhHDTXLmVg3TMqafXgnbF7ZY2gbW/UYsngPm41HMogEaygWFhf7Osd08tf
zym3DxTfrWg3CfRfiGYzRCIfQCLLcxNkj3WLRbkH54CKIjs94uJfDjWZ5YoYdjAl6DXRmirt6u2JUi
Xp7cE+9iFJ8LDxnh1N86CcwhtR9mA6Xhoh+ucsHkOGAAbmrijz5/66hljaR/1Za7jEx/xYUriv9/vnORi
cB/wmGZil9/2E8Yh3V///zvahn+ALCa7hfqAZA5MhXOAyMnHj8GlfrAjijqHp6c0jJqG0ZXDi1CGEYR
OihCFXXLOiN6EuGDuh7qEL06Fhq2Dwjtcbb+DYyZUhTaDKw4KhIGClshyjuSQ8rdBjVCPXqYMi8
KNzJS2inyMX4NdiUGLAnH6iBOJvGA8hvmljk32hfqHfjruhRSGiCX5hlaF7Q3Rg66EKMcnjimZerY
KjJaXKaTriwqU25OticeSvYJvil+QtnEph2OOy1+LhlCM+E1qhVmLRzqFhHaJwSWrg92lzg21guCF
n8XojZaiCrTPjAafBqO1inucBJKPiTqZOYFsiAOWhnBHhtiT817NhcWRgEzShMyPOTToXg+CNLSV
ugzGL8Q3MggOHE8StjRmqtbOni4mnAqKgif6jTpGPiL+fzoCFh4mcam99hl6ZKV4ihUqWEUxKhE
+TMzm2g1uQpCU5gpaO3A3kgUCIWsnUjLazd7J1iyWvDaF+iZiqpJCEiFimdX+VhyCiY26rhfOeeF
1yhN2avku/g96XTDIRguOUQCUAgg2Raw34gJmJcsIjG+8RLExit23H6A4iU2x/o9YiAetLH6lHsmo
em3ChZWj8lyvhHqfoUskg3ibozjignaYIyTAgZSTqQ4KgAyJfsDijDfFNK/uiqO/cZ8DiQ+5sY4+h8S0
On2Lhn+u5GznhUSpu1v4hCokz0qSgx2gQzh5ghWb/CSDgSmVmQ4Zf5SJiL+RjAnOWK7EinLIp
3+iNvB5l1Ph4q7uXy1hkC1sWwvhP6v11tdg9mqR0oYgs+IjzghgcKfMyRQgM+XNg4lfzCJkL95mc

N2wq8DI393cp6LIUR4JI3zk0h43H1ZkV15lmy0jd4d6U1uwbj97EEoejBR7zzfEip98kyNdidN9YQxEiJ
d+2r5omlJ/Cq4Jlk9+/p2nlCJ+9Y0fkjR++nyYkFd/CWwljo1/JVsYjNJ/TEmZizJ/fzdRicN/vyL6iPSAH
wwEh4OAu71Sl2qHHq0NIUCGXJzEkx6FnYxWkT6E/Hvlj22EaGtqja6D5FqOi/2DdUkhimeDHjbriP
6C3CKjiCqC2gvMhpKCX7w2lpGPJKwFIgqNsZvSkkqMQot6kG6K/3shjqCJzWq+jOOIsFn9izuHpki
uibCGtDaSiE6F5iJWh3aFjgubhcCDzrscld6XO6r8k7eVF5rbkZaS94qYj7yRD3pWjfCPPWoNjDWN
hlloipGL4Eg4iQyKWTY6h6uJBSIVhseIZQuFhPKFJboFIUmfW6nikyickZnBkQuZyomUjzOXPNlsjWi
UymIAi6ySdli8igaQOkewiH2OKDXZhXGMXSHihhKLZguhhBCGe7jolM2nmKjUkq6kJ5jEkJKguoirjr
udg3iZjPCaaGiJizOXcVgiiYuUnUc3h/6SADWDhoePwCG3hW+OCQu9g0iHqLe+IGyv7ae3kk2rz5
e1kC+ntYexlij2Xe5jlqgG2fFismchVd+iR6ZHEa2h4uV+TUohguTSyGHhN+QWgvWgpylALaVICS4
TKaEkgWzgJZ8j+euu4aRjgqqRna2jDal8mblim2hyLBiEiLud1UYjhyOaNDTAhZuXGyFNhF+SYwvqg
gylDrVck+zA0qVRkc67c5VTj622GYV/jcqXCHXAI+6sG2YRih2nXVYTiGSi20WZhseevTRdhTeamS
EVg/CUJAv7gZGIGrQgk8DJkKQ3kaDDwpRYj3y974SYjZO4NnTwi7Gyo2VeidmtQ1V/iBqK0Ukh
nmjaDQKhOKdgiDng5GVmwwKgSuJLMboXB2iqOhnnd3VJQnm4x4HoSMmQR413TvlpB5ImVFI
DV6W1U1kfV7IESPj+F76TMVjhh8vx9mjS99swnuiUz/irJJoD1+dKLcnVR+jJNwmnd+ooPhl/p+tnRT
IZF+1GS8k0F/AFS9kQp/N0Qnjvx/fTK+jTR/2R8fjEGAcgnOicaBPLFlnzKGK6lNnFKFlpK1mX6FAY
M9lw2Ed3PDIK6D+2Q+kmWDkFRQkDaDOKPJji6C/zJwjGeC5B7hi2qDLgmyiMqCubB/nl2Nz6E0
m4GMIJHrmLCLWoKFlkCKNnMek+OJJWOukZyIK1PZj3OHQ0NtjXWGdzlti7KF3R6qiqyF5gmZh+
6EBK+ZnamVeqBYms+Tm5Eal/+RvYHGIzGP/3JzkwOWGGMZkO6MzFNdjsiLU0MNjM+J+zHpiw
ml6B5+ifKlqwmYhxeFNq6wnRSdKZ9nmkGasZAl3WYOoDklQmV4XGokqyTomJokGORg1LJjjqPf
UKbjDuNpzGcimmMKB5hiS+LSQnGhiuGYK26nJak955+mcah449Klv2ezYAaIJGb1HDykjSY+GH
Jj+mWQFJFjb6Trkl1i7mRWDFXidqPcx5liGNkgnxhVqGuayxC+s352AmWKpJY5Ylpqla388lCyh1
nAqkcyEYmEcj32bF1G1jU2X+kHFi0OVKzELiViS6B4oh+aPIAoXhKiG0quhm960z5xnmRWwco06l
k6sF34zk9qn8288kXSj8mBPjx6glFELjOecg0FAiteZQjCviOKWZR38h12RWAo2hBGG56qAm6C8
55tMmNe3/Ywnlg+zFn02k5WuYW5YkSep0l+LjsmldVBojl2hV0DCinadqTBYiHmZcB3QhuWS3gp
Rg5GG+alfm3DFOpplmKW/4ltAldm6gXxgk1q1MW2ZkOawCF7mjoKrF0/fjD+mc0BXiiSh3TAOiB+
b+R2rhoCUDwpngyeHCabyqVB2f5hspb93RYnroix4C3tMnxp4yGyrnBB5jV39mSN6XE7gll57Kz8h
k9d8AC6CkbJ86ht9kLB+CgffjPGAJqZXqCh+CpfcpKR+KlljoSx+R3rNnhR+amw4mxN+mF2ZmC5
+1k6KIW9/Hz7Ykut/ei5lkMF/9RtVj66Azgfai8aBr6WppyiFZZc8o6uE4ojToDqEYXpRnSqD7mvMmj
GDil07l1KDOK46lJiC/z6WkheC4S4Uj+qC8RsyjseDkQfWisCDB6T2plWMqZaSotuLiog0n2uKbnm/
nF2JbGtJmWOIfVzKloWHpk3ek9GG4j5SkVeGPi3njyyF2xsTjfmGPwfTiduEM6Q/paCT7pXioiiSNo
eLnrmQgnkkm6yO72q+mLKNdFxSldSMFU18kyKKyT4JkKuJpC24jnul1Br9jTGlswfliPuFRaOGpQ
qbMJUcoZiY7Ya+niyWq3hnmX+UhmVmCWSfFu/IUSQIE0Cko+Oxj2ukBKNLC2AjdKL/xr3jF2K+
QgiiAeFgKK4plqikpRYoRufuoYFnbGc43e9mqWaKGI8l6qXjFs6lMaVFkyVkg2Sxj1bj4qQuS1OjTu
PNRryi6CM9Qhbhy+FpqHQpB+qDZN7oLKMmlUynUmjJXb7mjuf2WjNlz2crlqjIFaZsEwYkZiW4jz9j
w+UaC0RjLKSgRrjiviOswiMhnaFx6Dco8mxj5J9oF+tgoQunPapeXYKmeOlP2f1lt+h+1nPk/Cef0t/kS
2bPDyGjpyYXizDjDwVihRfimSQPAi1hdqF45/Uo4S5OJF6oBq0qIMznLCwHHUimZerw2ckloynk1k
1k5ejmErrkM2f4TwUjjackix2i8WYMRqmieKRjwjYhVWF+57l0zBG5CLn+G8HYJfnHW3H3RemV
ayN2Z0lkateliek0uo+UpukHykyzu0jeCgOyw2i2aaZhqLiXWR8gj1hOeGDpr/sWh2q11orU93UH/YqT
93/XlYpXt4vGSKodB5h1bUnkh6YEimmvZ7OjnKI/d8HSoBIXF9HReclFx+bQYNjr6Ar5qWsEd90o0J
rDh933+DqC998nHipHN+l2RCoNB+YlaXnU1+s0h0mf9/DzmiLv5/gCnmlG+AHReVlz+BPQYhjYm
CFJohr06E0oydq0KESn8gpz2DyXGQo4iDbWP9n+qDlIZcnG2C7UhDmSKCzTI9liCCzynPk4mDD
RePkkKD+AYzjHqDTZmfrn2LulwiqnCKn36tpmiJj3ElorSlrWOdnxaH31YLM5mHLEgEmFOGjTIUIV

eGEym6kr+F6xeKkWGGXQZDi46EPJkUrcaSmYucqbmQ9H4upbGPV3CvofyN7mMznl+MnVWxm
uGLbEe9l5yKTjkjKKJXSmikgOI1xeNklalhQZliqeEU5iHrSyZdosCqSOXTn2JpRyVLnAYoWiTPGK
sncqRZIU7mkmPtUdelwGOIDjflACMxSl/kU2L7xefj52KgAawiauEhZfirKmgblpkqKCduXz0pJqbDG
+PoOeYj2lXnUiWMVTSmcWT/UcllniR8Tiik3CQMClgkKyPDhewjs6MOgb1iM2EtJcerDuneYmoqD
OkM3xBpCug9G7qoHad72GdnNSbDIRSmU2YXUagl2V3jhVku6TvSk0kBuR9he0jheNvgcwiA6E3
JZKq+GujoJlp9mqvHtZo9Cm9G4ToBajcmDbnG6gGFOsmOGc9EYXIYuaCzftknSXCjyj5SUmBelj
XWPEgdhh22E/ZVaq5W1yYfdp42xfXp2o4GtPG1An8KpPmAenBWla1MJmIKh00WSISWegzelkgi
bYSixjx2W5BeSjOiP8AeLhuSFGpRgq1S9PlcCp0y4hnm2oz+z2GyOn3uvV19+m8mrBVKAmdGm
9EUHIM+jHjcykayejSh6jriYzheCjHKP5QevhnKFMo86uZ53loKRtM93unXusBF4V2koq8R5C1xgp5
J50E+Mo4Z6qUl/n7x7iDRAnFR8dSVJmXt9iBOMmFV/BgRskFuBKo8luLB95oJos7t99HXLrwV+C
GkGqr1+OFxDppB+ek93ooh+00lznr1/PzQ+m0x/xyVUmGKAhROtlxaBzASWjx2Cbo7Qt4iEiYluss
eED3WUrhSDm2jcqdGDRlwpaqDCE9goaWC5UllndmC3jQ7mmGDAYVel2iDchPKlfyEOgS7jgeD
M459trOLEIHesfGKE3VGrTuJHmiUqPiITVvjpNCHIU8soMqG/UICnQGGgTQumYyGNSVnlo2GTR
PjIQOGWATcjRSDSY4dtfqRkoGAsTeQFXTrrlCONmg/qDyNUFuYpBOMH07uoAyLEkHWnEKKIT
QXmMyJaiVolcKJNBQCIBCIPQUNjCaDao21tVuYDoEKsJyWFnRrq+eUJGfKp6OSV1swo3mQqk6
Vn2+PJkGQm6CNyjPrmCGMtSVdlP+MPRQvkw6J8gVkiySDpY02tNSemoCQsBaclHP4q2KZrGd
gpx6XXVrSovKVMk5FnuatoEFRmxKRbzPCI4qQAIvRIFKO/xRXkimLbwWzij+D24yYtGOIOX/2r6
WiN3Nkqu+fPGbYpqmccFpYonqZzE3cnmqXX0D9mpKVLzOHlwCTcCU2k7eRdBRukWCMuwX3i
XuECYvitASr4n80r0aoX3KZqo2k5mYdpkOhplmwog+eE1MnfibvUCImhqZMDMvloCXCSUCkyiTq
RRvklCN0QYxiNaEL4sNs7Gys35lrvSuv3HRqjq1GVkpeqnIVkKobCjoEy+nZSgYUATma+dejLYl
g6aOSTNkqqVIRRqkBeNzQZhiEqEUloos2i5wH2grq21YnEoqfOxC2TlpZ6s4Fh+oV+o60xGnT+IQ
z+xmVWhgDKOla6c4iSgkj+XMBRmj5WNywaKh9WEbIPBwgB3ynf3vld4TWwytyZ41WBBsIF5eV
RQRzP6M0hUqQl7CTvhpMV77C65oPB85iCOncV+Dg/Nm/Z/ywL+kcWBIYPRwOF+L3gJu3J+N2x
Ethh+RWBQsUZ+blRgrJF+r0hqqAJ/Dzv/o7d/iS7in9KAKyDFnluBEBASmpuCPwM8kH+CMIPXv+
SEdngGunmEA2w7tSKDlmbQsFWDRFRnq6ODDUh4pxaC+jwVosaDDi8EntODWiD0m3aEARBO
mWiEZgNj2KCVYO4vwaKonfhuZuJu2wTtEKl2mAqr3OIFIRlqsCHb0hjpjGG8DwRoeGGmC8Xne
2GfyEemoOG4xCDmFuGRQOijmmCdYOEvkAqYXeruNuPbWvbs4COF1/3rq+M3lQcqqfLyEhCp
WmK3TwBoReKHC8cnR+JpCE8maWJxhC6l1aH7QPgjXaCnoNAvZ+W5ndauDiVF2uBst+TTF+lr
g6Rn1PUqViQF0gHpMSOwDvWoGyNnS8lnGeM1iFJmNCMcRD8lkOJYwRBjG6C4YLnVRKdC3c
Et6yaw2svslOYf19arYKWWIOSqMqUX0fQpDSSnDuun9aRGi71m8WQCIFTmBSOwxE3IU+KpwS
ai4SDHIJtvJqjPnaMtzSgeWq6sdqdu17wrQablVM0qEuYt0eAo7GWjTtwN06UsC7NmzOTUCFJI22
Q0BFfIHlVwTmiruDUIHWvDSpfXXqts2mQmoQsXCjEF5TrJegDFKop9idPUcloziatTsOns+YhS6
GmquWViEkltSSpBFwk7mLygUmihKDe4Ecu9qv6XU2tnSsQmllsRWopF22rDeIN1lbp3GiA0aPos2
fHzqrnl6cdS48mjOY+CD7lk6UORF5kxOL0AVciYODoIBPu4m2lXSJtiaygWjUsMiueV0yq+WqnVGI
pxunA0YponKjqDpYnf6fyy3+mc2bKiDYldyVixGAkoeL1QWKiQuDv3ghyoN40G0GxON5D2H4v0h
5W1boudZ57UvatH56nEDEr097cDU3qnh8WyjypiJ9ZxuXopZ+rAxcnzCAgQGukxGBI3iWyWh+5m
1pw8Z+sWJKvil+ilcxuLd+q0wgs2B+60EOrrJ/UzWHqU9/4SINpN+ApBwCoSyBvAzBnb2CqQH+kc
WBWXjbyGiE1G2gwsOEMGJ3vSKDmVdj7WDSUxWsmKDGkFIRtODGjXJqEeDTsmao8ODyhyf
n++EvA0bnHaEjQJEKGBiHj5x36KrW2wwdaJn2J7vDCIn1dmtsCH5kxdsWuHU0FXrDuG8DXpp0
yGwinTosGG6BywntyHkw1om1iGMwKBj6OBsXj0xrWQdG2mwQiO/mJtu12NI1dateyMdUxVsJSLf
EFYq2KKuDX5pm+KLSn4od2KAhzvneGKGg22mkWHowLLjquB43jhxgiWKG2DwFqUSml+uqySe
lcwtTqQ7Ewyr+KPjEE+qq2OaDXtpbSNiCoBoRKNNHR0YnPSMUw4MmSOI3wM2jZ2CLHisxXOb5

G1Tv8OZIGISuhKXU1cJtKCVWUwSr0eTkkElqhCSDTXgpRCQ3CoHoGGQKx07nCSOQg5YmCK
JswOXjK6CbnhPxPOhsGz8vz+e6WHCuYqcM1bBtBaZ0EvSrruXpkDxqYGVxjW6pHyUSCn0n8G
S6h1Fm2uP8w6Plz6J2APri+KCpnfVxlanh2x6vs6kUmE8uROhMVZHs5mebktmrjmb6ECTqPyZtT
Vuo/KX4ym/ny6VXh0wmsaRbg6vlnOJ7QQxizWC1Xc4xCWtimvmvmqp7mCxuKqmaVXJsyujRUr1
rcagZkAyqIWd5DUfo3ablimFnqyXfh0UmjSStQ7ElcKJ/ARsiqOC/naKw9Gzx2tWvhSvwmA8uFKr1
VVfss6oTkqXrWWIFD/hqCGhvTTcow6d1yIVnj6ZRBz9mbqTmg7WIS2KCASeiimDH+lfeVZ0/9WN
ea911cG2egp2rq3Qenl3jZnUeud4aIW0e1Z5PXFJe9J6ElxsfFt65UbjfON7rC/SfWI8ShTPfeJ8r+dC
d5h+49OSeA9+0r/celH+xKwZeRV+y5hFeaF+14RWeix+6XAbesF/BFtve19/JUYWe/Z/QS8yfHh/Th
RlfL9/O+VUdh+lpNHEdqqHr74vdzeGvqqNd9eF+ZbfeHmFPoMbeRyEjW8MecmD7VqNenqDW0V
hex6CzS6je5yCTRQle8KBuOOQdOiSTNAYdYKQdbybduOpakYdsaNf5WPd3OLmYH2eCaKLM
4UeOSl0lnBeaqHiETAelUGSy4metaFRhO3euaD5eHic+Ob686EdISZQrshdSOWn6e8ddeUSJRvd
o+SBYDkd02P3G0reBqNwlkEeO+Lv0QueaqJ1i27eh+IVBOlehiF5eBScw+lhc0Bc7eiELmvdFyen6
ZqdRabg5MmddWYfX/adpyVk2xJd3KSv1hLeE6QC0OjeQqNfy1neXSLjBOTeUyH0N7ScmCvPcu
dcwerALhmc6ymxqU+dGyi3JlZdTKfC37vdgCbWWuAdt2XxVend7+UXkMpeHyRLy0eeNyO1BOie
JmJft1acdO5Cco9cni0Abcfcxuu+qQWc92qR5EQdKalr34HdXihN2q7dlmc5VcGdz6YykKvd/yU+Cz
WeFSSKROveACK79vxcWbC3sjacgu9ALXlCq23J6Lfc2yxu4/8dDOsaX0XdQSnOGnudeWiL1Zdd
sydZEIwd4iY9SyJd9uVJRO5d36MJtqCcRfM08eDcbfGPLSLclO/q6HBcw65gY79c9GzcXw5dJ+tg
GkxdX+nulXCdmaiO0G5dyCdLCxCd3GXtRPCdxCNLtkbcOLW9sZLcXbP2bN+cgbItKDQcr3Buo4
nc32613t/dEi0FGiSdSitg1U/dg+nRUFWdsmhWSwGdxaZ2RPKdrWNYdx0gBV0Dcmuf+N09rbIf7J
14qQJf5d22ZEaf4R3yX4Lf314smqmf4Z5n1bFf6F6jElxf8d7bSv4gAt8lRHlgJZ8t9rAfmR9asgWfk99
ebVofjl9jaKjfp9u4/TfkJ963zsfIV+HGmrfnd+WVXsfqh+oUF7ftx+4ytnfxx/DhGHf3p/PNkdfPGGoMa
OfPGF27P7fPCFG6FXfQOEi46pfSCD/3vmfUqDeGjlfYGDBIUqfcWCqUDafgaCTirmfkGB9xE0foG
Bftdx9GPzcT5e9uONbJ/e+OMpJ/+e/+LV411fCWKEnrZfFul2mflfKKHtlRyfPmGqUBMfUaFqSp1f
X2E2RDsfaiDdtXSet6ZA8N3eu2WoLEaevqURZ64exySN4xRe0qQOHnae4qOSmcOe96MclPFfE
GKtT/lfJaJEyoUfMWH0xDEfNqFSNRaehmiP8lJei2fGK+5ej2b+J11emWZLI suepmWcnj cet+TzWY
3ezmRRFMYe6KO3z9Je/aMpCnlfBWLahDVfAmHDNLveXOrmMC3eYensa6BeZijz5xVecSgP4or
ef2cwnf6ekmZX2V4eqiWH1J+exSTED7Ye2aQPsmHe3eORBDoe1ClmdGEeO21Ar9ieQCwV61D
eQ+rsJs0eTynYlkoeXejJXcYecWfB2S5eiebE1HlepWXWz5neuWT8ilFeuuRXxD5erKJ7dAdeli+bb
4AeJm49avreKazhpn8eNGuhogTeQqpXyYoeVSk0GPvebWgMVfBeiOb1z3uem+X3yj+em6UFhE
Gei2LDM6ueDrH8ryneEfBy6qpeE+7rpjaeHe1/YcSeKywYXVKePKq4GMzeVGikVCqeb6gkT1+ege
cDCi8egCWbBESebuLi81Hd/zRn7tteAPK+qmZeAXEU5fmeCq94IY6eFy3gHSOeJ+xOmKVePurLI
AreWelgT0gea6f5SiEeaKYXhEceV2Lkc/Dhv1zMr4Ahj10LKw6hYF1KJpfhOd2KIh4hFd3JnZ5g9N4
JGQag2J5J1EzgwI6LT2Tgsd7Jygsqsh78A9Fgwd86s5nhU98GLy7hKt8RqsNhAp8dplFg4x8t4d3gx
Z8/3WXgqt9TmNSglJ9rVB/gg9+GTz2gdx+fyeqgdh+yQ7xge9/P80Og+qE07t5g1mEOanigsuDo5g
5glyDL4algfCxHTFgaCCZmKZgVyCIU/dgS2B9DxqgQaBySc4gP6BnA6ogPqBS8uOgtKNh7oQg
kyMKaiRgcaK0JcFgWCJqoV0gQWlknPUgLiHjGHOGlKGNk88gGOFyTvvgEeFAibSgDqEaQ5ogC
SDEsoMge2WUrisgWmUMKdLgOWSFZXbgLeQOoRpgDOOcXLqf+2MwGEKf8GLJ06gf62JrDt7f5
eITSZ9f4CHUg5Hf1eEusikgTefMrdNgLacUaX5gDSZeJSmf9qW6lNTf4qUbnH4f0iSD2A/fyCPzk3/f
w+NtTsGfvSLyCY6fsmKeQ5cfoCGXMdKgJmKbYKgbmkkaTMf5ig/pOSf0Gds4JbfvWagXEhfreX
bF+KfpKUfk1vfoKRxTqffmOPSyYBfiaNpA5zfcOHY8XvgBixLLTBf5ms2qOWfxiojJJ3fsGki4FefnWg
pHBFfjic3l7SfhSZR0zdfgSV8Do2feCS7iXGfZSQdw6HfSCJBcSXf7i6MrNnfzq1HqJBfriwEpFAfl6ra
YBKfg2m229WfciucI4LfaSePEw/fZKaTznEfWmWziWEfROS8A6YfJeJ3sMyf2/DVLITfu+9mqEAfm

q36JAdfgqylH9HfbStXW53fWyoS11RfUKjckusfS2e7jlafP+a3CVGfKGVEg6mfCOJ58HSfzjMoLDdf
rLGbZ/xfifANo8nfcO6KH5sfWi0N227fRyubVyo06o5kswfNajxjkBfKWeXSUSfECW2Q6xe8KJ78M
mjiZyhrJjjNBzkqGfi4F0npDAimV1pn/aiVd2sG7jiFp3u12Lh254yEulhpx51Dj+he562CRjhaph7uAzjhT
99J8H/jlZ7BbFQi017UaCgihl7no/UiRZ78H8FiB98S24ohzd8sFzkhmB9IEsQhaF9mjh5hP9+EiPyhL
h+fQyZhCp/Q8DaiyyDS7BBigaC35+niOOCdl71h++CHH4/hwiBz216hjKBj1xMhW6BZEQJhL+BTT
gEhCiBPSOOg9yBPQxYgziBHL+UiWlH68QiQWKZp6Nh+iJR431hvyISH1ahh2HV2yyhU+GeVujh
JaFrkoDg/WE+Dedg2eEViM2gxaD9QwfgmaCuL5CiUyT5a3ZiC2SCJ1xhxKQLozyhiyOhXxyhVOM
72vohlqLcFr7g9iKA0l+g0GlsTc5graHgiLrglqGzQwEgZmEOrz3iJ2cWqyVh4KZyZw2hmqXPlvShY
eU6HtvhLGSqmsHg+uQhlo+gzuOfUjpgqOMmDbQghCK6CKygZuJ8AwegL2FvLu1iAik6Ktmhu6hq
JschedeaorPhPabZHqHhCOYd2o9g1+VqlmWgrCS/0hlghaQhjZzgXuOVyKBgPGM3ww4f/uHD7p
wh4+thqowhnapkpn2hV2loYnChH2h7nmWg6meVmlsguSa4ljogjSXmkfdgZiUkjYSgPaR6SJNgFm
PcgxPf1WIMrkuhZC2KajphhqxfJiuhQKs1oiVhB2og3iJg0OkT2iEgnmgQVgngcWcZkdGgSWY1TW
ogHyVvilRf9KRtAxifsmIX7fhhuS+6aemhc+5n5d4hLa0W4d5g8yvZneLguyqkWeqghyl51dZgWOhd
Ua5gL+dXDVEgBGZjyHYf1yTpgxyfIKlaraWhqfH1KZ/hZHCfZZzhHa8UYaKg4e2rHa2gqKxKWbyg
cyr01bagRCmwUZCgGmiHjTwf7WcwSGofveVRQyAfe6ldLbGlXxyAab0k6RzCZclkdF0E4dAkCp1
JXdbjpR2OmdqjRV3VFcWi6Z4a0YvilN5fTR7iTR6jCCjiLB7hggq6h0d9bbXDK/t6JKYKkj6dZZSkHd
6yYaBjuR7LHawjWF7m2bUi/V8FlaRipd8mEW4iVV9IDQQiEB9qyBEh7l+Ogp7hjV/VrTPkq6CB6U
skPiBqJWLj0aBTIXPjcOBEXYSjFGA4WZJivOAwwYXiaWAr0VMiHCArjOyh2SAuR/vhtmA6ApDhU
SBARPHkauJ0KQ2j/ylypSojlGHYyUDjNWG8XVdi2mGKGWsihGFclWTiM2EyUTjh6SEMTNghqCD
tB+mhhCDjwoThHOCdrKqkNeRv6MwjyqQB5O5jYCOVoQrjAmM3nSciqKLe2UEiU6KLIUHiBCI7U
R0hu+HwzMNheuGxB9ohU6GWgn/g6aD1rGkKc+Zx6lOjoeXaJKYjOCVEIMhi2uS9nOuigaQ9GQ
1iLOPDVRbh3SNOKPvhk+LiTKxhUCKFh86hleJXwoegsWFObBjj56h6KD3jfme5JGTjFSb5YlyiuG
ZJHLXiX2WfWN7iCuT91PAhuuRj0N3hcKPVzJehKmNcx8Sg9SMCQo9gf+Gcq8zjyaqG5/VjYOmb
ZB/i+CixYE0im2fYHHziQecGWK2h7OY91MdhGV/UL4hUOTQzIHhCCQ9B7mgzWOYApXgVWG
/q4GjseyVZ6fjSit+49Gi4epq4ASig+lrnDviKSh02HWH0qeIJKhgKaoUJqhM+XbDGIg6OUuR6wgqiQ
bgptgMaHDazOjn26sJ1ujN+1xl4eiz2w338AicCsSW/4iE+n2GEChu6jIFG2haGfiUHjhGib2zFHgzW
YNh59gi2SNAqAgEyHGquajkPDPJxXjKS93Y0jiv64fX4YiX2zQ28oiAauLmBQhp+pTFEjhU2ksEFy
hBGgcDD5gtebHR5TgcSTrgqQf+aHJKrXnPFx05vzmmtY840UI/B0EX4klcJ1JG8yk6l2PWA1kat3X
VDMj8p4e0DGjhZ5ly/njKp6tRy6jBd70AiRiYp+O6oHm4d5mps1mRR6Dlxolqp6fX2llx67W6nkoJ7a
F+8kJR781BijsB8hEBojRd9Hi+Wi7B9wxx1ixJ+hAhuiG1/86k2mlWBIJp6l+qA7YvDIYmAu3z1k3iAl
G4mkXyAel9Mj5qAcVAAjdKAd0AUjDKAKC9Ois+AwBw4iieBMwhQh3KBdKhUmVqIkJmqIveHvos
EIJ2G7nxKkpCGNG2PkJiFi7JjrQe9U+UjPuEaz/Ai2WD9y8PigeDqRwDiVWD3Qg2hpiCxDumlyQ
FpjQliyOoYo5k9ONL3uTkcmL3mzsj9SKol48jfmJgE8gjD6laD9miq6Hai7QiU6GphvailqGmwg1hcO
D/aaBl+GXqJfbiYmVnYIAkzaTIHqkS6Rq2wfjzmP2l2KjV6OJ06Ni6KMhj73igyLDC6liJ6J4BvBh7W
JQQhkhNqFM6WFI06fXZbtIPuctYhgkqyaDnnfkKWXiGtljrGVHVzqjNaS1E4JixeQqT6SiX2Osy5HiA
KNlxuthveLkgiShA2Fy6R6ltOnKpXtllWj3YdskjmgkXj+kDGdcWqbjuacVw8jF2XmE16ipuU5j4liPuS
eS3/h3SQjhuShk6Nmwi4g1yF5aNuInGu+JTVICerCYZNkd2nH3f1j9Gjcmmtjdaf51twi/GciUzTiiqZW
z2liIOWfy2phvKUChtqhbiPZgjZgsiF+6JPlIO26ZO8k9qyb4U8kZCt+Xb3j3+pvGjKjX6lpVqti5KhwEw
zicWeEz0riBmazS1Xhn+XFhtDhTWQ8gj0gkqGDqErlea/EpK3k5y6loRUKU61MHYijzmW2gJjTOR
rloli0KnNkusiW+jBzzFh76e+S0Shh2ZoRsihMaSPaKlgeGGHp8JpJtxyZEVoyJy44MpnnNz/nUsm
7F1FGcvmQh2MklIn93XEqmICB4gzt8kgF5qitukEN62xjej6V8HAaoi4p+8Z5qo0d5NJCD0D15rIKk
nTx6JHS1moZ6oWbGI+l7LFjKIWx7yEpYkxd8azs6kP19GCs3jz592hi2jo5+1AafimKAfp3GoieAYo/

xnyKAQIIknCeAH3REmXyADWZjIumACIh2IHWAG0oPkimAOzr9kBWAcCsHjISAyBiSjZSBIaAyiV
+B2p0QoTaHeY9KnjaGwYGLmz6GDHO8mJWFcWXslgSE51gQk5WEdeM9kU+EDTrAj0SDvSrdj
YWDoxhzjLWELQaRiH6DCJxboGmOmo6cnW2NTIDnmnaMANMol86K2mVoIT+JyFeeKtGI0EIhkJ
CH5Dp8jomHFCqxjMWGjxhfi9yGnwahh6GEH5ugn72Vwl3VnMiT6YAYmdeSFXJplzCQXmS9IKG
OwFcLkjKNQUjppj+6L1jokjeGKKyp+jA2Jshheival7wbfhrGEpZrQnymdCo0QnDmann9emUuYNXG9
lqaV62QjIBeTvVaGkaeRtEh9j2CPyTnUjU2OFipQi2qM3xhfiimK9AcYhd2EzJnnpnykblwym8ChY36
LmNSeXnD6li6bhGN0k52Yy1XwkSqWPUgCjuCT1jI5jMaRuCoYitWQJxhViXSMugdKhSeE7Zj7nk
erzos3m16oMH2ImHOKl3AKlcmhOWKckzOeAFU3kLqa90dsjmqYHTkljEmVnSnQik2TNhg8iNOO
SwdzhI6FCpf3nfWzVY03mwyvM3yOmCGrF28ilXKnNWHLktejelSGkFef9EbbjgGcqDibi9qZwimJid
OV4BgiiEaPpQeXhAyFIzbrnbg7FIJmsi2fnu7I9qx6m5elSite2EckoipNIPwkAOILEZhjaehbDg/i3uda
yIOiWuYGRgMh8+QQge0g6CFNZNerIRx5oZbqONy5XlipUpz6mxYoex1AV9Mnqh2Jllym4p3WkS
ZmKZ4jDZJlhV5wCcEIAF7BBUFk2F8bwT5jU1/kpLvq0B49YX3p6t5XXkJPbt5zGwJoMZ6Vl8HnYt
68VH4mnZ7n0Rpl5h8VDYjIQI9Fybqkux9+BT9kjJ/MQUHjB2A+JKCqih/0YWTppd/qHiwowx/hmu7n
79/il7FnI1/n1HBmYB/y0Q8lqeABTYBIBmAWSbUkfaA2xT1kSKB4wUTixKCM5H8qT+Gj4UYpa6F3
Hg/oiKFMWtXntWEtF5um6SES1F4mJqD+0QEIdtZxbk0GDjybAkR2DqxTukDKERAUdiimDRZF
0qHGNT4SVpOOMEfCoVeK32rmngqJ314Lmtql9IEl9KIK0PCIQKHazWvkn6GyyaqkFOGiRTyj0
aGbQU9iUaDipDop8WUEYP7pDuSVncfoLGQompPnWWPHF2DmjWNsVCwlyuMaENmIFmLMjV
ukc6KKYaNj5CJmhULjkyIcwWHiFCDvZBEpy+a8INgo6eYqnaMoB+WamnHnNWUW10JmaWSa1
BHlpmQo0MSk8WO9zU0kTONiyZyjuKMshUijWyKNwXMh3eD64+CprGh5IKmoyqfC3Xan6KcO2k
inFeZplx1mSWXNk/JIheU8kKskz6S1zTrkKWRCyZLjkWPnxUjKeLxAYHhryEE46wpkqo2oHHosSI
eHT0nzqilmhMm+ufE1u0mLWcK08llaGZdkloksKW8zSKkCKU0iYRjbSSSBUIi/mNIgY4hh+ENI3lpf
Kv9oDhomysG3QTnuCoS2d6m46kwVr4mFOhYk6FITmePEGmklSbUjQqj6yYnyXWjTOUnBUVi2
GOQQZihZmEUYzapaS3R4AMoh+y/XNUnpSuvGbKmz+qsVpZl/+m1U3+IOCjN0E5kfaf2TPZj0ibz
yWkjMaWjBUJiuGOOQaFhSqEaYfitl9yT3vNsDtzQm/Aq/Z0O2OUqBZ1SlkdpFJ2aUsl0LV3nT5inV
140zDjmml6DyJmAp7YhEplyJ9DgN2juKAIYeks1J4+XuYrwp5ZG+Sqs951mNrpn6Y1dCozV7A0
sMn6B7uj5UnEI8fTDgmU59VSJult9+UxFEIdZ/zQOYjamBZodvsj1/fHtirfp/Y29eqcR/UWM/pe5/Xlcf
ojd/gErynqd/vT5Em1KAEDDcmFOAhCJ3IdeBMxFclK6CNwO2jJiCg4cZsVSF43sRrRGFTW8SqN
qEvmL8pQKEU1bmoUuEAERfNb2Dyz4nmmyDqDDSI3CDqSJ/IO6D/xFwk6iEUgPQi6mCIIa8slmM
R3q0rEeLNG64qA6KKGKrpDaJRFaf0H6le0qLnPGH0j39maGHPDC+lqSG0iKCIbaG2BGNkqqGN
wP9isCCsoZW9mSp3pAq5yRHm45p2aPm2l1o46OOIY2n9WM+EOynEaL3D26mPKK2zCUleyKF
CJ7k0aJ3BG8kZ6H9gRTicWC7IXZr0KZXhXnHqwiXEm3GptSVDWHLovyTLVXXn0KRcEnim7GP4
D18mFmOdTBulUqNViJ1ko2MpBHokK6JewShiOaDIYU8rsOfpXkwqomdEm00plWahmFDonuYLF
VensCV+UI7myuT+D0rl86SKDA3lLeQtyJfkeePIRIDj9uK0QTliCeDT4SJrlmmMXhxqiCjIWxtpemgG
2CLogydT1S4nkyar0jvmrGYRzy6l06WGS/mlC6UTSIykU+RYBIKjyCL/QUdh4aDdYO8rfys5Henqc
OpZGuppYyl7l/UoauisVQVneaf0hkmkWc1DxKItyaTC+Uk7SXhyICkMiTVhILjnyMMwVOhv6DloLI
raiz0HbqqXOv4WsDpT2r/l87oVmoSIOlnY+ky0fwmemhkzvtlnueVy9Pk02aOCHbkFaU+RILjfKMM
wV2hoyDsXyqvMxy2nF1t89zv2ZGsup0qFrXroZ1qk+VqkF2wUQmpiV38zg0olh5LCuAnvx6ch2+nF
R71Q29mlR9+AlkkESAn3ynu495IXF0tp95jGZGscN5/VrwrWF6iU+YqR97LEQxpQh76jhKoTZ8vC
uknc19qx3ymwp+yg34mPCAaAJYjwWBIXyrund/UHFutY1/Q2Y6sLV/O1rorFh/Tk+WqBx/ekQ3pAl/
yDhaoDSANCvCnL+AzR4gmeaBrQ4sl7SCiwKFje2BtHyJuYiFZHfItJ+E52YSr8aEcFrGq2eEE096
pymD00QmoxaDtjhZn0ODtSvWm8yD5x5ImOeEfw5Zlp6EZwKtjPmBz3xXuLyLcnETs9KKg2Xbrve
JmlqUqpWlzk9QpIWIIeQHokKHmjhKnm6HMCvcmvOHAB5ml/yHWA6MIZGGEALiJAUb9XwWuAa

RcXDDsyOQFGWArkuOvVpAqeiNfk8GpaeMYUPKoZKLcTgfnbeKqCvKmjCKKB53lxuKDQ7PIHe
HjwNFiwuCNnu/t2uXf3BusouVqmUtrbWT2lnzqVGSJ07CpQ6QmkOSoPePQDf3nRiOGSu5mYSN
UB6GIIWMAA8Lk32I3AOciieCcXtHtuidoG/4sgibS2S6rTKY/1mJqMyW205kpleU40NDoGyTJDe7nl
eRoiuVmOmQkR6BlaOOgA81kp6KAAPniWWCo3q0tnmjxW9asZmg/2QVrMCEq1jxqFabt03cpAy
ZWkLQn+uXPjdgnACVaitUmFqTpR5ilQKQYA9KkdiKVwQmiMGCznoCthaqFW6usTam5WNwrF2j
wVhYp++gzE1To6CeDEJdn3qblTcDm4iZWisRI9uWVR4/IHWSAg9YkSqKYARbiDWC8nlCtb6wn
W4KsOGtAWLlrAmpcVfYp5imEUzho0Wi7UH8nxmgETa1myOcvSrYl2+Ykh4hk/2TXg9jkJiKZwSlh
8GDEHE9xTJzrma9wAt0WVxNuup1EFHbtep2BEdhsQl3FDzRrFV4RjG7p/p5hSXcpCV62hjaoSd8
VQqYnSl+0gDskYqAoHGYw/t5nGcFvtd51VyEubZ6HllLTLV6qkeQr9Z7UT0HqyV8GTH8psB8/iYuo
tN+CxlBn69/Vwrxm7GA9wExkESAznHSwUB/f2cyvbt/SVymuJp/IVlus59/Oke2rsV/cj0zqhZ/0zlzpaq
AXyZ1oamBJRmanmWCRws+mmWC2AftjyaA9nHtwemFSGdBvMKERfYrt5yEHllysp+D0Ue9rcK
DpT1CqRODpjJSpKWDziaroJ2EORnonUaFGguCmUKEegGhji6BGnHtwROLAGc5u+mJ/VydsCJ
CFllsb+IVUe1rOGHxz1CqDCHZzJho7+HMSbQn6+HRxonNECHpgvJmCmF7AHkjTyBR3HYwF6
QmmcVuzSPMlxtgeN2FH4sQWMvUeOrCWLyZ0lp3KLDTJUovqKgibantqKWBpUm0eJ7QwdlwS
HMQJNjDaBjnGkv8CWPWbjupWUZVw8tWeSnFHOsGORGEdqq4GPwT0Jps2OpTJGok6Nyibhni
CNYhp6mmyL5wxplgCISgKsi0+Bz3FLvzib82aPugqZpVvutNiXaVGG9KVf0crqu2TxzzWpjaSUjllo
bKRKybRnXmQMRqJmauNpQyhlRiligL+iomCBnDXvsShuGYUuZGe/ltvtFmcWVERr0+aEEbDqm
aX/zyApaqWODHeoSGUxiajnN+Stxp7mP6PMAzEIEmlQNCieKCNHBGvl2npWWLuSakhlrus+m
hflCartye10Zaqe+cbTwnpS6aWDGWoKCYGyZxnFiU6RplmGaQhQzck5OlsgN9iVSCXG+ovgCtw
WUHuMmqOVqBs4qmylA2rnuijw0YBqYqhAzvcpMSeRTFboDKa4iZGm+WWvhpTl+eRogzxkvqlw
AOuiN6Cfd6adcJvmsuldm9w+bisdxtYV6Wld8hznPKJeHV03X9MeSV2FGvEeeN3Q1fKeq14ZkMg
e295eizOfBV6WBKwfMZ61Nx+c7F5fMmvdll57bbddWB6XqP6djJ6zJEHdwN7P335d9J7tWqieK18
LVbYeY58okJcemB9DiwxewN9XhJFe6N9bNqYcfSDJcfrcuuCsbU6c92CP6J5dMiB6l+tdbKBlnzLd
p2BTWmfd5GBDFYBeliaA00GveWmAmCumegmAXxHneqR//NjgclmMpsZNcZOLVLO3cpmKB6E
Vc5OI645tdl6H3Hu1dYqG22i0dpGF41VAd5yE90EWelmeEysreSaDWRGWecaCP9c5b1OWGcT
DcGmT+rJKcXqR3p/HcoKP/o1Cc4uOLnqxdJaMc2fYda6Kw1SLdsiJI0CJd7+HmirDeFWGaRFneP
eEUtWvblGfhcNHb3Kcm7DfclyZtZ5+cZ6XFlwccrKUhnmyc8mSD2b+dOuPqIPadg+NYEAEdwml
Oypzd5WJohF0eCyGTdQvbXWpB8HlbpqIwa+cb7mhrZ1bcNSeRIsZcfGa8HjPcxKXt2Y+dD+UmF
M+dWqRoT+PdmiO4SoudumM6hGE3qICNKvbMOyl8CEbemula5abwiprpw5cCelgloYcUqhanfw
cnKdcWWBc6WZm1KjdNeV+T8addaSnynqdk+QPRGSduGJhdE5bDi8Kb8cbV+24a0Fbn6xoJsIb
5usvokLcL6n9HcHceeJR2S7cx2ew1lAdFKaez6fdVKWjimgdceTNRGddmCKxc/Aa8/F1L3BbPG/1q
vJbgm53ZnubyS0QogTcEWuv3YwcW6pWmQEccakl1Fqc92fMz4udN2asylcdU6VxBGndfOL1c5T
a4TPpbyHbJnJGqq+babCiZkBbr68JYdCb9212HV8cQevqWNrcj+pslDsc3mkEz3OdHmezikjdOiX5
RGvdZeL9dlrfEdu58AzfHhwSa47fKVxrJwzfM5zBloafP90VHfhfT11m2VWfYx24FJPfex4HT6QfIF5
SSkLfr96Nw/6f3R6/9CieIV4M76seqI4uqzOevl5Q5ree0N504jie5R6ZHbRe+969mRqfFI7j1GDfNF8
Kz3kfUZ8uyh9fbB9JA+bfIz9kM71eKsBRb0zeRWA+6txeYGAAtJmgeeKAI4fHekyAZHXaesCAQWO
Ue0OALIDMe9CAKT1LfFOAISf/fLiADA9JfVx/5s1Ud0uKRrutd86JLqoGeEqIHJhYeLqHOYajeTKG
X3TcebWFjmK+ekmEzlAfeumEHjzEe3iDdieQe9mC7A8AfIOB78u7di6TU7o0dreRcqitdzmPmJchd
7WN+YWOeDmMaHPreMmK52HzeWuJd097ehmlHjxGerCG2iczewiF5g7Ze7WD0MpldT2caLjMd
c+Zw6dUdlqXJZXmdt2UzoR1d2qShnL3eASQUmEleK+ONk7VeWOMOjvMefqKZCbrekKJFQ7seu
SFo8jddG2llLd9dQWiMaYhdZSe05TOdh6bu4N7drKYtnled1WVyWBueAiS/E5BeMGQXDtgeViN9
CaueZKMVg8Bei2HPcdwc8Suy7YqdF2qpqTpdO2mhZOzdXqisoJ/dhKe9HFEdrqbUV+2d3KX1k2

ueDCUITrZeMaRnyZwePSPcQ8SeY+InMYHc0C3/7TKc9mzD6OWdGiuKZKDdPSpqoFydYylQnB
cdjOg9V7zduyc1k0Qd62Y+zp/eEGVfyYseGeSJg8heQqJxMSZctrBR7Nzc267qaJZc/i2FJFmdIGw
54B3dRer0G+Edb2m1V4+dnaiD0x/dzidmDoUd8yZmSXtd+yUfA8ueJqKQ8M1cozKrbI8cxfEi6FNc
5e+apB3dB+4f3+kdLOyqW7OdVes8F2ldhGneEwEdtSiYTM6d2edZiW4d4KWbg85eDyKS8YGgvh
uP7T+gqVvpqP3glJxDZLhggVyaYHBgb5zw3CNgYJ1HF7+gVt2dkzqgUx3zDoVgU95DyVTgYR6D
w2Jgd57UsS1gRB3CbPIgN13pKLagKp4QJHZgH144YDSgFV5h2+8gDV6NF5FgCl67UxCgDJ7rT
mAgEV8YCTVgHZ85A01gMZ9uMN2f25/k7Kcf1Z/bqHdfzx/S5DefyN/O3/0fxJ/NG77fwp/OF2bfv/
UUurfzN/fTj8f1R/oyRlf39/swzsf9B/0sIBfiGIFbFAfhuHMqCBfhCGUo+7fgWFlN7xfgCE5W4ZfgWER
VzffiCDuksVfk+DQzilfnuC1CQDfqKCewytfvqBp8CFfRKQtK/lfRGPEJ9HfQuNcl6ffQuMBH30fRCK
p209fSCJYFwmfUelL0qDfYKHGDgaFbSGFiOwfdCFYAyMfi2DW78kfDOZba6NfDiXDp38fDiUs41xf
D6SlnznfEuQi2xUfGKOmltjJCMxUnofM+LFTeqfP2JiiNyfQSIhwykfVeFC73He3KiQa1Ke3qfKpzTe
32cFoxke4mZQHv3e5yWgmuFe7qT4Fq1e+2RY0ldfC+PGDdHfFqNBiM9fE2Lsgy9fJuGhbxoetGrI
qwCetynTZujeuCjfYtPeu6f9HsBewSchGqxeYWZNloEe1uWFKjRe5+TMzbye8eQoCMFe6qOhwzSe
/mHysTelKz/aqtel+vZppUemSq14ofem+mqXnzeoOil2nJeqKeqVIEetia70g5exyXfjZ2e0GUciLHex
mRAgzke3Glt7m2ee286qlhefm3rJkaefqyelkCegOtonj3ehOo6Wjwei+kWFiQemSgAkeseqmcAjYR
esuYciKNepqTJAz0ev2lwrheeaDF86gxeaXAOZgQeaC6fogSeaa08nghebOvhWg5ec2qQlf4egGIR
kc1ekWgtTW8emWb6yJdei6U7A0Aep2lyrnwifFtu6nXiRBvKJnAiDBwllmXh2lx83lrhqpzVWk0hfp0u
liihVt2HUeHhNZ3fDWghHJ4yyGdhG553QtRhA97tbjYiBh2EKjTh1h2xJjRhpp3dYi9he94lniphUx42
GiKhLR5mlgLhC16ZUb+g717NTUkg2d7/CEwg158nAslgtv93bflhoV+lafYhd5+l5fqhTd+JYfwhKB+
Lnf0hBJ+QWftg5B+YleBgx5+lkaCgr9+2jS3gnV/HSDPgmmh/VArlggp/wLaZhUWGJaa6hK6Fepbgh
BWEz4cBg4qENnchgweDqmc2gpCDMVbogyCyUYHgd2CdDRXgZyCKiB6gYqCAwqPgTeBZbV
OhD6OU6WMg66M8JXQgxuLj4YMgpiKUHZHgh6Jl2Z5gbCIDVZKgVeHCUWLgRKGHTP5gNaFS
yAzgLaE1Ap2gGuC8bQBg2uWqKRJguCUk5SZgIKSglTxgdWQmnVLgWGOyGWfgPiNEVWUgK
OLdET8gGCJ+jOVgBylqx/+f+OH9gqSf5GEgLK3grafGaMYgi2cVJOCgaGZkIPygSmW+nRogLmU
fWTagFaSIFTxgASP5kR8f8GN3TM7f3eMFB/RfyWK6gquftCF37Fqgh6nmaHggZekHpJfgQ2gpILo
gJadZHN6gCiaQGQOf8aXQVRIf3WUbKp4fzKR2jLffuKPnR+gfnyNgQrHfiuHDLAmgaOwGqCZgS
Cr45EZgJens4G9gB2j1HJvf6ygFmMpf0ecglOLfvWZIUNmfrCWCzJ6flqTZR9pfeaPxgrcfZ+HWK7e
gT64r59agL2z14/mgDOvB4Cjf7aqiXF0f0GmLmJSftiiAFLafoOeD0LefjyaeDlbfekXLx80fWKRuwruF
SmHZK2dgOvBX545gGm8D47if962vX+1f12xk3CffuWsjWGcfnmnuFJFfiGjL0JrfdmfFzHMfXuaXB
8lfPKTXAr8fMaHbq4NkRntUZ7Yj7Zuso+njlwFIBqjRFxenExi9Jy5GH2iqN0VIJfiYV1xUI5ilR3LTE8
h7Z4iB3uh315rwlQhhF8Ha0Sj111R535jhl1+l7jjNV2qX+9i6J3ZHCainp4K2FxiWJ5AlHpiFI530HPh
2l6wTDahqZ7nB2Thmp8WgkRhQB+Dawrjd987J0pjK18944si3t9A38cilt9JXANiUh9VGD3iEJ9k1F
+h0t94UFuhmp+PzCEhbF+nh1ChXF+/QjahBF/wKs6jKeEfJxHi4OD541ail6DVH5iiUiC329riD6Ce
WBrh0OCJ1EJhImB40EQhYeBrjA5hNaBiRz7hJGBmAiRg0GBOqoji6iMN5tJioqK9ox2iWqJuX2WiF
ylqW63h1mHrF/PhmaGx1CHhYWF8ECqhLuFLi/shA2EjBzAg7mEWAIYgnSCoaj7itqUJpokicOSPo
tXiKqQWnyQh6COq23MhqKNEI8FhbOLik/ghNOKMUAPhAml7C+Ug1CH2RyXgtyHYAi5gZWEEaf
RiiucMpkNiRiZoopTiAGXFnuihvyUwWz4hgGSh15OhRWQb09lhDeOdz+0g2qMsC9FgqeLMRxzg
haKEQjagNCFVaaeiZikTZfpilmhElk/h3Sd2Xqkhm+a4mwVhXWYCI2KhImVWk6og6qS1T85gtuQjy
7yghCOqhXmGwaMbgj3gCeGEKVtiR6saJaxiBWofYgEhwSknHmAfh2hDmsOhP+dpVyqhA+aaU3
xgyyXYT6vgImUpy6VgYaSZRwbGmQOggkPf5mGIKQ1iLu0mpV+h7WwG4bZhqarpXhqhZungGoV
hJmjglvWg6Of01Egr2cKz4sgeWY/i49gQuV4BvsgEGQTAKjfyCGLqMFiGy86pRoh2a3+oXdhIWzC
neAhUeuQmlChEGppVsig0elQEyygl2hKD2+gYKdey3zgKOYxxvFf82Rykg0fruGOaJ+mFZtLJQzllJ

uooXulFZWf3edkopxfGIQkM1y6Vr+jyV0YUxIjZt11jz3jD53RizDiyx4rxoTivJ58gdPiEp9B6Gylr90y5
N7INB1mYVLkud2ZXcOkS13KmjUj4N3/VqTje5440vpjHV5zjyiyV6wCx4ihd7shnQidR8nAcrhy5+x
KDulVp8HJLMk3p8TYSyKaB8fXaJj/d8smhhj1189VovjNh9SkuTi2x9rxXiih+lyw2iR9+oxmUiNF/Pwc
LhjaASqAZIDqDT5IlkmOC7IP+kl+CiXXrjuyCL2fXjVmB5lm5i92Bs0syin2BjDwLiUWBdyv8iEKBfhlg
h+mB2wbwhV6BnZ8xk0WKqJExkXSJpYM7j6Wlo3U8jgaHuGc+jHmG4Vk1iwOGJErGiauFczu4iH
qE2ivBh3eEbkh5hwiEkwbuhlqC3J5BkniSIJA8kLGQilJDjuqO8HRZjU+Nb2Zzi8WMBliHilGKu0o3iP
qJhTtNh8alcyt9hraHohkmhh+HQQcfg6WEHp1GkciZvI9OkAiXh4FkjkeVUXOMjK+TNWW8iyeRNV
fqibWPWEm3iF6NmzrshyaMECtAhgqK4BkXhU+JmgdOgtqE8Jw5kTWWhb45Nj3mek4BwjbybuHKrj
CSZBGTyp2WcVc+iSqUB0krh9GRxzqDhpWPYir9hW6OQRkAhJWLqgd1giuFC5sokLupIY0zjwWI
oX9RjUuiJ3Ggi7Ge52QDiiWbzFZyiK6Y4EiFh1GWJzoihhCTwCqthN+RvBjeg/GNeweXgZmFIpoM
kFSw8lwajqOs4n4+jOqo2XChi02IC2Meib2hZIWviECd9kfoht6awzmShZmX9iphhGCUyhi8g2CPD
Qe0gRyFNZjzj/2454sbjk+0Yn1WjJiv3G/KivireGJciWSnP1ULh+OjQUdjhn2fkTkvhTScGSohg/SXV
xifguaQXAflgLsFRZcln9FtH4mrnT5uknxWmrNwA27ymFhxamGRlhBy3FQpk+J0XUZRkeF12zfS
kCF3VyhhsR40BZAjpJ6MwWKij991ZZtnld0aokdm9N1PXvWmVZ2D26Elwl24WEzINB3wIPYkrJ4
uEYLkL15tDeVjwV6uCgulah7xBYyJWN83wV9iRt/ZZXRnQI7a4iSmpF7qXtdmB976G4cld58MWDa
k7J8ilOMkaF8+EXLj7d9dDdfjgd+AigBjKp+pxX0jFJ/hwVyiBuAxJUem/iCR4fvmYWB+3rKlxWBsG2
clNiBdGBrkrGBSIMukKeBOEV/jsiBMTcmjSKBPifai8iBcxXUi16ClgVohzyB9ZRImwuJP4dBmJyIYH
opljCHg20Mk/aGvl/tkdKGEFLDj86FfEUjpfSE8zbnjFSEhSeyivaEUxXCinGEkwV3hmKDEZOnmj+
QSIz5I9qO43lclXaNfmXOkz6MLV9EkR2K9llzjqJ3kS1jUCI2jaUi5uH/CeDii6HcRXHiXeG7AW1hX
aD3JLamY+XeIW1lZGVf3iilNKThWujkp6RpF6rkH6P4FGxjnyOQURNjKGMwDZliviLdSdaiXyKlxX
OiJil+gXvhKaEA5H2mPqewoTalqKcKXfPIEaZk2rfkhKXJV38j/KU2FEcje6StkPWjBKQvTXximOPC
icoiNqN2hXKh9OKyAYhg/OEJZEhmlCmC4Pfliui23bNk9GfsWnvkZqcv10kj3eZ9FBmjW6XWkNDi
42U8jWFidmS4CbmiEWQ7RW3hyOMYQZLg12EQZAHmBatdILhlcWpvnXUk2ymD2kHkTOimVxU
jwyfTk+1jP2cOkK1ixeZYzUdiV6W9Salh7+TnRWihomNwgZvgt2EWo8GI7q1CoH4IW2w3HUBkxas
s2hDkNuoslukjrCk308gjJ2hSkI9iOeAzTFiPSanSZvh06V2hWQhgeOIAaNgNKEbousp41tLn9BpH
VuiXLfoWFv52Zonm9xUVnxm5Jyyk1vmNV0VUBzllV13jLAICx3ZiQMkoR48RJ+kkd6gAP6i/p+i4tF
pip0J37goxt063KFoBB1tGYcnSh2k1mxmld3hE07I6d4jEBIITB5mjKekwp6siP0kVt72xJ0kQB9NgQ
Cis1/84rWpO962n59oeh7E3lvnuN7UWXTnAR7r1I0mT18H00llph8p0AflC19PTJ+kgZ95yPfkFJ+sh
Jqj9t/4AQJicaBL4pMo+KBZX3/oOCBHHG+nd2A2mVxmv+Au1khmDuAsUzDIzYAwD/qkzaA2zJbk
RyBDSPMj2eBdBjJitaCPQQPiOGCQonBovuH/316n/qHL3FBnPiGZ2UBmhuFyljAl1qFRExliMCE2
j+rkl+EfDIxkEiEOyO4jo2ERxJnjdIEZwQsiAKC0okvojOOoXzenziNWHCenDqMFGRqmWCK81g5lq
CJ7UwAIAaJCT9SkaSIOTH2j4mHkiOhjbqHURKFjMyGdgR2hxGDBliPoYKVaHxEnoyTk3ALm5O
RxGPhmLuQHVe/lf2OIUuZk2ONND8CkQCL8THAJt+K6COLjP+KYhKhi96IQgS6hjdMofVoOucS
XuOnfmZ3m9amwKXe2M9mCuVUlcrIW2TS0sdtkGRcz6gkGuPwjF8jkSOXCnqjFWNVrKwiwqJ2
AT1hYWDWocloG6jJ3qznX6gM252moedSmJol66aplZtlO2YK0p8kk2V5D4gj+KTzTEgjbSSFCM2i
7eQBRktik2LQAUmhOuDe4YmoAKqKHnSnRSmuW2XmhyjVmGYlOKgOFWylH2dR0nekdeakD2ij
2eYFjDFjTOV4CMCiyuSYBKniaeMdwVRhGmDmlU/n6SxVnkBnLitdGzbmckPnGDpluWI/1UVIB2ik
0lakXOfZz04jv6chTB5jMWZEyLWirSUVhKhiRuMmAV0g/yDsIByr2ptenT3q6Bu0WmDp+VwK13pp
HVxjVJKoR1zAUabnel0jTpsmvx2Gi1/mHd3qx+Clo95RA72IZl7PwKUjYd/LoBArg90GHTEqlB05GI
Rppx1tF2+ozF2llon+J3jUaDnLh4njpemdB5ui18I0d65x+MIVF8Kg8MIDp99gKujFOAc4AFrNt6enS
LqSJ6xWkcpXN7FV2TohB7fVIGnsh7+0Zpm6d8IDpOmMR9QS14jl+Cx+VIDZ+/Q8fkwCAXQLGi0
aBkH+oq9SAtnQ1qCCAimjOpHOAZF1PoQ6AWVHNnceAZUY8mquAjToxI8yAyC1ulUaBIR+ckz2

BvA8vkeuCdALailuB7n9EqvKG93PVpz+GUmhzo5OFslz9oC2FMVGGnOeEyK YDmc2EgzoKlvGE
TS1blGuEPB+gklaEiQ9KkN6EWwMDiXeCCX7dqi6NPHNngoCMJGf0otaLEvyHn3GKFIEdnCyJO
EWsmRKlgDnJljOH4S02k6SHdh+fkXeHhg98j8WGIgNXilGCQn5iqYCTmnLppdeSAWeAojCQbV
wbns2O9IC+m4mNoUVemG2MdjmoIYyLcC0UkvWkrh+dkLGKVQ+qjsuHsAOlh6iCd33KqOuaD3J
UpUWX7GbvoaGV0FuUnj2T4VBFmviSGET5I9qQgDIAIPWPFizhklaOAx+Mj/+M2Q/Jje2JDgPohu
6CpH0dqG6ghnGZpMqd5WYqoSWbT1rdncCY8E+hmneWu0Rwl1SUvjjUIGmS+CyVkcKRIR9mj1
yPlw/VjSaKQwQghlGCynxXqAGnlnDUpF+kD2VooLqhB1opnVOeN08AmgablkPolt2ZNDhok+2XFi
xJkT+U2B8+jsuRlw/bjHeKuQRQhcyC6nuKp6Gt7nAcpAKqaWTFoGCm71mRnPejq054maagnEN
2lnid0zgOk4ObICwJkM+Xjx8bjlGSzg/gi+SKvAR3hV2DBXV2t39t3Wrcsw1vLmBlrrFwgVWFqsJx2
Uq4pu5zRz/Vo0F00DRsn+V2Yig+nQF3/hrwmtx5pQvXmGZ8SgFajuJ/vXV6tiN0Imrasbx09GBBrW
h1yVV+qXp2rEq1pax3pj/aogd4vjR+nqx56ihfm717LRsimYB8jgwJlvR+tgGFjaiA43V3tOx6NWrPslx
6kGAwrD5671VyyqFZ7X0qvp176D/boPB8kTSKnZR9WCh6mp1+RRtOmEx/ZQw1laqA1gGrjJWBI
HVMs+aAKGqkr4uAF2AHqz6ACIVOp1OAEeqSo4qALz/Jn+6AcDSlnJWAzCiMmZyBUhtzlz6CKA
xblleCwHLi6aBN3USswOGFmpqrquFmF/Oql+FHVUapnGETkpnogeEbD+pnwyERjR4m7KEOiiS
mLWEXxuSlkSE9gyKk3CEWQH/ir2BWXTWskCMAGocreyLFI9yqaGKMFTFpbOJWUoboemloT9t
nk2IFDRQmu6HqSiEl+WHgBuolVWhtQzNkk2F4QJdicOBmXR/sZWR+WnGrUSQmF8eqPuPO1R
2pQ6N8knWoUOMzT81naWL2DQqmkKLECh2ly2KnRu7IIGKGQ0KkUuHNgKziOeB03QHsQCYB
WIRrLKWJV6sqGqUTVQMpHySIUI4oK+RBz7onQ+PrzPxmaaOjyhWloiN1xu8k8WMOw03kGWIY
gL8iCmCBXN4sIKeGGi2rDWbxl4Jp+2ZfVN1o/yXX0jyoCyVbz54nlWTuzOamReSSSgblfCQ+Bukk
xqOKA1Qj5eJAAM6h4qCL3LPsBOKU2gQq8ehlf1mp3+e4VLfo42cXUhsn7iaCz4lnAyX/jNBmJiW
NCfdlWmTtBuHkoSP1Q1ijuGJDANvhwKCUnlcr7Kqvmdyq2mnjlzdpYOkbFJgoy+hf0f7n1aezD2qm
6WcZDL3mCuZoSeqIPeV/BtvkgWROw1wjkmJFgObhpGCCGpNv8lugWBkuydvnVaLtopww0yusg
ByEkLCrZRzfDi4qVV1By4hpXR2oiK6oiF4ThYdn7h6CQj4muB9RQA6kCGAJ2qSvmp0dGCCuc11
Gla4tTN1y0zSsKp2sELlrEJ3sTjhqAI41C5ZpCN6FCMGoLx7dhZ/ni58/wlFmV5/ZwB0juKATmq/vTV
6OGC/uJh6clbRs/16t0zrr3l7MUMAqxZ7yTkDpuN8hy6Hovl9bSNHn4F+hhbVnNV/4QmlmAeBRAC
mjcqAcGrUvCh/5WDKt45/vVbUsvR/oUzsrnx/uUMEqgh/8DkOpdaATy6joemA0yN6nmmBjxcfm6e
CrAnClitqC5QDSjNaAjmrWuzmFh2DctqSE+lBESgyEekzdrYGEL0L4qRyEBjkMpOmEBy6woPiEL
COenW+EkhdmpOFpGoDlBmEWAEQi+mAuGrFunmLEWCfteGKIVaSSUWJPkytrLqljLQqFOIAjj
vpB6HpS6koCiHdyOqnl+HnBePmYyHkQpWll+FpAF2iuyA/WqUuc6QpGButTaPRVZisJeN9EyBrA
yM3UKpp6SL7jjTo22Lni6Xn3GKtyOzm8qKoBe6mKWJlwqh4eGxgHUigyBPG09uTuWSmAbtJ+
UdVYTr/6SsUw4q3GRN0JqpwiP6jihos6O2i52nsyODiOomxmNfBfOl9mLYgrakpuHVwlkiUuBcmnK
uL+b+1+itB6Zu1WWr3eXkEvFquiVuUIFpnuUFTThNojyStC42njSRoiN/mnmQExfHlySM+wsAkcaHc
AJniKmBoGk/uFGH0F8cs62fK1UXrwGcnktRqnCaa0Gepf+YcTf4obqWwy3zna6VCiNSmeuSVhe4l
oWOXgsbkQyHggKgiCCBxmiqt++nzF6ds0qkvVSsrp2hyEruqqfOUFHpZac6zewoU2aui27nTyX4
SMtmXSUOBERlf+PhgszkG+HkgLQh62B59OrcgBqXMGWcvBsla+Ac99t5p1bdNFvmlsidcVxP3jJdr
1y2mYqd8R0ZIMYeNZ13j9OedZ3QSm4ep14XxCZe755EdGdb6J0I7+tcMJ1AK27ceF13Zu7cwJ2t
4msdCB3kneEdTx4cGUvdmN5SFlyd456Fj6VeKB61SkgeWR7aBAxep17ts++bZl9pb3xbuF9o6wic
CZ9pJpFcWV9vohdcqJ923Zic919+2QedSB+HVFmdmN+QD3wd4V+XSiZeEV+bA/XeZd+Ws4Na
+yHBLxlbVCGJ6qwbrCFT5jzcAaEqYcvcVmEC3VccqiDeGNBdACC6FCwdVaCXT1fdoWB1igid0C
BaQ+leLeAssx5an+QVLrla/OOqqISbWKNBpe0bsyLnoYTcdGKRHRncZOI+2Jxcv6Ht1AFdGSGfz
zadZyFWCe+dk+Eew9cd+iC1sr1aUiZj7lwasqXH6fvbEaUtJZ2bb6SkIT6bzOQfXNzcKSOFGjch2
Mik9dc5CKrjxbdM6l8idzdXWHtA9sdx2E38l2aDui3rgSacWfrKaza0mcfpVbbM2Zl4QAbk+Ww3Kbb

8+UBmDscVWRYE7lctKO3jvsdBWMjyczdLKK+w9/dmyGpsf2Z1ysNLa0aOmoPaV2anGkS5RBa/2
gpYMJbYidFHHGbxKZnmA3cKKWR040ciiTljt9c2+QQibzdAOOTA+PddSILcaAZqi1frVNaDiwu6Q
kacKsApMWa0+nrYIEbN2jbnDlbmyfSI95cAKbTU2YcY+XjDslctiUlyauc2iRQA+cdVSJdsUHZhq+1
bPzZ6a5YqLqaSyZ+JlBarmu8lETbEip/3AVbdqIKF7Kb3Wggk0JcQecJTqcclKYNiZuct6Tyw+odOe
KjcObZazlQbK5Zy/CR6HhaKu8SpEYaji2f4Bla8iwym9mbV6rM142bvyl2kyQcJKg3TpBceCcPiY5c
mmV6g+xdlyKnMfBeGVp2LaXeNprmaVveU1tXJQ/ecJvGIL9ej9wy3Geeshyc1/xe2N0Ek3KfBB1p
DrifLI3HSYKfVF4SA4XfmR5X8YpdiZzB7Ucdsdz86QRd2Z04JL3eAJ11YHTeKR2ynCceU53wF8T
egl4t00Kes55qzo/e4l6jSWBfBx7NQ28fUV7/cSadDJ7+LOtdPh8FKLBdbI8NZHHdnJ8doDFdzF8uG
+yd/l8/F5JeM59S0xeeap9ojmvenN97yUHewJ+HQ1sfEI+ZsMJcpmE0bl3c3eEJaFodE+DfpCSdR6
DCX+0dfKCm27Gds2CNV2Dd7WB20u9eKSBjDkweXmBPyScegGA/Q0me26Af8GAcT2NtLDNci
mMQqAecxCK149pc/GJq36udNalim3kdcGHeFzFdrqGc0sjd7iFgDi5eJWEnSRDeROD9g0CeqCC
b8AUcBqWma9tcROUaJ7NcgWSP445cvKQXn2gc+OOjGz7dNyMy1wBdeGLIEqEduiJkDhFd8ell
SQBeDOHJQ0YedCET76rbx+fk64icB+cp52gcRiZwo0qcg+XJnywcwuUm2wsdBCSJ1tSdSCP0En
4di6Nozfedw+LqCPJd2yKZQ0veRqF870+bkuolazSb0+k6pxvcE2hRowXcUmd9Hu+ckyatWtbc1m
XkFqjdHGUKUlrdYeRyjd3dmmPSSOPdrnNfQ1EeH2HXLvabZ6xjat4bqatHZsib6WoulrucKGkvXq5
caeg2Wp7credD1noc9SZc0jVdO+WGjcJddKTGyNQdhqQMQ1Vd/mljLpybRK6kKonbhi1dpnrbsS
wZ4nZcBCrwnnGcRWnM2mrcieiwFk7c0iehUhKdGiamjakdUyXlyMWdY2ShQ1jd4mJDbkVbKTD0
6j2baO+CZjkbpi4b4jvb5OzEHj6cJmtx2j8cayon1ioctCjvEfVc/SfPTZOdNqa4yLldRWUdg1vdyJFb
wlfudpW6vbfuBrGpuUftps2YtEfuFuk3rufuxwTGqHfwByAlnNfylztkiQf2p1YDalf7Z28CJngCB4JwvV
gMZ50rrbfLZyA6qtfNly+5qCfPtz9lpKfSh09XoOfVd1/GnHfYp3C1kjfdF4lUf1fit5OTX9fol6PCHufu56
+guEf618SLmietF6aKmOexV6opl8e1d634lee6B7NXk/e+t7lGkUfDp7/FiGfJt8d0dpfQt8/jWBfXd9eS
GCfdZ9xws+frd+cbhDeUuCvahFeaWCPphLef2BxIhOeleBcnhPerSBK2hEexSA81fZe4iAzkbffAqA
uDUVfIGAoyEjfNmAiwsBfeCAUrbUeAOLLqb2eGmJ85cfeM2lvodBeTaHwHdieaGG0md3ehCF91c
uepKFMEZWeyKEfzSue6CD3SDVe+uDbArjfROCErV5dvCTvqWld2KRzJXbd9GP4lYdeESONXZ
geLqMnWaZeTSLHFZ1ecCJtkXDeIWlctREetKHSyCcewaGkwr+fD+DzbQfdgOcZaRndnyZv5S5d
vGXHoUYd22Uv3V5d+ySdmXSeHCQSIXPeQSOQUU/eZ+MZZPmehmKwCBsejiJvQsbe4SFUrLC
dTqlFaMjdbhehtpOPdi+eXYQLdrCbTnSLdzaYWWUHd8CVhVUleFuS3kS5ePqQcjOHeXOOTyA5e
YCMkgszeuOGn7FwdJStvaHVdRSpn5JJdY6ljlLkdhCh23OGdpaeR2QndyKa2FRtd8CXnUQpeGS
UqTMgeNySFSABeN2PDQtHelyHoLAbdAq2b6CRdlqxrPEXdQOs+IHODYSoonKQdggka2NWdpa
gXVPAdzacjUOid92ZEzLBeFWWBh/LeE2RMAteYeemHrK7Sc5i/Mp9sdBa59ZAVdly0ulDkdQyvsH
HBdZGqymKmdh6mEFMvdsCho0Mwd2udojJxd+OZdh+fd9KS+AtneYmHtbCZha1o+6EshSBqtpH
ChJhscoJPhC1uK3Lgg8dv52Nsg2lXqFOlgx5zZkNYgu11Gzl0gtl2tx7DgxV3+wnlgbV6WK+Bg5JxL
qAugy9yM5Dfgs5zOIGFgoR0QXluj11VGLTgf12dVMdgc13nkLbgbJ4yTHCgax54R5bgeR6tQmB
gd18jK5+gb15EZ9AgXp5apAlgTI5xYDIgQl6L3GKgN16pWJFgLZ7KVKhgJ17v0JrgJV8YjFdgJt8+h
3+gM19aAlEgOt+eq1kgEaA4J41gBeAlI8Of+mASX/sf8iAF3DKf6t/82Gff5R/4FIXf41/3kH8f5Z/7TEF
f6d//R2tf9GAEAkOgBiAKKwmfwel5p0UfuaH444KfsOG5H8Ffq+GEW//fp+FT2DxfpaEpFGGfpyEC
0GJfrGDhzCufseDFR1qfuKC3Aj3f02BvargfgCRFJvXfeqPZYzafdKNuX3xfceMQW8LfcCK3mAffcC
JIVDXfc2lZ0EBfeaHWjBOffeGbx07ffWF/AkXfnSDV6mYfR2ZXZqpfQ+XAYvHfP2UqXz5fPmShW4
vfPqQel9hfQKOKFA6fRaMyUCGfTKLMS/6fT2J0R0TfSGI8gk3fbSEv6hLfF+htJI0fFSepYqpfEWbm
nvzfEaYzm1HfEyWIF6bfFiTI0+XfHCRPUAIfI6PIC+jfJWNURzofGOLiwlSfRCF9acKe8GqB5gwe7q
mQllne66ignrMe66fGmw/e7Sb1F27e8CYuk7ge9qV1z98e/mTPy9Ee/2RDBY2e7qN0wlqflaGXaXG
ezuyZ5b1ezeuAog3eyypnm0eyqloWtGey6hwVznezeE040e1SapT75e3SXky7re3aUzhyGeyaPy

gl9fBGGa6SMesm61pXYese1+oc2eryxHXjlermsb2p0eryn6Vw1esajmU2jeuGfmz6LewOceC6gew
KX9hxeeqiRbQmOe6+GdqUpjJdoqZaXi5lqVlgLip9sAXI2ibptu2rsiNpvfVxqiAJxSU2Uh1zEj40hpN
01C3vhhp2fRskhjF30AftHox64aQ8iqNwepXGicVxeldViOlyeHjaiB5zimprh1p0q1v2hp114E0the93
HT3WhVh4Wy2XhOt5iRrOhP16dAexg9t82aNjiOp375UDiCN4Tlarrh194rHhJhq15LGnshgF5uluLh
Vx6W0zPhMJ7DD2BhDx7yy1Jg9l8gBqBg+V9Dwd9gux+kqJ3h39/QpQrhsl/CoXohhR+1XehhW9+
w2lchNF+wlsRhDt+1Uxpg7J++D0ugz1/KS0FguR/Xxo+gul/oAdPgh2AE6FvhkmG1pM6haCF8oUQ
hPWWFFHbihFuEbGi2g8iD2FqCgz+DXEv0gsKC8TzQgliCmiy+ggOCVxolgfaCWAc/gVGBgKBXhUq
OpplfhK2NHoP1hA6LnXXhg3uKWGfRgvCJklm+gm6lG0tSgfeHJTxUgY6GTixrgTCFnxnkgQCFYg
dkgHSC+p8vhHGwj5ELg9uUZIL1g0GSPnT3graQVmcBgjGOilkLgbWM4kq/gUKLXTvkgnNqKCCw
ggHSl8BnGgCSIFwelf7GER535g7qeh4/mgymsYHjgpSY4nP7ggUWWmYggYqT81hLgRGRt0ojg
KCPqTtugDeN2yvTf8uMYBmkf1+KeAeofwqFLZzlgYgmfl6tgpei+YcmggafgXLvgX2cZWUYgPuZb
1dsglCWq0lvga6UHzrof6SR4it8fzCQERl5frCMkAfCfnyFP5uSgqKugo17ghyqbH95gY2mX3G8gQ
OirGQcgH2fJVaXgACb1kjEf4yYyzpqfyCWISSqfqiTiRIQfheOXgfZfgWFTppmgju2noxmgbWyFH56
gSatjHDPgJmpOWNHgBKIFVXjf5KhMEg0fx2doToAfrCahCrmfjOWbhkufZSP3wfrfaGFW5n6k7Boi
YxOk9g9qSn6qkHhsCXD7jxZtwmNVjb1vhVWxjHFxVkewi0RzJTkZikd07SmPiZN2oxdVib4DAYThx
h75Jk7kdtwBouhkFZxln4QjtpyPXB5jY5zWWLnjEx0hVVTixd1xkdeif53DzjOiQ94WylLiGF5nRcUiHt
6rQXwhf99ppiEkEJ3LYr9jtF3r32AjWh4M2//jC94xGKAiwF5ZFT7ieF6GUcRiNi63jiKh/Z7sSkPh058
gxbbh1t9RwXRhQh/MZewjvB+Joo+jYt+H3zXjCx+GG9uivx+IGIFidl+OISSiMZ+aUa7h85+pzhHhvp
+9Sjbhlh/Uhaqhlh/2AW2hDGAiZbTjcyFVllxjHGEShwbixqED27lifKDimF1iNiDGVQYh8+Cw0ZYhu
GCezf6hheCSSilhXaCNxaGhV6CiQW1g2CBzpXrjNOMp4iBi4aLdXsnijuKRG3oiRmJK2CuiAWIK1
NvhwKHS0XPhhiGgTeVhU2F2ShnhKGFZhZ6hF2FPQXpgn2DGJTiy/yUJleUirmSV3pJiXaQim0di
FqO2V/6h0uNRVLVhk2L1kVThWaKiTc5hJqJbSgvg+OlnBZxg3aHnAYagbWEIJPmi0abvIaVigqZR
3lWiM2W1Gw+h7SUjl8yhqeSallshauQcUTLhMWOpTbUg/eNGifxgzaL9RZggqmJsgZEgQqEPJLZ
iq+jToV8iXmgNng3iEGdJGsyhyeaUl5ChhmXqFFihRqVMUQphDKS8DZeg2GRACEogpiPbRZFgf
KLiQZngHmEVJHBii2q9IRmiP2nTnckh8ijr2oyhq2gUV1chZydIFCghJmaJ0OOg6+XcTXugtuVICdig
guSexYogVKNIAaFf/6EaJctib6ytoNoiJOumnY8h2Cqf2lZhkSmjVyahTCiy0/8hCufSEMMgz6cGjW
PgmmZNScngZOVCBYRgMqOcwaef5iEeY7pmwBofoljmNJqOnVnlq9r92iflMRtsVvdkuVveU8YkR
hxUkHrj3tzKTQYjiJ0+yVFjS52vhONjWl4QwRxiQN8yY5RmUhvp4GYlZNwyXTrlSZx6mg4k0xzElul
kYB0TE7Rj8d1nUGujjp29TPijOp4UiUWi/d5qhNmjBt65gRjh+J+XI29l8t2gYEulcZ3EnR4k8d3pWfY
kfx4SlS4kD95AE6Njpl5zEF0jRl6pjOxi9J7kCTtit98gRNFiux9hARXhuV/vo0Klo19J4Bzll99N3Pqkpd
9SWdfkNF9blrRjx19pU42jYJ99EEvjA9+TzN+itV+vCTJied/QhMnid2AFgRMhgiA8YxTIXWD9X/lk3
+DdnNMkY6C+mbTj82Cm1pYjh+CUU3Rjl2CIUDfiySCADNEifKB9SSliQCCFxMZiNeChwRahTGC
EouZlIOK2n8DkppJ3HKAkLkI3mYXjvaH9VmyjU6HJU1Fi8CGdUBwilqF2jL1iSWFYyR9iCWFMBM
lh8WE5wSahEqDHlrNk7CR7X5AkdGQXHHGj/GOzGVsjmNVFkajJWL+0zGiwuKx0ALiaeJszKuiH
CI0iRZh2GIThMyhtGG/ATWg32DRYnrkvYzh31lkSOW7XD0j0mUv2SqjZSSvFhti/KQ3Uw1imiPKj+
YiQSN0tJch8qMwyQshq+LjBM1hfll0gUJgs6DZ4kAkmagTHxtkJKdgm/0jrqaWwO6jQWYpVeWi2
GV4EuAidWTuD8JiG6RxDH2hzCQlyPxhguOohMphTaKcQU0gjqDhYgHkeKnkXtzkBOkQ278jj6g/
WLTjImd9lbGiuKbHErRiVGyfd5/h+eWHTGUhqaUJSO3hXmRVRMBhlyL2gVZgb2DnYcPkXCu9X
qPj6arMW4qjdSnc2IPjB6j41YXinaghEo+iOGdZj4Kh3SamjFBhjCXzCOHhPyTIRMQg/yM4wV3gVS
DsoPyooxoiHgUn91qLWxBnTVr12BgmrtdlFSBmElvYkiZlfrXrjw8k+BzJy8pkiZ1BSEGkPF22BAUk
MV4rgMBirV9koN/opBvXXesnINwc2vmm7txjmAXmUpyw1RGlut0DUhqIKd1bzwVkp922C8KkOp4
RyDvj695uBAKj2B7WQMFiY1+/IMPn4p163dJnPh2emuRmmp3EF/RmAN3yVQOlBF4lkg7k3x5ezv

wkX96bi7uj897cSDbjpB8hBABjh99+wMJilmAOYKEnlh8RnbLm8x8W2sgmUJ8d19xlt58uVO9IJJ9
D0f4kmZ9fzu9kHR9+S7Njs9+iCDKjZB/Ow/5jQGAVQMNh6eBTYHynUmCunZCmsSCTGqimD6B
5l8Bld2BrVNdK5WBikepkXGBhDuBj4eBii6mjeeBqSC5jKOCaW//i+yCfwMohsuCloFjnF+JPXWnme
GIWmn/l2KHfF5qlQWGWlLWksGGIkC6kKGFpTstjriFOy5vjRWE9yCoi72FBxAiis6EIQNzheCCVYD
Em5GP6HUNmRqOe2lrlqCNE13glEeL1lJdkgeKt0bWj+qJwDrhJgKI5y4+jFqIQyCYivGIEBBDic+Ga
AO4hRCChIAKmt+WsHRYmGyUrGi7lfaSsl08k6CQ9FHKkWGpWkZdj0SN7jqDjVyMqS3/i7CLqyB
9ijlLBxBXiOulBQPzhF2CrH9AmkmdnOAI9ma6WfZlWOYaVxpkw6WMIEOkM6UJEW/jq2STDohJ
MGQpS2pixCPViBQiYyNvhBciB6JdAQlg8eCzn5nmcWkWXKkl1ehUWb9lOOeV1ubko2bp1BVKEy
ZJUUKjiaW3zmOjDeU2C1VioGTHSAjiPOQHhBch2mKswRQg0eC6n2OmVGrXXHbluen5GZDIHa
kdlrukiChSE+5j9yeTUSijbObITkoi8GZLC0OigiWUx/9iHGSGBchtGLDwRzgtYDAnj/qjtouW4Rpt1q
YmMso5BsDVgmoJltxU0bnbZvkEH/mvFxczZlMhdzWioLlml1QRyRIQB3Hgzfk6x5jAG2jDI+R3jDq
KtvPG3XpV1wYGL1oh5xh1f6ny5yv0z5nFd0DkHnmaB1eTZxly528CoJISB4cxyak6t5+gzwkjR8Og
HMiwt/jHiDp1B1e22bpAt2lmK+oNN2zVfNnet3k0zWmx14bUHNmHN5YzZllgl6bCoFk/t7jByjkn8w
w0AkON+ngHfigKAqHgmpiz7km1Doul7xmJun7d7/1elnM58VUydmgN8wkGgl2B9STYrIP194in7kv
V+IRyqkW9/dQ0Oj7iAsgHviRuBT3e8pSOBsmzfoeuBb2lQnryBMlc1m9SBFkxWmQuBE0FnIm2BL
TYEIA+BWSnqkgiBpBywkHWCNw0ojpiCmQlViDuBaHdUpD+H3GxpoQ+HKMGQneWGfVa9mwC
F6UvsmDqFckESlZ2FHjXhKz6E5CnJkS+E1hy0j4SFLQ1djXGEZwJqh0yBonbZo3eOIWvxoE2M8
GEcnSiLxFZQmkaKt0uMl4KJykDFIOeJBzWPkoelZymrkHGIBBy4jq+lAg2OjGuF+wK4hniB1nZCos
qUf2tcn6SSxGCKnlKREIXJmaKPjksTlt+OMEBilEONAjVFkeCMACl9j8OLTxyuje6KjQ2wi4GHYQL
6hcOCA3WYojia3mqknxWYp1/lm/OWe1UUmRSUiUpylk6SwT/ck62RMDTdkUaP1Ck3jyOO2RyPj
T6M3w3Diq6IngMzhSqCKnTbobahX2nlInpeetV8Jm3ecGFRhmJeZt0nTlc+XhT9XkymVkTR2kL2T
3yJxjpOSlxxujKKO6A3OifeJVgNihKiCSnQboUSoBGk1nimk6l5nmw2h3FPLmC2fCUINIWOcaj7okri
aETQfkeiX4ii2jhhqU4RxSjB2Qmg3YiV6JXAOKhDyCZW46sido9GQwriBqoFovqjVsTVABpsJt/UXG
o2dvxDtwoDBxpjCUUnU9zkiTsmux1hRgRmVR3bQollhx6swCVjY5+5W47sJVvKWQqrKBwWloiqM
Fxi0/1pVJyxkW+ogB0GTtwntZ1izChm/l3EiUKmY94qxhBI+J6TQo0lJR9HAC5jFqACm4wrzp1HWQ
Yq0112VoLp3V2mE/jpAx3aEWzoMJ4TzttnaJ5VTCqmsl6dyUjmFI7uRhrIp9GApakzd/OADZi0yAk2
4GrhV69WPsqjJ7RlInfpmB7m0+8ovN8BEWTn6p8hTtYnl59JTCImbd93iU0l0d+uRiPIxl/zgp7kgKB
DwD1imGApW3KrRiAyGOWqTqArVmkpWuAmE+Hof2Al0VmnrOAsTs2m5qA7DCVmMWBPyU6lk
+BuxivlG6CkAqmkNuCugEliX2Axm2JrDeGmmNfqGGGGFIgPJeFmk8voSuFK0UZneOE2jr7msm
EsTBvl/CEpiUvIW+E1RjKk22FVwrrj66ESAGCilqBBW0yq3OMf2Mlp6OLh1jxo92Kk07eoHOJtUTS
nSul+TrEmhGlajBMlZWIBCUIIKmH6RjikoqHxAspjQWfPAHXh7SBP2y8qsqSd2KUpv6RAVh/ozqP
k050n9GOSUR0nlmNJjp5mW2MNTAWloyLeCUJk/eLGRjpkcCJ7wtYjbuG1wlhhvyBcGwwqjYc2
H7pnGWjVfdoq6UsU3en0STAkPvm/qRfjoMmNiQNC/DlfKPJiTtk1WORRjZkQqL6At2jOqHwAJeh
mGBmmuPqb2ek2FZpfacQlc8ojSZ/U1JnsqX6UNrm3yWBjmemFWUZC9vIWmTByScksSRDRjDk
GqNoAuLjDSHzgKThd2BvWrnqU2k22DBpYuiGVa0oc2fZ0zLnMkc7UL8mxGaqlCl+WYsS8olPW
WfiRtkkqTXhixj+KPDwuei5yH2wK+hXCB22NGukhpbFnwthJq51CtseZsa0dprdpGD4Oqexv3jSLp
ilxxCp4osxzuH+JoAV1vhNSnjN3ugdtmEJ7zAAAjgB/xmOBuLJvUlogtlhwXVDRsGRxcUeCrFxySD4
kqHV0CzSppL11iiqloV93Jh/Onoh43hOunJl6ogewlq596wAAjRyAAGOKt1J0+Vo4sy51mlDhrw92R
UeRqwp3lZ41pyp4GzTAo3t5NirLoBx6eCAKnTI75RP+my99dwfqlUd/xgAAjFOAAGO+tjl6mFpEshd
61FDgrfd7HEeNqe57mz40pg18NTTHomB88yrjnv591CA5nBJ+4hRFmfGANwgdIAyBZAAOi4yAC
mO7tTmAllo5sRx//FDOrP9/40d6qPSAAj4kpRGAPzTBoWWAoSrungCBJCBcmwmB2xSDmMyCz
ghYkuOC1wBliqWAMGOttFqFk1oWsECFEICZrCWEnEdlqByEVz35pDqEMzSjolyEOSrjnSGEaS

BrmhqE3RS6l7SFLQirkbeEKwCtia+AdWN7s5mLGlnkr4GKKFBoq2WJRUCYp16lmz3Qo3yIFjSHn
82HwirXnF2HoSB3mUiH2hTqlr6HPgj2kLCFUwEJiNaAs2MjsvaQtVmRrtyPS1AZqr6N9EbPpriM6T
2QotWMCDRWnyOLXSq4m66K7yBvml2KwhUFleWJFQkwj8eGNwFYiByA6WKysm6WTVkZrICUf
k+dqj6SxEZdpiaRXz0tokGQJzQFnomPLSp8mw6OeSBNI+SNaRUFIswKvAlZjveGUgGbh3+BFml
ssficBliVrdeZ1k8eqbCXvUXppaeV/jzHob6UczOzngKTLSo+moKR8CAIl0+PuhT9IH6MLAl3jkOGZ
wHUHvqBPGGfsZGh51garW6fSk6zqUacyUWlpTyasDxyoVCY0zNtnY+XJSoJmgqU1iAFItKRqRT
3k/CNYAmRjayGeAIDhoqBXMl4bhdIMrd5b2RnaKY7cKxpnZTyceFrslOWcxltt3lcdFdvr2BZdaJxkk
4ddvJzWTsaeCZ1ACYTePI2SA3NetV3M8bMa1lu7rWrbONwOKSMbmdxgZNob9Vyvol2cT5z/HDt
cqR1OI9YdBJ2bE1ldX53jDpwsN4ISWNd415Wg2leaR5+cUCaPt4TLQCarh4u6MFbG15LZIFbf95
sID8b416M2/ecRd6t15zqcV7N0yLdCp7sTnadX58HSUXdj58Zw1MeJI8u8NcZv2BhbJ8aOKBHKG
garyAuZDDbGuAgH/ebhSATm7pb7eAlI2lcV1/9Evjcvd/xDIVdFh/ICSvdQ1/bA0Xd7B/K8HNZUqKqr
ELZ0aJeKBOaTiLS4+TawCHVn7SbMGGbG4CbnuFj1zkcDWEs0tGceKD2zjdc02DEiRac/aCgw0C
dtyBYcBUY9aTtK+jZeaRvp77Z+qPzo5kacWOIX3Ja5qMg20gbWeK9VwmbzSJb0qscO6H+jhtcmG
GoiQgcV6Fug0ldhWDcL7fYpCcza5RZK6aF53MZsCXZo1YaKyU/HzfapKSomxXbHGQXVt9bk2OK
kolcBWMFjgLcY6KMiPvciGI/A1ldWeFO71pYXul5qz+Y6Kib5ydZbye/YxMZ7Sb2Hv2aaeYyGuQa5
SVzlrXbX2S8kmeb1CQQjepcM+N1SO8cVuMQg1mdNKGxrv/YJSu7aumYsKqrptaZOKmeosxZt+ir
HsAaNme8Wq+as2bUFonbMCX00kRbpyUkDdDcCCRoSOFcKmPKA2AdFSIErqVX9W3+qpZYgS
zDZouZCWuK4oqZiWprnodaCKIRmn8ah6g+VmFbBic20iQbfuZAZbkb4SVlyNScAuRqA2Vc+qJLL
k4XznBEqktYWS7mpkwY4K2I4IMZYSw6XldZ4arxWIYaYemwFj9a4eh+EgibXKdizaVbv+ZgSMnb
4WTvQ2nc5GJO72AdFZk8K0OdSxnE5ykdf1pN4w/dr1rTnvNd4RtXGtAeFZvXlpieTlxUUkCeihzLT
bTewp06CJ9e7t2OAuUfWV3tLvecb9uBquRcs9vVptJc9dwp4sAdMVx+nqvdbdzTWpLdrB0oFmRd
7R17khPeL13MDY+ea94WCIGelV5LgtYfD16bbpTb3921qoncLx3W5oBcfB35InbcwF4h3mtdBV5K
2ludTB50FjUdIJ6eEewd3J7ITW7eHJ7uSGceQt8Hgsjezl87bjPbZx/jKjAbvp/TJi5cE1/EYi3cXp/A3itc
qh++2iSc9l++VgcdRJ+/Ecbdkd/BDVHd1N/ByE/d+B/BAr1elh/G7dUa/iIRadkbW6HRJd+btIGSYeec
ByFiHe4cWCE0We/cqaEJ1dsc/KDhUaOdTmC7TTddk6CYSDzdsyCAQrleYeBGLXyapaQ96YQbB
+POpY6bZyNhYZ9bvCMFXa6cEWKsWbmcZyJXla3cvilGkX+dEqG7DR3dWOF2iDCdc+FLQsMe
LuC/LSQaWGZuKTPaviXQ5UablOU1oV7beSSsHXXb0iQmmYlcK+OmFYXchmMr0V/c3WK6jQd
dJGJVCCZdO2IZgszeAmEo7MtaFmifqOLafmfTZP2a4ycJIR2bPiZTXTzbmaWiWVib9mT3FV2cU6
RUkT/crSO+jPDc9OM5CBtdCOLcAtUd3CGDrHXZ3qrN6l/aSKnRJK5arqjX4NdbCqf5HP+bZ2cfm
SSbxZMITJcJSWEkR3cgKTLjNjcyWQoiA8c22OGQtwdu+HP7B/ZsCz9aD+aGqvWpGRagOqzYJ
Xa3Wmq3MabOuioGPQbmquesVQqb+6a+EP6cWOXizMLcoqUjSAOcsyQYwuldoGHZK8zZia8vJ/c
Z863m5CXaWayf4F6atmtqXJabFKo62MtbDwKTIOjb1+f9kOQcNyB/jLacgaYNR/ockOSTAubdiaH2
bJterVkoaLMexJmuJM0e21oIOne8Rq43QVfB9s9GR2fIBvAVR9fPVxBEP6fX1y9jKcflg0xB7wfp12
HgmTf7N4VLEReDNtLaGUeMZugZlfeVZv14KtedlxNHM6el1ylmO6euJz/IPae3d1aENpfBd2yjlefl
N4ER6HfTZ4+QlffpR636/Qdgd1caBxdsR2DpEad3x2rYHGeCF3Y3JveMV4H2MNeWt45FNFehx5t
ULmetN6ijGve3t7TB4qe/J7zAkwfZh9G658dDV9o581dRJ9jl/5del9eYDGDqZ9inGRd2F9pWJNeBx
9zVKmeOJ+AUJnea5+PjFNemJ+dB3Zest+IQklfL1/DK0fcqSF7J34c5iFHY7bdISEVH/JdVSDvnCz
diODNWGQdvGCvllJd8qCVUHreKWB+zDxeWGBqR2YebiBeQj9e+6A16vScVKOR5y0clmMxo2lc
1eLTH60dDaKDW/AdROI32DBdfCHx1FedtaGxEfId7uF3DCVeHaFDR1weLSEmwkoex6CIKqEcC
6WuJuDcUKUhYyQckuSWH28czaQam7ndCGOKWAHdQyM1FDFdf2LNUDuduqJvTBDd6Slcx1P
d8yHuglSemiEGqkybzSfMjpMcE+cR4t1cV+ZY3y9clOWym4Hc0eUSI9KdDyR61ApdTePs0B1diu
NsC/xdual8R0rdvyKgw12ecuFaKfqbmCnnZkLb4Gj+lo/cJagXnumcY2dJ20ScoWaDF56c3+XF09/d

IGUUT/zdXuRzS+YdjmPoh0AdkKM9AmUeUeGeqakbamwEZfVbs6rylkdb+Wni3qgcN6jtGwscdif/F
23ctWcbE7hc92ZGD95dN6WFS9GdZ+TehzXdZ6PDwmueNeGi6VrbQ64jpa/bjWzwYgob02u+Xn
DcEiqdWtpcUSmFF0TckSh4E5cc1Cd9z8TdFiadS8AdRuW1hy1dRKQ0QnDeHmGmqdXgVBkW5i
lgS5ma4nDgQ9oe3sJgPtqimxUgOlsm12bgNxur06MgOFwvT7vgPtyui5rgS10lBtggZ51+AfCgct5A6
YzfudsdZeEfvpt0Yjdfw1vLnpAfx9wj2umfzJx+10Hf0hzC04Lf2p07j58f5p2ZC4Ff9h3vhsHgD14uAeU
gLN7SKUofM50OpaSfQl08lgHfUJ1pnmGfXR2bWsGfad3Plx+fdt4HE2VfhZ5Bz4Uflp5+C2qfqN61R
q4fvt7cAdsf759RKQTew1755WPe2d7/lcYe718Eni1fAN8P2pSfEp8elvfJN8xE0WfON9Gz2ufTp9e
y1bfY191hpzfdHAdJful+/6LleYqDv5R8efqDJYYhemWCjnfber2CH2mVexWBwFtDe2+BdkyRe9
CBOT1GfDWBDC0OfI6A6Ro9fMSA6AdCfh2AnKG0eEWLvpNReMiKe4T/eUSJO3bXeailK2iweg6
HL1qAenWGTUvxeuGFgDzLe02Eziy9e6OEOBoge7iEAgdwfUmCNaB8dyyT15lyd7yR6lP7eEKP/
HXteLKORWfieSOMp1nReZeLKEthegyJyTxden6lkix0etCHixoHesiG6wedfl+DnZ89dj2b/ZEKdtSZ
WoLqd2KWvXT2d9qUYmcleFKSJvKZeM2QDkrMeUuOIdvrecGMaiwqehKK+RnrefKJegfEe++E0
p4GdXWkGo/UdhKgwoG4dqSddXPedX6agmYQd5mXslhleBeVDUojeJmSnDtteRSQcCvYeWSOn
xnGeTKLuQfke2iFVpzNdMisQo6ldWqoTICXdf6kYHLVdnig1GUmdvSdbLeCd3SaOkmFd/qXRDr3e
HqUoyuMeMmSSxmjelqNqggAevWFaZuddDO0dl2UdNmV/X+jdXCrinH2deqnWmRgdmejVFbcbui
fhUj/d3GcBTqUd/aY8itMeEWVYxmGd/qPSAgXepaFeJxTiA9kHY5Zh4JmGoBphvZoGXKEhm9qJ
WSrhelsOVbZhWZuV0ixhPRwcDn4hJtyeipHhG90YRfThM11zgYdg7V5r5tqhcxr1o2JhWptKH+zh
QdufXHqhKBv5WQqhDtxXFZqg9ly5UhRg4F0czmhgzt1/Sn5gxl3axeKg2t4dwX3gqN7s5qJg8tzL4z
Dg4tz5H8Jg0l0nXFcgV91d2Oygrh2XIYEgnJ3Vkf5gjJ4WzITgf15Zym0geR6YRdLgih7FgXVgbN9d5
mbgiN6Wovtgfl6gn5Ngcx6rHC7gZR6+GMqgV97U1WRgS57wUeagQF8OzkHgN58vyl4gNB9PhcT
gQR9qQW4gOJ/AZiegLCBvYsHgJmBQ31+gH6AznAGgFWAhmKOgC+AU1UNgA+Ankcuf/KAJzi
zf9yAJik7f9GAMBbqf+6AYQW2gBiAdJeWf3OJYIn7f3GIRnxzf2eHMm8Uf0qGUWG5fzCFIFRYfxq
E3UabfwaERjhEfvWDyij1fuSDbBbYftmDYQXpf0CB7JZ/fmKRllj1fm2PYXuAfnCNp244flyMKWD3f
kyKxIO0fkGJhkYVfjWIZzfgeHcSi3fg+GrRbKfeCGCwYZfoKDN5VafXyY7ofgfZCWgnp8fZqUHm1
KfY2SA2AkfYOQCIMBfX6OPEWGFxiMmDd3fWyLLSh1fU+KDBa2fQKIYwZDfd+EPJQ5fL2gtoa3f
NidnnlPfoiakmwzfNyX5l8fNSVYllxfM+TD0TgfMuQ8jb+fMKPHCgrfKCNpxaZfDyKdQZmfVaEU5M
OfBuojlWRfDuk3ngwfE6hPGspEkd/V49fDma61FpfDSYEKRDfDGVezaNfCmTPCflfAWREBZ9e5
GMPgaEfOGEZ5Hle5OwdlSEe7asSHc9e8qol2pHe72kSF10e7Ogn1DBe66dNkO+e6yaHzYte6eX
dyere4CT6BZlewKNvAacflCEeJGWjvhkCYR5jcxmGXdojKdoKGpii6hqMl1kiq5sR1Bnib1uaEMJiO
pwgzUKiEVyYkiYHh+l0gRQQiG11/QR3hc56xJDPjNlraYPHi9Ns2XbMitNuSGngie9vuVz4iRFxOVA
MiD1yzkK6h4F0ZjTFhup1+yXMhpN3eRPfhwF4ogRmhLV8j5ALivxyboMcihRzS3Y7iTB0KWlmiGN
1E1yTh552C0+3huN3FkYjht4LTSIhbJ5SiWYhV96XBO0hbZ7PQRYg8B+IY86iXF5QYJgiJt5m3W
Wh8d59Wjahwd6WVwdhIF6zk9Thad7V0lihRF76jRKhJh8hyVqhEt9JxOOht9zgRLgup/f45kiBOAQ
YGZh06AB3Tghol/zmg7hdV/qluUhSt/mU7ghJB/n0HFhAd/sDQFg5l/0SU9g02ABhN5g22AfARXgh
uAx42KhuKHaYC1hjKGpnP2hX+F4mdmhNWFLVrZhDSEkE5Dg6KEEUFIgyGDpDOqgrSDUiUKg
l6DKRN+gkyDKASXgT+CElychdmOwX/ThTiNYXMihJOMAGalg/GKuFovg1qJi020gtClgkDWglSHI
TNXgemG0yTdgYmGUBOEgUqFgATSgH2DMLudhPiWNX7fhGKUKnl6g8aSlmXRgyqQSfFlygpeO
j00WghKNAEBZgZqLmTL9gTCKbCSqgMeJIROCgGeHkQUFf9eDZYqbhDudoH3Tg62a8nEpgxiY
S2TTgn2V6FiQgeuTrExZgWWRoJ/DgO+PyjKTgISOiRtgBWM/RN1f6CJZQUwf0uDgomMg5qlHH
zlgXKh3XAjgoCep2Pfgabu1e2gVOY+0ujgMyWcz81gFWUKjluf+uSPCQyf3aQAhnmfvWK+gVUft
SDmoh8gxKsrnvSgo6o8W9Fgf6lOGMSgWShulb/gNcebksKgEibYj68f9GYpDHZf2iWNYQBfu+Sh
hNZfmaMTAVyfnCDr4bulhVkAnqylGBmDW6EkrNoGWJekS5qJFY8j7NsPEoUjkhuZD19jQ1whjA2

jBZynCHai4Z0IRCSi812TQMDh6h7uYZIIBhrC3oekoRsgW4DkPZt+GH0j4hvdFXmjiVxAknNjNNyp
T1Ci6p0SjAEir117CGzii13fRB8ik948gMFhop9UYWhkl1xwHmPkOFyrm2Mj2pznWGSjg50mlWVjL
91p0mKi4N2yD0Limt38i/XiYd5lyGRIpZ6UBBpiPR7kAMHhY9+t4TykOp4Q3jwj3x4tGz/jhB5J2EajL
55qFUxi3t6Okk2ik164TzJiUV7jy+oiG98RyFzh+F9CxBYh75+HwMIhLV/74Q+j55+5nhlj9+0GxkjOB
+vGCSi5Z+vVS8il1+0kjWiTt+/jx+iD5/Mi90h3F/diFWHuB/2RBXhpWAjQMig+KB EYOMj nqFo3eJjS2
FEmudi96EgV/dipyD+1QdiWuDjUhRiFGDPDwXh1qC/C8thoyC1iE3heyC5RBxhW6C6ANsgwOCL
4LHjXuMIHbLjDyLcGrpivqKS184ib+JO1OLiJaIR0fZh4OHdzu5ho+GwC7thcGGMyEchROF8xCLh
GuE9wOxgj2Cf4HsjJ+Tp3X5i2uR32ofijGQG15+iPyOg1Lmh9eNDEdQhseLwDtOhdaKmC6khQaJ
qiD5hE2JIBCag4eGygPsgZSCp4EKi+WasnULirmYUmkqiYSV+l2aiFGT41lbhyuR8kanhhuQMzrK
hSmOoy5lhFiNWyDjg5aMLhCcgr2IaAQegQWCylAZi0ah0HQbih6e6Wg+iO2cDFy+h7qZdFFXhp
SXCEYDhYGU1TpLhl6S3C3wg7yRQCCagvKO2xCbgg+J0ARlgyC5X8mir2pB3M/iZqlpWd3iGuiS
lwEhziFk1CxxhhKcPkV5hP2ZkTnghAqXLi2mgzeU2iBygmiRFxCZgX6K/QRrgCeC/XxWnW5kB3EA
mzll/WW5mQpn91p0lvZqBk8ulPJsJEPZkwZuVTgHkVtwgCtykA5ynx22j0x0ow16jq123wG7iUI8kH
vSm5FqvHCMmXZsKGVXI19tmVollV5vl07vk25wwEOnkZdyczfgj/t0KCtVjrN12h2mjel3fg19jR95hg
HOiCZ9/XtNmfnxJHAcl+tyEmT8leVzBVnak/F0F06ykhF1OkN3kE12cze7jr53tCs7jXx4+x2YjKx6RA
1/i7l8lwHehyh/PHrDmJF3Xm+clpV31WSlIjH4Ull5kqp48U5hkNN5okM1jxt6aDeJjZp7NCscjGJ8CR
2Mi5J88w2Bin1+ewHshkqAUnovl1V9q28SIWR9pGQJk3F9pVkkKkYl9zU4Dj7p+CULnjgx+XTdQjJ
R+tyr5i2N/Hx2Biot/sg2RiVWAogIQhXSBUnmflj+EDG53IFyDk2NknODH1h0kJOcyU1/jsuCi0J8jS
OCazb/i6+CWCrHinuCyr13iY6Cqg2/iC6CswJkhJGBnnkAlUmKmG3ek3KJIWLUKZOImFfsj7qHw
U0ljfiHB0lbjFWGcTa3iuSF8iqbiayFnR1uiK6Fow3phyyEggKyg8iB0nhLIHSRQ20tkqWPqGlokM2O
GFdNjvmMw0x7jTqLkEGni5qKiDZfiiqJoCpiiO+l8x1ch+Glkw4HhkeGHQL0gxyB/3elk7+X6mxekfS
Vx2FRkB+TsVaEjk2R5UvHjl+QQUETiu2OzjXsiXuNhioWiD2MiB06hySLRg4XhXyHjAMsgoyCJXa
2kyGeqWuLkVmcC2B/j4aZe1XAJbaXOEsZi/aVIUCCilKTQzV9iN6RmynLh56QPh0XhnuNpQ4hhM
ylywNbghGCRXXjkpilgWrJkNSibl/OjwSfaFUbJtScqEqGi3SaG0Alic2XzjUfiFiVxymMhxaTbRz5hey
Pnw4qhDqJkwODgaqCYHGzpOZkKGdOogdmJFz2nzholIKHnK5qLEgPmjlsRz1/l+NueDJnldpwpY
Z/IEFyzRlek1t01wqgkSF31wCVisB9UnFkox9qjGcGoFVsCFy2nZltiFJTmx1vF0fmmLdwuT1ilNyd
TJWIHV0NSZ9ktx1+BliskeZ3rQq7j4h6fwC2iZl+lnEOoZFwqma8ntdxsFx3nClyu1lgmbZz2Ue/l151C
z1ElSp2VDJFkzV3qSZ6kZx5CRI5kjl6bgrSjiF84ADTiJZ/s3CvoD12oWZgnY93OFwhmup301HXm
Hp4h0eClid5Tj0Vkt6LDlnkhB7EyZwkH18CBmEj3N9FwrnjOh+8wDth7WAoHBBnw18oGX4nGx8
w1u/mc987VGBI2J9Mkc5IRR9jTzbkvB+ADIAkQt+fSZgj3p/DBmPjmF/zgsl8SA1wEbhtyAwG/Vng
GCqWV/m22CYls9mNuCH1EllnWB8EbPICyB2TyHkgyB4THFkCeB+SZDjo+CMhmajVuCtwtFiqC
CogF4hfWA/m9YnRel1GUFmo2IDVrlmAWHTFCblaSGpUZvk2CGGzw7kUOfTjGPj16FayYojb6F
URmljHKFhgt9iaCENAHNhSqBOG7EnEyPGWRzmcqNyFo5l0iMf1AWIOuLZEX6kqqKazvckl2JnT
FJjqal8yYAJQGLiRmii6CIEAuoiL6FlwIWhHyBaW4fm6GVXmPBmSOTkVl+lqKR0E9nIEiQR0Vfkga
O5jtdj+aNtjDojfyMsyXCjFCMARmMit+KZQvDh/WG1gJTg+qBkm1omw2bv2MjMjGZgFjGhKXT0
68k7iVWETHkXWTjTrfj1GR/DCIjWSQoyWDi7OPSxl0ijSMbwvXh0aH6gKHg22BtmytmoyiPGJemB
Sfi1grlZec6E4skz6agURHkPmYSzp1jtKWVjA4jOKUkyVPiy2SCRlgiaOOlwnhrWIDAKzgwWB02c
xrKFkSF22qRxmSIRGpbBoTUqroqxqUkD9n8NsajcsnQFumizKmpdwziGKmLRy/RT4l7B1DAgbkz
R5EgAAi1B+RWchquFqYV2hp3Br7VQtpBVtekqVoRpvDUDtnj1wtTcmm4tyeCzSmSd0RyGmlzx2
HRUrlh534whMkZ57eAAAinN/R2cCqVlwoI2BpfZxV1QMoqdyd0p7n7RzokDbnOJ04Tcdmj12OSz
XI953pSG9le15lxVYILI6pQh2kDp9kgAAibKAAGbRqA119F1NpLd2qVPXoXF3YkpMnn54KkC1m6
95BjcCmQ95+yzOlrv7ACHMIMN8GRV/k399UAicjwR/ZgAOiPOACWaOpuV7q10Jo5x79lOVof98

RkoQnW18qUCCmqF9ITbdmAV9tCy8la1+VSHSk7d/DxWiki2ABQjMjeKBDwBFiBeAL2ZlPd6BYF
y0oqOBSVM0n3KBNUm1nlaBK0Azmb6BOjahlySBaiyXIMmBryHKksaCHBXFKUiCzAkWjMKCnA
Cphy6AcmXvpPuHLlxbocuGoVLcnqKGGEIjm7uFoD/rmPaFRjZqll2FEix0k/+E/SHCKfKFIRXjkFOF
PALai8WD9gEFhmGAsGV7pDqNE1vnoRGMB1Jqne2LA0j4mwmKID+OmEWJXzYglauyyxAk0ql
YCGpkTKIPxXxj3mHbAmPiuWFKQFThbCA5WTyo5mS+VtSoHKRfIHmNVCQDUhmmmm2OxT8MI
6iNpTW2lQqMuCvxkqOL/CF6kIKLbxXpjrSJagm0ih+GOgGWhRyBEmRVow6Y/Vq0n+qXGFEvnMi
VQEfvmeWTIT6Mlx6SFjVNIHuQ0iuikg+PyCFlj+aOPRXcjgaLKAAnQiXKGogHOHJ6BOGOvopqfIFo
gn3aczVCqnFWaiEdbmXKYdz4glqiWmTT0lAGU/StfkZGTRCEej2KQkxXQjXSMnQnniOOGsgH9h
DWBWFX+tJpkm1O9sOFmb0sQrTBoTEJcqZRqUDmJphtsajCFotRunybin/Zw3RxPnbtzGhCPnCI1
XwXKIRt6QQAai1/gVysstxqaFPfrzJr0Esmq41tQkJqp/lu3jmWpIlwkDCZoVByYScInnJ0RByQnCZ
2MxDkmn94OAYPk4h8XAAaiGAAfZFsVVv/VPtrbVw/0sqqhlyC0JupoxzRzmeoyZ0mTCpn/d2Byc
onRp3kBzImr95MhEumQV6+wZLkid+NQAaiW6AAfZKsBN1fIPnrHt2HksbqON2ykJepVV3qzmRof
B4ojCnnsR5tic5m+Z63xzzmYJ8lxFvI7p9qQaAkPR/0QAAiNOAAfY4rvB67IPPq2J7Lkr+p9J7fEJBp
ER8ADl5oOF8nDCanbZ9Vic/mtV+lx0UmGZ/EBGqloiAQQA9j9WBQwAAiF+AAfYlreuATVOnqmq
AM0rFpuSAJklJo1yARjIKn/uAgjB6nNCA4iczmemBxB0kl2mCBxHjIWAcpQcVjriClgAAiCqAAfXzr
Q2FwINzqZSFOUqPphSEvkHWopGEezkenzOEzVzBcnAeEXCcnmRuEhh0ylouE9BIVIGeEuwdjib2
DvgBNh36ANFwdrFWLS1MfqN6KS0o+pV+JXkGKod+lujcnoGIOjAqm1SH6CcImGKHxh0slcWH4
BI0k4OGMAeijOCEwwChhsyAbVuvq7+Q0FKoqEaPbknBpMWOH0EXoUWNITh4neWMSy/ZmrO
LqibNI7uLPx0NIROKkxI6kreISQfPjByFSADnhjaAnFsqq0OWcVImp8WUsUIDpECTB0CkoL+Rrzg
SnV2QhS+GmiaPliaQlyiOvhzqlHWM8RI6kgOJwwfzi3KFYAEjhbaAxVqaqt2cMVGpp1qaCUjYo8+
X+kBDoE2WSTe9nOmUzC9Bma6ThyZdlquRsBzMk/CO6hI5kWyK/ggRiuSFdAFVhUyA573gafpg
Ha12a5Rit50QbShIT4yfbqhnxHwdcCxqLGuAcbsdsg1qUc0xuvkkndN9w0zbgdkVyvSJVdxV0JgsZe
gV1d7wFZtFpwau9aLRrc5t4apFtl4subFRuxHrXbhJwZWppb8tyBFmqcYhzkUhmczh1AzZldK52VC
HmdWp3RAR6eMd4X7pJZBRy/KokZjVz2JoEaE10tonhaj1onmzbCV2jWlwbgl3dVjZb+I4U0e5cbR5
ljXCczl53CGEc+B6XArfd7F7PbinYbl8DaimZA18FpirZIR8JlitaGh8WXilanN8kmiKbHV80FgZbnF9
BEcfcFR9LzVMceZ9UyEucnh9awrHdr99xbccX7OFBac9YieESpdmZI6DlleMZsKDEXepaOuCmG
e0awmCKFdnbR6Bs0aQbxaBPDTicLOAziDrcTKAiQrKdeaAC7WtXfmN3qXjYlaMY5YjYwaK74Zv
ZVOJuXazZ5WikGbjacuHdFa6a/WGWkYHbf6FSzSCb6WEViDFcBiDwAsBdSOCHrRFXHGwvqS
fXxKUiJUEYaaSWIVyZAmQaXXYZmOOimYraK6MvFYiauyK/EWOObQSJVtQvbrOH2iCmbx2G/gS
2dHmD7LLeWxufmqNcXcycp5PIYG+Zu4R2YuOXGHT/ZU6Uh2V0Z6ySC1WLafqPp0UYbCCNaz
PdbdeLbiCEbjuKNwtjc+eFeLGEWfqoYKIUXLWkrJKzX2GhBINsYd+du3QaZFWahWSyZr+XZITsa
RuUaESca02RnzOlBQyPJyBebW+NEguJc2yGxrAsWQOxJaDZW8OsyJGYXnSodoJ1YPmkg3NH
Y3egpGP/Zeyc31RaaFKZR0Qqao+V8TM5bFeTAiA6bLqPhwuqcwSH4q7IWDK565++WvS1ApCo
XaewHIGIYDKrcnKVYrim3mNpZTaiaFPeZ6aeLkPKae2aSzL4a7yWziAcBb+RkwwFcq2H9bNfCBN
gEaOacT1ikJP2cmFIEYRac3NnhHSzdIxp7WT2da9sSIThdt9ukERAeBVwtjK9eS9ysB7Weep0Hgk
nfH12KLGWbRtpDKIVbopqvJKcb/FsbYMicTpulHOhcoVv0mQNc9VxglQddSlzJkObdnI0tIT5d6F2l
R50eEZ3HwkMe0x4/bAKaoNxvaCubClyp5FabcVzIYIHbzh0nHKrcKx1omM8ciN2p1Nsc5l3qUMld
QB4ojHEdjV5gx4ddsR6GQj0ekB7IK6OaEd6UZ9Tahh6epAha916qoDzbXJ7A3G8bwZ7YWJwCJI7
w1LDcip8l0KBc6p8fzFddOx80h3RdWN9CAjgeVh91a0eZISC4J4FaESCTo72aiiBw3/ra9uBbXDXb
YqBIWgtbzeA31licOCAnklBcnSAYDEAc8CAKB2YdCGACwjmeIR/36vFZKiLXJy9ZrSKFY3BaLGI
1X7aanqH1W/qbECG4mDIbgOF+1F+b7+FHkGCcWKEUDCrcrODmB17cwGDNAkgd7yBxapvYz
GT5JulZVCR6YyvZ2GP9H3naT6ORW8XaxiMpWA0bO+LFVDtbr2JmkERcG2IOjBgccSHBh1jcgC

GZwIXdw6DbakaYeucbZpSZBmZuouZZjeXEHZxaCOUtW5CagySa1+Ba/SQNIBbbdGOIECgb42
MNDAVcOmKhh1JcRqJYwmGdniE2KfUYNWk5ZkYYwyhelpuZTOeGHvqZyabHW1faRmYNI7Ba
wyVZ0+/bPWSwEApbr2QTy/GcCCOMB0ocEuMAQmtdfuGCqaPX+atXpfpYiKpTYIYZE6ISXr0Zke
hq2yLaEGeJF4PajyauE8vbDCXfj+7bgOUiy99b22SAB0Jb5qOQwnPdZCGoqVWXxq12JbXYVqxP
ohsY4isq3olZYaoX2vYZ4akK115aYmgGU62a4acSD9fbWOY0i8/btSVkBzvbWwWJQnrdTaGtaiydk
5f7JnNdWriWYryd7lkyXwmeGdnNW1WeRIpnl56ec5sAk89epVuVT9qe2dwjC6qfDNylhtdfNx0CQd
mfrd29adEc3RoXJiHdGtqC4nUdV9rvHspdkFtdWx7dyFvM13AeAFw9U6feOpysT7iedR0Xi46eqt15
BsJez527wdPfZF5laX2cPVwhpdcciBxgYjLc0Ryf3pBdE9zIWu0dVd0sV0Wdl110k4Pd2R2+D5oeG
V4Gi3YeUZ5IBq+ecF5ywc7fl9746Slbst4mpYpcCB45oe6cW15N3IScpZ5q2rlc7t6KVxndNx6sE1/df
17PD32dxZ7yC2BeAR8Rxp8eGd8nAcpe6995KNZbOqAvJT8bl6AVoarb8h/9nhkcQx/x2oYckp/pVu
5c4J/kUzwdLt/hD2GdeZ/fs0xd1/ehpMdyZ/hQcyet9/uKlba1KI4pPIbN+H0IWGbmCGxXdgb7iF8W
k1cQqFL1r6cleEf0xVc6GD3j0QdNmDTyzmddKC1Bo5df6CowduehSBdaDBae6RGpKla4yPWor/b
R6NoXZ2boiMJGhqb+2Ku1pQcUuJbEvLcqWINDync+iHHCyjdOOGLRoqdPuFtweneWKC+p+YaL
iZV5F9amKW5YN0bAGUenWHbXeSVmeabuqQTFmicFeOYEs+cb6Mljw9cwyK+ixgdAqJmxoXdBi
ldAfXeMmER55eZ62hhJBMaWGeXoJQawebQ3SCblSYhma3bf6V5Vjjb3WTAeqIcOaRFjvLcj+O/i
wXc0KNNxn9c1GK3AgAeEiFYJ0oZsapt08naH+l8IE/aiqiNXOOa6ye3GXibSybo1gxbqmYkkoXcCS
VtztgcYiTJivTcpcQ9hncqam8Agjd9uFgZwCZgCx644iZ72tnYBYaWupVnK/avKIV2UubHehe1ebbf
qdzEmfb36aZDsHcOuXWiuacfiUPxnNcheOrAhAd4CFIJ4KfMBfwY/ufQBil4HdfUJkh3PdfYxm7GX
gfdZpU1fgfiFru0mDfnpuFzqMfuNwWiqbf1tybhfgf/hz5wXPgX3y5zbegFnw47heoBpdYDyev1rKHM
Re3Rs42Uye+luqFdNfF5wd0kGfNdyQzogfVZ0ASpBfdd1mheaf12sQW7f6B6H5vCd5lvcl3meEdw
f4AXePNxj3JWeZBys2Sueitz4FbHesR1F0iUe1p2VTnAe+x3kinxfHd4sRdbfON5cAWqfqd8Kpqsd
Yd3AYzodl13cn8ydy135XGQd+V4c2PreJt5DVY4eU15tEgfe16YjljeqR7ESmsezl7shcle418JAWbf
c998ZmNc7Z+r4vidKt+eX5HdZl+R3DCdmh+O2M6dzR+QFWkd/5+VkemeMJ+dDkGeXt+mSlrehZ
+whcAelf+9AWmfQN/lZhucl2GflrlczqFo303dD6Ezm/PdR+EJmJndf6DkVTydtmDFEcXd62Cpzibe
HCCTSkqeQqCBxb3eSyCBwXkfDSBLJdJcNaOYYm7cfSM33xCcwilyW71c/mKF2GodOil5VRTd
dSH0UaXdraG1jg8d4KF/SjweBuFTBbyeCyE5AYee36CkpYcb66WUoikcNeUH3tCcfar8m4PcvO
QCWDfc+6OPIOqdOaMlkYQddSLEzfZdqmJvyi0d0GIqRbnd02HaAZReuODxJT1brOeNod/b+SbV
HojcQqYfG0KcgyWAF/6cw2TPlRdAyRdUV5dQKPczdrdeCNrihydngMOhbTdouJoAZ7el+EYpPKb
dqmlIzhbxCio3kTcDqfMmwTcT+clF8hckSZNVI3c0iWeUTrdEWT9zcEdSuRwyg0dcmP0ha/deWLi
wafefCEepKnR+uElvdbllqD3gub4emFmtCcl+iZV5qcZee31Gfcp+bkERzc6OYijatdJGV5yf/dTKS
2haudVqNJQa9eZOEjpN0g1Vfl4YjgzFh5Hjgew5kNGuqgutmlF6BgsZo+1Fcgp9ra0PcgoVt0TXAgo
BwlSaXgqJyQRRgg0FzvQRegpV4l5KGGltNooVPgMto4HglgNpqiGsPgN5sR13/gOFuFFDwgOJv
80OAgOZx0DVwgPRzoiZTgR91TRQqgaV2bwRPgYF6p5GZfmxuflSBfqVviHd3ft1wmWp/fwRxf2
JfytzDFCMf090W0Msf291szUof453CSYYf8B4QxP7gCx5FQRBgl98dpCnfHt1i4OrfNJ2DHa/fSV2k
mnjfwR3OI0GfaJ38IAefeB4u0LTfhh5ijTjft6XCXkfoZ7HxPSfuF7rwQ2f7x+CY+yesF8xYlNezN8rH
X4e558mWk3e/F8s1xzfEV830+jfJh9IUJwfON9ajSYfSV9uyWyfWR+DxO5fbF+aQREfvN/gY65eU
CENYHRecyDhHT/ek+C2WhXerKCXluwexaB+U8Ae3qBsEHue9KBdzQ5fB2BUiV8fFaBQBO9fI2
BYASFfiKA9o20d/KLxYDdel+KcXQdeSSJI2eMeZaId1r/egiHFU5tenuGPUF4et+FgDPjezCE5SVM
e2aEchPDe4+D/wTCfWmCPYyjdWTZH/bd3+RZXMQeB6PbmaveJuNwlo9eReMN03LeZSK1UD5
egGJmDOlelmIJSUZeouHvxPBerGGTwT2fMqDVYUwdeia+X7FdpqYU3IQd0CVumWrd8CTgFIXe
EGRbU0NeMKPi0BkeTWN2TMfeZOMZSTeecSLRRO1efGIWgUifEWDeYp6dR6imn2vddWfYXE
Cdn6cNGSzdWCZa1h7d4KWzKxWeAaUaT/WeH6SQTk8eOKQaCSmeROOnxOoeUyKIAVle9SD

kolbdHKqR3yudSumjHAedda22PhdlqfelfCdt6cS0u9d2SZXD9fd+GWuTJpeEuUfSR2eHuRahOce
MOLmgVne3WDp4kpigxfj3ysiU9h7nA+iJlkTWPiiAJmqleMh21pEEszht9rgT5whmxt5zEBhiNwOSJ
2hiFyXBDhhqN0AwLrhJl5u4hVh5tm1XvxhwxomG+dhoNqXGNchg5sllcdhZ1t+ErXhTFv3z4khNlxx
DDBhJ9zniJFhKB1VBDFhPx2sALug4J7j4eBhXNtvns8hQpu9G8GhKRwK2LfhExxbla3g/pyv0qDg6
10Hz3eg2x1hTCIg0F26ilag0R4NhCtg4l5UgLxgo19KYavg6V0dHqAg1R1Km5jgwV14GJVgr92olZ
EgoB3ckojgkp4VD2Sgh55OjBQggR6lSH0ggx6/xCYgkV76AL0gbd+jYXdggR7THm+gcx7dm2ygZ
N7oGG8gV573VXCgTF8Kkm3gQ18jT09gPJ89TASgOR9YiHQgO192hCTgRt+IAMOGot/2lUNgJa
CR3jlgHeB7WzVgFWBkmD1gC6BQ1UUgA+BCUknf/mA6TzMf+mA1i/Df+CA1iGqf9+A8hCqf/qBN
wNagBaBHIQtf1aJdHgQf0qlgGwMfzmHimBAfyCGq1R2fw2F5UikfwSFQDxmfv2Esi98fviERSGJfu
6EChDBfv2DhgOhf1uCOIM8fkGQwHcpfkSPIGsyfkCNhV95fjCMF1PlfiaKyUgUfiWJpDv2fiWlNy8vfi
SHyiFjfhOHPRPDfh+FkQPdfrqCnYJKfVeX/nYufWWVv2oxfWqTiV6MfV2RlIL2fVaPyUdmfViOKztvf
V2MuS7TfV+LhCEzfUqKIRDSfV6HYQQQfjKcV4FlfJCfSnUxfKScfmk7fK2Zu12pfKQXQVltfJ2U8kb
AfKCS2zrvfKiQ+i57fK6PaCEGfJaNkBDRfLiI9AQ7fb+C3IBDe+mmqHRHfACjWWhrfAugEVzpfAKd
CIGDe/6aM0Y0fAOXnDqDfA6V/Sy4xfBeTSSDfe/6QCxDrfC2KRARffV+C9X7pkPZfjnNlj7Rh6Ge4j
npkRFw2jWFmolC0jFFpC0Uni1BrgDkginxt6yxZietwQB5micRyaA3KiYt0gAGjhmJ6vn4yjqxmenKpj
ZJoRWcxjH5qEFvGi4Nr4lBZipJtxETdibBvuDjliPjxqCwriGtzjB5liEF1Tw3Fh+x3KQG3hUd8XH17jK
ptEHIOi7BuWGazirpvoltiddw+FAFiP9yXESYiDhz0TiuH451SSwChxJ2vh4uhuV4Hw3ChoF5yAHI
hE99xnxKivhzdnFuihN0RWYliS91FVrniFp180+ih5N24ERHht533jhwhkZ43SvXhd152x4Xha161w2/
hUR8VgHXg3d/AnwViXB58nDGiKB6QGWLh9B6kFpjhwh68k8yhk57ZUPrhal77TgphSB8dCuohL1
9AR4Chl19nw3KhB9+wQH9gqmAJXtmiBeAh3AMh1yAX2TLhp+AN1m0heSAFk6ahTeACKNvhJ2
AFzfKhByALStqg7qAVB3ug3uAoA3zgwCBewJSgdGBPNqohuSHUm9Vhj2GmWQehZCF31kXhO
CFN04QhD+EqUL+g7CEOjd0gzaD3isygtSDoR3dgoeDnw4ZggSDHAKhgRKBx3nZhdmOQW6Nh
UCM5GNdhKCLjFhmg/mKYE11g1+JUkJ/gtalazcTgmKHoSryggGHBR3FgamGuw41gSeE6QLlgG
+B9XkDhPWVJG2shGaTNGJ2g86RTFeRgyuPoky4gpOOHUHjgg2MxzacgZuLliqjgTuKoR2igNyJ
wQ5FgGWGhQMef+aCG3gdhDKcGGzKg6mZo2GZgxWXOFbDgnSVEEwCgd6TEkFNgVeRSjYp
gOePtCpXglQoah1/gCWMZw5Pf72H7ANOf3GCPHc0g46jlgV4gwigLGDdgnedQVYWgdealktogU
KYGkDPgLyV3DXJgE6T3ioXf/KR+B1hf4qOnQ5YfzKJFwN2fxCCV3StmBtfkGnyllxh2F9lIKJkJFSik
v9mhkn0kWto9D8xj+9rcTPijrVt4ifAjdhwPRpdjY9yawsNjBl1NQCBh/R7onQZIfvMj2lylFdn6l7dkrxp
sVRKkTNrkUmuj7ltgj76jlhvhTO6jTBxgiemjFlzchpVjAV1QwseioJ32wCkhtZ9EHOEIBVscWj2kpFtul
55kQ5vCVP4j5dwdElsjjNx7j7GjOpzejOUi9F1BieOiwF2jhpOiqR4BQstiR56dADChdx+UXL0knpyjW
hykQhzYl4Dj5V0PIOVjih1O0kajNB2Rj6Ei5Z3YzNkio14fCdyic5lBpliWl6rgs6h+d8ygDchQJ/aXJdk
Qp4tWflj6h5E12Cjkr5eVMljOF6Aki8i5V6nD44imd7SjMsiWp79CdSiK18oRpDiER9ZgtUhsZ+7gE
MhDGAaXHGj8Z+8GdHjnR+4lZgjR1+2FKRi8Z+5kg8ioV/CT3QiWF/RjLgiGp/hicmh6p/1hpDhyuAU
wuMhamA+QFqg1eA9XEpijqKFW2avjWCExFxOjBiEM1IMisyDw0fHiZSDbT1ziHmDNzKbh4eDDib
+hsSDAhpChjGDPavAhK+CwghAgpaBL3B7jaCL5mYEjGyKuVuoiy+JIVFyieulqkdAiLqH3j0Fh6S
HODJJhrWGqibMhfCGSBo5hU6GJAvog9OEWWQIKgfGBYW/EjMWSbGVBi5iQuVrcimGPE1C1iSG
Ns0aYh/KMeDx8ht2LaTHihe6KfCaLhSmJyRoiHul0wwDgxCFxgJlgWaBi277jAiZBmR5it6W2loXi
aqUvE/+iG2S5kX2hz+ROjv2himPwjF9hTuOdyZLhHWNbBoLg8CLLwwXgmiHBAJ9gPCBr24vi2af
sWPAikCdD1lxiQ6ae09jh9OYK0VthqaWCzuGhY+UJzEohKKSfiYVg92QIRn3gyGNJwwogdulDgK
pgl2BzGpfn15fo2CVnPhp9FbYmqVkr00Cml1mpUMcloxpETkVIK1rji55kxtuASL5kf5wYBYgkaJyJa
iJlX2RgAAiIF8xmn+nVZl62A/mw1nwFaOmNFpmUzEls5rf0LrlOJtdjyxlvgS5lkZVxiCL3kHZzhBY1
kAZ1XgiujMt46QAAh7F96mmWm4xr7l/mmVltUIZFizFuukyJIT1wM0K8k2RxuzjQka5zVi5RkDV09

CL0jxd2jxZljpZ4GgjPi2t7RwAAhvt+6mkvmgRxx1+DI+FyvlXnlcdzu0w6k9l0zUJ8kg17ziekF93li4yvjv
F4VCLrjdt5hhZYjU96vAjrijl9WAAAhlx/yWi7mKJ3pF8VlpB4LVWAlIR4vEvgkp95ZEIwkNZ6HTHjjzZ
66S4LjdF7siLdjLx8gBZojCF9agkUiR1/OQAzhZGAI2hGI2t9i16WIWt9r1T8k21910tmkZJ+DEHHj9N
+VjgQjjp+uC3SjNZ/ICLEi7l/mhZ8iv+ARwlaiASBAACZhLWAZ2fllliDmV4blGaDP1SFknaC6kr4kKW
CqkFnju6CgzfFjVqCey2di/iCgiKsitSCqhaOif+DDwmZhw6CjQD2g/KApmc4lWiJxF2Mk4KI4FP4kZy
IBUp2j9GHUKD1jh+Gvjdpjl+GTy1biy2F+SKJigOF0haTiRiFmAnLhjWD7gFFg0uA3GaalJyP7Vzhkr
2OkVNFkNyNP0nPjxWMIEBijWSLJDbxi9OKVS0Cim+JpyJTiUCJNxaJiEOH7QnvhXWFKwGlgr6B
CWXqk++WLVwxkhKUyVkwkDOSoUktjm6RFj/SjL6PsZ8iyqOgyyqicSNfildiJGMgBZ7h4eJ+goL
hM+GPwHBgkeBMGU1k12cgluOkYKaRVIDj6OYFUiljeCWGj9ZjC+UTTYZipqSuyxhiTKRWSHwh/
yPPRZvhueLrwojhESG2gHxgeOBUGAppudfqlDko99iB051oPBkYkVvmBmvTxNm+9pJzL/malroy
kQl79uGh4tlmFwfxHvlc5yuwZPkFV3kQAAiD1+EmAFpPBIqlclogFnk05QnydpfEVOOnKZrajw0mkdt
aDLxmBRveSkTljBxjR5HIMlzmBljB11igaHjsh59AAAh35/Cl/SozNrbFb1oFds604jnY9ua0Uqmxtv8j
wamMtxiTlilqhzMykTIMt05B5dk1x2lxJQkpt4Qwa5jW18DAAAhTz/4l+WobZxB1a1nupyIU3inC5zPk
Txmb90Zjvrl3R1njLBIVp26CkHk4N4NR5qkhV5hBJ4kUV64wbjD993wAAhkWAAF9JoFt2oFZona
N3VE2XmvZ4DEStml140juzlkl5qDKYIDV6kyjykmJ7fh5xkPB8cBKekAp9iwcZiyZ/hQAAhdyAAF77
nyp8OFYNnld8jk0xmex850RPI4x9QjtilU99sTJbkz9+OijNkWp+yx5rj+p/cRLHjuGATQdqihKBDwAA
hbWAAF6fniSB71Wwm4+B0EzWmQGBtUP6lqqBozsYIHOBqzljmeB0SiqkJCCCh5ljwSCaBLrjDq
Cvge0iR+CZgBFhSKAL14snUaHv1U9mrqHlUxjmDSGi0OPleOGEDq7k7CFsjHakaSFeih4j8qFXR
5QjjSFdhL/jO2E8QfviEqDlwCZhHqAaF2mnlqNjlSrmgSMhEvJl4OLg0MBITSKpDo9kwGJ6DFzkP
OJVigujxWI6R4njXSlpBMAjBaG9Agah42EqADhg+yAmF0Mm+6Td1QRmWqSB0sxluiQoUJ0lJuP
XznBkmiORDEOkFWNWYfijnKMnh37jMqLdxL8i1iluAg9huqFkgEdg3OAwVxrm3CZeVOAmOmXm
0qwlWVvykH+IBmUJDIZkeSSqjC5j8+RaSekjemQHR3WjDqN0hL4ireKMqhahmKFpQFQgw+A41
W4rsFf102aq3xiC0WNqEBkRz1ppRImpDUdohppEyyVn1ZrISNenQBuFRkdm1hwgw4JmZdzNQR
EkiR40QAAh/9/RFXTrM9ljU2rqaBnWEWUpndpLD1qo1xrJjUgoGxtMiyhnbhvUSN9m2JxdhlZmaVz
lg5Yl9Z2AASNkJ166wAAh0+AAFXfqxZrE02qp/hsekWKpN5t6z1koc9vhjUfnu1xMiyonEVy8SOWm
fB0vRmNmCJ2jQ6dlkd4tgTNj0V8wQAAhrWAAFXMqZ5wdE2Qpo5xfUVqo39ykj1EoHVz1jUFnZZ1
KiycmvN2kyOfmJ94ABm0lsl5dA7aOd7UwUFjhp+WwAAhjCAAFWnqEx1xk1opU12c0U/okt3LT0b
n0Z4GDTjnGx5FCylmcx6JC0gl3Z7OBnSIZZ8Vg8Tk6V96AVHjQV/ywAAhdGAAFWJpyJ7EU02p
DZ7Z0T/oUV7yDzdnkp8UDSum3Z87CxkmNh9oyORlnx+ZRnklh/QQ9NkneATwWji/WBHAAAhbK
AAFVUpiWAdEz+o0aAXETEoGCAUTylnW2AdTR+mp+AsixDmAKBDiOElaGBghnzk5uCIA+AkW
2CaAX1iwaCQgAAhZyAAFUCpVWF60yqonuFXkRvn5qE4TxWnK2EpTQ6meGEhiwOI0OEjCNkl
N2EsxnvksiFCg+ikH+ESwY5ijSDRgAAhYiAAFSZpKuLYUw1odOKckPxnVojIjvinAeJADPUMtuliy
u9lpmlQCMqlCylHRnUkgmHyA+vj6WGAwZriXqEMABGhQGAMFQVpCOQ8EuzoUePpUNynmO
Objttm3iNfzNumKqMtStqlgSMHCLuk5CLnBmzkWGKMQ+0juSHhQaTiNqEcGghliAW1ODo7eWI
Es1oNSU5UMHnemTTTsMmv6SBDMZmC6Q5iskiYWP/yK7kwuOmhmYkNKMMw+5jkKlXwa1iFS
EiQC7hCKAf7M2ZZpbHaOkZ2NeDJQXaSxg+4R8awJj03TSbN9mnmUObsRpVIT3cK5r60RTcoxu
UTLCdChwfx6kdN5yAwiveUtz6rFSYfdkl6HzZBRmrJKWZjJowIMjaFdqx3OlanlszWQQbJVu0VQjb
qxwukOlclKlyfzl9clR0GR5OcuD1LwiyeAJ28K+XXsptsabiYTBu9ZE2Y5NwPIHsZe9xIHKXaEVy6W
MtapN0OFNlbNN1dUMKbu12mzHHcKZ3ox4CcRR4Uwi0duJ56K3wXAR2l57qXqZ3EI/nYUN3j4D
HY854NXGaZk543GJXaMF5hFK1ayJ6HEJ+bVh6ozFfbx57Gx2/b2p7bQi2deh8haxnWZt/Y52JXG
V/H46uXyt+4X+1YeB+03CvZld+zGGQZxt+yllSaZt+vUH9a+l+qDEEbbt+IB2Qbfh+kwjPdQt+2asD
V42IB5w7WnaHC418XVqGFN6pYDCFWm/KYviEqGDSZauD/1F4aEWDU0GGaqeCqzC2blSCFh

1/bMqBzAkZdEuA8KmqVbSQr5sHWLeO/4xsW7eNVX28XqyL5W8AYZGKgmaRZGGJLFDwZxOH
30EdaYeGpDBya2+FkB1ya8aFBwldc6SCwKhUVBGZUJnTVyqW6otcWj+UinzQXUySa243YEqQ
W1+EYzKQXFBpZfmMcUC2aH6KqDAwanGJGB1iauGIOAmYcxWET6cKUquh1ZidVdCeulo8WP
GbpXvXXA6Y421jXx2WMI7TYheTIU/bZPGRFkBLZ4eOxi/saYaMwR1OahiLCQnJcp2Fn6XHUXO
qVJd0VKCmllkxV8ii3nrXWvKfd2yfXg+cIl4vYRmY5U9YZAKV0T/pZqiS+y+vaLOQhh06aWuNdgnc
jeGuqSXUGeyx5ZsU5mugYhQVseqPHouWfymI2v6XSailF2mYD6eOU7qYzaaiz+XZeqXLY97Z/+
UOR0paNqPfAoWceGG0qkRa41bNjPDbPBeDlt6bIRg5nyib8Njtm3EcTImfV7UcrppNk+GdEJr0j+f
dcZuRC7CdxtwfBs6d81yAQb9e650yKdWaCZkEZi4addmllOea4hoMHtybTpqRGy/bu1sV134cKFu
Z07OclJwYz8lc/FyQS5PdVNz8Rrvdd51EAb/enV3taXEZSZspJdQZxht8YjhaQdvRHpgauxwsGvUb
NFyG10zbrRzGk4qclt04D6CckN2Ky3pc693VRqtdBI4FAcCeWJ6YKRIYn91GpX9ZKd1qoe4Zst2Q
nldaNt3BWr2aud3y1x3bO54k02QbuZ5UT4lclp6Ay2PcjJ6pBpzco17DgcEeHN8sqLgYC59fJS6Yn
19WoacZMd9P3hlZv19WGohaSx9eFvFa1N9n0z+bWh9wT2Wb1R93S1AcNh9+BpMcTR+GAcEd
5x+yKGRXi6FwpN+YJ2E84V2YweEK3dkZVuDn2IFZ6iDHFSoaeyCpUxsbBqCLz0nbhiBwCz7b6S
BYhpFcA6BQgdptmAr6BKXGiOEZJYXvGMlIRxYXSLH3Z/Y+OJ6GiCZkylvlpSaKqHo0vqau+GID
zFbP6FmizAbpGExRpAbxCEbweudi6CV58HWtuWXpEyXXWUMYNqYAqSDXWZYo+QL2e9ZQ6
OYFnKZ4KmpUtnadqLATxka/qJgSyEbZaIORo4bjKHYGfPdZyDw53QWYSeIzAIXCmbulJRXseY5
3SjYVqWbWbrY+iUBlkBzm2RtUrdaNePhjv+aweNhyxGbK6L0xopbW+J+AgbdSGE9pydWFimyY7
qWwOjUIFMXaef4nO/YEacymYoYuGZx1h5ZXS3kpcZ+2Uljugai2RpywNa96PjBoabMiMMwhGd
LiFmJt5V1Wu9l3qWgWq94BvXK+m/3L+X1ijPmWDYf6lFfxZJ2cC0nwZySYvztRaXKvYcVday2TC
BoNbDyOEahpdGGFsJ7zcZpbNJDkcpdd94Lec5hgvnTRdKpjhWbBdb5mSFildtVpBUohd/ZrqTr5e
RtuKCrQeilwbBfTettx8gV1fdV1vJ12bmBjil+Sb6RIkYG1cOtnnnPQcjpptmXmc4Zr0lftdM5t8EmHdh
VwADp6d1Jx9ypveGhzvReUePI04wV4fKI4b5waa35rm45dbP5s84CoboBuUHL0cAFvx2UgcX1xR
FdHcvJyxEj9dFx0QTOKda91sCoads12+xddd1N3yQV7e6N6zJrGaPBzll0yaqR0QX+mbFV09Hllbf
51zmRib592sFancT3mUh7csB4fzmjdC95WynPdVp6lhcsdeV6owV9er5825mHZrZ7kywSal17k3
6namJ7m3ErbC571WOlbfB8G1YJb6V8bEf7cUx8vDIBctJ9CimMdAZ9VBcOdJ99kwWYee1+uJhW
ZMyDg4rtZsGC4H2SaLKCRXA2apmB32LTbHeBiFVbbkeBQ0dxcASBBDjbcZuAzilRctSAphcRc36
AsAXjeSaAdZcmYxmLgonYZSWKOHYZZy6l9W9baS6H6mIXaySG81TAbQ2GE0b1bt+FQziAclW
EiykeccOD9BcWcoODuWypeHiB+ZX0YZuTg4jBY7iRjnudZdGPoW58Z+SN9WFXae+MYFQha+y
K6UZ4bdGJjTglb4WlVijqcMmHVRcWcaiGcAZld+ODRpTKYFGbclehYniY0XqMZJuWPm2JZruT/
WCGaANKR11N0at6P0kXwbNON8zfEbpeMRIiyb+OK4RcNcOql0gaYd2WEXpOIXzOjXYaOYV+gJ
3mMY4ec/WymZbCaJF/CZ9KXalLTaeqU1kVya+6SczdpbcGQUyh9bxaOixcCcEaK4QbDdvqEpkK
TXj+rRYWZYG6njXi0Ypmj22vnZMqgYV8dZvWdCIJLaRiZ30UlaymW9TcdbQmUYihRbmeRxBb6b
76MmQbndqGEqpS1d+dbJYdoeHdd2XojeQ5gkGzaecFjTV+UenJmDFJKeyJoykSbe9IrdjZGfJlt/yb
gfVpwTBRkfh5x0QQRf8p2tJN9dMZjD4ZRdZtlFnkwdnVnlGwKd19pN17leEVrVIG4eSVtfkQiegFvn
TXietdXpSaTe59zehQyfFt0pgQVfqp5F5JTcf9qqIVMcwxsDnhQdBttd2tNdTRu+l5HdkhwhFE0d1Vy
F0O1eFJzqjWJeTx1MyZQegp2IBQHes93cAQYfa57LpE6b5RyHoRRcM1y6nd0cgdzumqSc0N0ql2
qdHp1pFCwdad2q0NKdsN3sDU2d8V4riYUeJ15IRPheXd6LAQbfNN9AJAmbWt5qoNXbsp51HaVc
Ch6A2nRcYN6XI0GctH6xlApdCJ7P0LddVl7uDTkdnB8MSXfd098oxPOeEB9AQQ2fAZ+qo8Ta5W
BQ4JMbQ6A0nWXboaAZ2jwb/uAKVxEcWl//k+Jcsx/6EJfdBh/2jSKdT1/2CWvdh1/4RPcdyCAEGSC
ez2AQI3+afSI84FPa4KH33SxbQ+G0WglbpqF9luWcB+FMk75cZmEiUHucviD9DQ5dCqDdyWfd
QyDGhPrdiWC5QTJeouBo4zlaleQrIBKaiWO73PCa8GNOWdPbV6Lw1rdbvSKak5icICJM0F4ce+l
GjPlcy+HKCVZdBOGaRPzdUqFYgUFefSC04vTZ02YUn84aPaV8nK1apyTnGZdbEKRIloLbeOPsk

21b3uN9kDycPiMYjOlckWLAiUmcy+J6hPxdl6HIQU4eXSDh4q7ZkGf+34sZ+6dCnG4aZeal2V5a0
SXjFIEbO2VGk0Sbo+S2EB0cBiQyTMwcXOO/iT2cmONcBPsc+yJfQVkeQeDpYmpZV2npH04Zwy
kNHDhaLagyWS3ammdmFicbBmakUylbcSXwUAJb1mVMzLmcl+S/yTOcbaQaxPoc2WLFawleK
yDvYqOfIBbEH4AfoZdrXF7fsNgTmT3fxBjA1h7f1hlwEwCf5hogz8mf+BrNzGigDZtzSL6gKpwJREg
gVtxyALPgZN3nYmVe1VinH0je8VklNc9fDlmk2RZfLdoq1f5fTBq0EuXfaFtBD7Ofg1vMjFZfndxTSL
CfvJzMxEBf6l0hgLUgH55uliceK1pynxNeUxrLHALee5slGPHepduJFeDezxv0s1e9lxZz5+fGVzET
EYfOJ0siKRfWF2KhDmfix3NwLZf4p7kleidmlwwnt2dypxm29Wd+1yeWMveLVzgVcFeXI0lkrNejd1
vD4ret924TDZe3J4ASJme/l5CBDPfOF52QLefrZ9LlazedFh34XqedTx4lW6XdiB4aGKMdwN45VZ7
d+N5c0pbeL16FD3ReX56uDCYeiN7Wyl+eq179hDle7J8mAL6fe5+qIXHco5/Gnmyc41+zm2wdlx+
iGG+dYZ+dFXLdn1+dUnLd25+kD1heEJ+tbJJePN+5SlaeXp/HxDhepR/hwNlfSGAGYTTcPuGdHj
Pcg6FjmfzfcyGErWEFdDCEBFUrdTyDdElldkGDBDz8dyeCqTADd+OCaCH5eGiCRRD5eZmCHA
OQfG6BXYPVb56N4XffcMGMVGv9ceGK0WA5cv+JIVR7dBuleEi4dTCHgjyNdiSGrC+3duuF/yHV
d2+FgxEJeMCEZQPOe9OCc4LabniVQHbbb6STF2r2cMyQ+I9JcfKPM1OocxANkkgMdDSMIDwJd
TOK2C9edgWJxSGpdoul+BEOeAOGbQQCe1GCtoHRbX6cp3XXbq2Z82n7b9iXS15jcQOU+FLfci
2S0Ednc1GQ4DuLdFqPJS8KdTeNryF/dcGMRhEQd2KIMAUQueuOC04DGBKukFXTobdqg5Gkmb
wWdul2hcDWa1FI0cWOYH0bcc06VqTshc6CTeC7DdleRoyFbdRSPBxERdtuJqQRTeoeC7lCwhN
NbFHTuhHhdw2k5hCtgc12GhAhjJVHWg+Zl30Yfg8VooTn1g7trUy0Qg9Zt5x73hDNwPA4JhIByPw
Glg4Z4z3/OggdiQXQpgeBkWWiRgcNmcVz7gxcokFFkgdZqvEXDgeFs+DmrgfZvKyzWgiJxSR7Qg
n9zNQ4Agtt09AGegm96qn7uf4tpFHNqf45qomf0f5tsMFx7f8htz1D8f/hveEVugCpxMDlogFxy5iyigJ
Z0kh6ugPR2Fw35gWl3nAGxgXt8S34WfWpvsNktfY1wwGdSfbdx0Fvzff5y8VCNfkl0HkUSfpZ1Wjk
gfuJ2kyxvfy93xB6Qf5F44A3zgCh6NgHCgKZ9tX1De3x2aXHse7128GamfAR3eVtgfGN4GFARfMV
4xkStfSt5hjjSfYx6RCw5feh7AB52fkl7tw38fwB84QHpf9x/Anx5ect9N3EceiZ9QmXVeoZ9Tlqlevl9Z
E9xe3B9jkQre+p90DhufFp+Fyv3fL9+aB5ffRZ+xw4kfeZ/ewJAfw2AQ3ujeEuENnBSeLyDrGUZeS+
DI1n9ebWCrU7gej+CUEO1esuCEDgUe0mB4Su8e7WByB5LfAWB0w5KfO6BwwKQfleBW3rCdvu
LUm96d32KJWRMeACI+1IEeJSH+04/eSuHGUMyecWGXTexek+Fuit7esGFPR4yewyE+A5mfBa
DxwLVfBuB6nniddmSX26QdmmQnGNcdveO4Fhod5KNXk1+eDGL/0KVeNGKzjc6eWSJwCsted+I
5R4ReiilQw53e1qFIQMPfTeCEnjzdOOZeG2kdXmXL2J2dg2U7VeVdq2S5kzFd1CRCKH+d/WPYT
bleJCN6CrjeROMsh3weVyLNQ6DeriHJgNBfMiCM3gAdBagn2zLdK6d1WG0dUObElbjdeeYgUwp
do6WIEF/dziT/DZpd9qSFiqkeGWQeh3VeK+Nqg6OejKldQNqfGuCT3bUi4tbGmvqirFdxmEOieRg
dFYwiUVjJktNiKxl4kBYiCBopjTeh71rWSiUh5lt7RsBh+BwQgtOh0dy5QBohT953nYOiOlh62s/iDxk
DGB/h5tmLVW5hyJoV0rshrFqj0ALhkts1TSjhgrvEChqhhexxNRrthi1zKAtZha11kQCMhCZ7gHVOh
pZobGqchg9qC1/5hZFrVVLhTdtXkqUhOdvGz/FhKNw5TRthHRyqihEhGh0YhrbhKR19wtjhEZ4M
ACrgzF873SUhJdutWn3hCxv319qg8hxC1TUg4NySEozg0lzkD92gxx05zQygnW2Nigdgwh3ehrLg
0N4rAtsgw16vgDHglx+LnPZgsl1DmlNgnl1vV7Sgi12b1RSgft3M0nGgdV4Bj8egb146TPwgbZ5xCf
0gcd6mRq/gf17bwuEgex9JQD4gZB/UnMngTZ7emiUgPp7ul4YgMJ7+1OrgKN8P0k2gl98lj6qglh9
AjOZgl19bye/gKF94Rq3gMd+aAu5gNV/cQFXgMCAZ3Jvf8iCHWfkf6GB0F1wf32BhFMUf2+BRUi0
f2yBHT5Bf3WBETNlf4WBDyePf5uBIBqwf7OBWwvrf+CBcwGvgAiBl3GtfoOl5Gcmfm+H+Vy5flyH
ElJufluGUkgjfmOFrj3MfniFLTL0fpGEwCdZfqqEdRqkfrmEaQwSfwmDPAH6f2qBVnDofWuPnWZVf
WSOJlvhfVuMt1GnfWKLfEd0fXGKYz0+fYuJdDKKfaqlcXfcmIARqOfdGHaQwufk2E1gl5fuWBgX
AUfHyWZWWCfHuUb1sSfHiSglDoflOQy0bNfJePOzy2fLSN3jlkfNmMqSbXfP6Lsxp4fQOKCwxDfa
uGOwJvfnSBpW88e7WdO2S/e7eayVpje7WYX1BHe8SWJ0Y/e9uUHTxEe/ySTTHPfCeQsyaiFG
PMxplfFWMPQxVfSSHZQKcfhaBxGzzkoFbHGLukSxduFj3j+JgWk7vjrtjEETZjaRI0TqljKRomy/Yi+

RrUSQki4Ft5hcQi7dwOgjibluQAahcR7MmxQkARhlmJijtZjsliDjbJl0k6OjK9oC0SLi7pqUDppit1sp
C+uijNu6iQMidZxFxcRif9zEQkAiCl2XQAahP58gmu4jddnzGHfjMlpblgVi8NrFk4zitls3URCigBurzox
iUFwjy+GiKtyZiP2iFZ0LRcRiHJ10AkaHsp48wAAhE99qGsbi/VtyWFSiwBu/FeaihBwNk3OiTdxkUP
wiHBy+Tnwh8N0bS9Yh0F11CPdhv13LRcRhxB4dQkxhZh7RwAAg7h+qGp4ikZzzmC/iWZ0kFcXill
1WU1eh8B2RkOThwp3Pzmlhm54Ry8jhfp5QiPBhbt6MhcVhcd7JwlUhH19aAAjgxF/mWnViNN532
AbiAB6PVZ1hy96n0zLhnd7FEMUhdF7mTIBhUR8My7ahNh8xiOahJd9XBcdhlx+CwmWg2l/bQCK
gj+AXWk6h3eAJ1+DhriAAFXjhfp/30xHhVF/2UKihLp/6DjIhDuAES6Yg9eAPyN2g5SAexclg3OA6A
nSgneBMgDogYWANWiYhjYgk17hhZGF2FVChOWFJ0uzhEiEqElhg7yERDh+g0aEAC5NguiDyy
NMgqWDsxcmgnSDyAoDgaKCxQE5gOaA02fvhTGM914qhJCLvFSCg+2KjksBg1eJm0GFgtGlyT
gAgl6lHy3xggWHjCMWgcSHJBccgYmGdgoogOeEMQF9gGCBAWc1hE2Ta11wg7CRvFPKgxCQ
GkpXgoCOs0Dvgf2NczeHgY2MYi2YgTiLciLhgPmKtxcRgLeI0QpFgEWFbgG2f+6BKgz3g46Z61z
EgvKXyVMugIOVs0nGgceT1UBwgUiSljchgNqQpy1NgliPWiK1gEyN2xcHgAWKyApef76GdgHmf4
+BSWL/mZBbIVnll5xdzFDXlbtgeUealCJjLT5GkqJl6zTHkUVotCqjkDZrax+Gj51uABL1j8VwUQari+
J04AAAhZ98o2KOIzhhUFmCIWVjglCAk6RltkdSkiln9z4MkMZqQzSdj4ZsniqMjoZu6x+EjexxIBMqJf
1zIAbZill3fQAahOt9vmldlShnQFkfk3BpAVAskclqkxcMkGRsmz3VjxlufDR2jfFwaip1jQByUB+BjGZ
0JRMojGN11gcDiQB52gAAhEx+tmGmk15s+liwkbtuUk/GkCZvsUa3js5xlz2RjZFYOTRBJHl0KipUi5
d1px96iwV3FBM8ivd4cwcmmh9R76QAAG8J/jmEpkcJytlg5kDNzpk9ZjrB0m0ZYjWV1pz1CjDZ2vzQ
Eiyx35CotilR4+x9uicZ6BhNTiad7GgdWhr99yAAAg2OAAGCmkF14eFezjt95C07SjWt5oUXdjDB6
PDzYiw965zOyihB7pSn1iT18XR9XiKd9FRNviGd97AejhbB/igAAg0iAAGAmjxh+aFc1jax+gE5YjEp
+nEVuix5+xTx4igp/AjNniRR/VynCiEV/sB9Dh6iAGROIh0yArQfphMKBewA2gtuAJF+cjfaEeVaajJy
ECU3Ni0qDoETuiimDWTwliR2DKjMPiC6DGimFh2GDGB8lhr6DNBOWhkuDNggig/KCcgCLgjQAX
l8FjP+KhVYGi7CJoU0iimalx0RPiUylFDt9iEWHgDKeh1eHECkzhouGtx74heSGiROWhV+FjQhOg
zqDrgDTgbOAj15fjDCQplVgiuWPV0x8iZ6OEKO2ileM9Dr2h4OL+TlxhpWlKijkhcmKex7MhSCJ0B
OThl6HnAhxgpuEwgERgUCAuF22i4aW1ITGijqVGUvwiPKTZkM0h9+R2TqEht2QdjHVhfCPRyihh
SSOQB6mhHqMjROQg9yJVAiPgheFqAFegOGA21kaoOpbD1DonlVdzUi9m9xgiEBMmc1jOze3l+
Bl+C7rliNowSVslMNrexiR/RuEg84k11wqQS1jCR2OQAAhX595Vjnnqtg+1CynDVjRUiKmdllj0Afl+F
n2TeUlqxqLi7VIGdsjyVqkw9u5xr6kjVxJg9rkYdzcQT0jEF4mgAAhNp+0lienKxmQVBzmk9ojEhQmA
pqcd/wliZsVTdxlGduRS6/ktZwQSVmkYZyOBsOkKR0Hw+Zj+N2IAUqiu16sAAAhEp/o1hRmu9sKF
AkmKZtqEgBlNjvLD+slJhwtjc5kuRySi6XkV9z6iVUkBl1fhsajzl3Aw/Ajm94tgVaicZ8gQAAG8yAAf8
mVxxok/NlylywUeslQlZ5D9gkzt1Dzb6kZJ2Ri5pkBd3iCU9jtd4vRsgjF55w/njRI7VAWViLR+JgAAg3
eAAFeHl/53G09pleJ340dCk9Z4rD7+khH5bzamkHp6Py4qjwd7ISUXjcd7/xscjNZ83hAUi9p+DgXsh
6t/rAAAg2SAAFdAls58uU8lIMN9D0biksd9Zz6lkRd9vjZaj4V+KC3xjhd+qST1jNd/MBsZi9l/yBA7isC
AfwY7hsKBAAAAG1mAAFBnlciCc06Tk8qCTEZukdqCKj46kDSCGjX9jqmCIC2pjT+CRCTFi/6Cdh
slivSCyRBVicKctAZ7hfaCMAAAg0+AAFZlIOMlE4EkvWHm0XXkQ2HEz2vj2yGnzWDjeSGRy1G
jHqGEiR/izSF9BrliiGF8xBfiNeEvAashUODQQA/gt2AK1WzIDGN/E1ukkcNCUVDkFuMHj0mjr2LS
DULjTWKkizli8mKBSQ4ioGJlxq/iWWlZBBjiAeGhgbVhKiELwB/gmmAVIUbK56T4EzkkauSgkTGj8K
RLjyzjieP9DSmjKCO4CyTizON/CP9ieeNFBqgiMWLLBBmh1eIAwb2hCaEtQC1ggiAek7nqK5bFE
8pc9dtEAeov9gWDiLoFhjETDlneFl1Si8m61opR/tmeprZxX4mN9uAwvll0xxSALoj4B3iQAahWB/D
U7opnFgtkd4o7Vi7UAUoQVIKziAnnFnHzDBnA9p7Ci+me1sWyAEmCpuwhYwlwdxDwwSIKx0CA
M1jgF5oAAAhMt/0U7gpHBmLEdmodFoAj/7nztP4DhtnLtr3TC1mmtt5Ci9mFpv9SAWlphyAxZglWJ
0AAxTkWB2sQN4jLJ7dQAahEeAAE67orlrcUc4oC1s6j/Hnadubjg8mzfFwGTCNmOpzxCilluFziSAXI
SN1OxaCk+d23QyMkYV5QQOzi499DQAAG9WAAE6BoStwoUb+nrdxxT+OnERy9TgGmdp0SDB

gl5x1piillI3DSASk914aBafkpZ5tgzDkC17zwP4ioJ+fAAAg4mAAE5Pn9t100bAnXx2pj9Fmx13gTf
BmMN4cjAllpB5cShglJB6gCACks97ihaykXN8lwz+jvB+OgRYiX1/ywAAg36AAE4Ynr57JEaDnGx7i
j8Emhp7+zeDI858iS/xlaR9KSg8k6l93h/zkeN+mxBCkHN/ag0zjdqAVgSviJmA7wAAg3qAAE3LndG
AjkYum4aAfj6rmTyAfTcwlvmArC+plNaA8CgFktuBUB/TkRCBwBbBj4+CTw1XjN6CPAT2h9GB8w
AAg3aAAE1mnQyF90W5msiFhz4rmlKFJza6lkOE+y9BICGE6SeykiSE+B+ZkFKFHRaojsCFFw1pi
/WD/AUthyGC3QAAg3KAAEzknG6LdkU4mimKrD2rl+KJ8jZElaWJbS7ak4OJBidekYOlxR9dj6mlm
RaLjgqHig10iyiFhQVZhomDngAAg2+AAExUm/GRCUS6maaP2j0/l1qOvzXhlR+N4C6DkvyNJScY
kPmMlx8qjxqLoBZzjW+JIA1+inyGzQV/hguDtwAygxiAlqg8YTZWLZl2Y3BZgoq0ZZ5c1XvkZ6ZgA
W0labZjIF4Va9BmLE7LbetpET7vb/FrwS4YcZ5uLBqGcitvwwZ0eJ5ye6ZHXQ9fgZfCX7Jh/olAYkhk
eHqgZKZm22v0ZwJpO10vaVprl04Oa6ht1T5XbdBv6y2pb4dxyRpNcBNy/gaWd0x1naSBWWtobpY
3XGRql4fvX0xr2Xl8Yettj2r7ZIRvQVxgZxZw7U1laZJyHD3Pa9h0AS1HbZl1VRobj52MAazdiN4raL
dVjRxKJTDWxxyFoarXK5zB3hmX4R0EGoQYk51G1udZQp2l0zGZ6p3HD1Tag14ACzxa9Z4zBnv
bKZ5VgbMdSF7XqFdU2J5vJNsVuN584V+Wk16L3diXVR6i2k0YEt661rnYy17UEw0ZfB7qjzkaGx7
+Syoaj58QxnWa0l8hQb6dEB9v6ADUPCCIZIsVKGBp4RaWDmBMnZmW2mA5WhfXoaAoFo6YY
yAY0urZG2AJDx/ZwB/6CxvaN1/uRnfaix/vgdVc5x/2Z66TrqKh5EFUpaJXYNYVliIN3WGWA2HPGe
jXO6GTFmgYBWFaUszYxGEkTwnZbmDyiw/Z6ODJRnqaTeC9weocxCBj14TMKS4o/kUL+RBoJ
YVKGPMHSqWBWNimbpW3OL81kIXreKbkq8Ycyl/jvSZImHsSwQZoCGmhnwaF+GGAfvcpiDPJx
ESwybG47CTx2YjoFNUxOWCHPEVpyTxWYOWhCRk1hpXWqPdUpAYJaNeDt7Y2iLrCvhZW+KK
BnwZ6GI2AgqcjOEjZsYSYejSY2wTaigGoBXUayc9HLwVUeaEGV1WM6XQFfWXDuUiUnOX3yR/j
srYmKPtCu3ZHmNyBnuZvuLNwhdcd2FqJoASDGrY4y8TGKnqH+GUHaj8HI+VCCgYmTfV7ic61d
bWzaZk0luXomWdzroYYKTryuTY6eRURntZnKNMQilcZWFxJ6lZwFWa5CCaNpZo4KBaqlc23Rw
bE9f/GZXbf1jFFgrb7ZmHUmdeXVpBTpvcyprvSo9dJxuLxcvdXlvwwT8eu9zgZzDYyJfJI72ZVxhtml
EtZ4hkC3NMaXdmVmVga2ho3ldfbVxrQkj2b0ltjznqcRpvuSnfcpNxqBcAc4hy4QUaea52hpsqX61nl
o2PYjhpTn/4ZLJrCHJFZt1sz2SDaQduklaaoazJwUUhhbUlyAzl1bzRznymMcLN1DxbXcdV18wU1eJ
R5Q5mtXJpv44xBX21w437aYipx5nFPZlhzCGOzZuJ0LFX8aTR1UEfXa292aTkKbXd3cilDbv14YB
a0cFp4+QVMd597pZhPWeV4EosJXOx4Zn3JX914vnBjYmp5O2LtZO15vVVYZ2R6RkdVacF6xzio
a+F7QSkFbXB7sxajbxB8DAV4dsZ9xpcMV4aAHYnbWrx/yXyzXdh/e29yYlp/V2lhYzJ/PFSyZct/K0
bTaER/HDhMant/EijVbBN/ERa2bfZ/NgXSdil/rpXXVWSILYjDWMKMHMu7XAWGOG6bXtqFb2FsY
aKEslQfZFuEA0ZgZu+DYjf6aTqC1SisatuCZhbJbQGCXQYldZ2BV5SnU4GQN4ewVv2OjnrFWI2
M623GXU2LgGC4YDGKJFOMYwSI3EXvZbGHRDeraBCGoCiEabuFxbWbCuFPwZsdSaCxJOG
UeCYJlacVWyV0HnDWN2ThmzmW+CRg1/6XtePIFLxYb+Nu0V3ZIKMBzdYZvWKgyhZaK6JRBb
ca3CHxgandMGD95JnUG2gB4WVVAedGHjVV4WaMmwXWpeXI19KXZ6VEIJhYJiSqUUIY26Qbz
cMZfaOdigyZ72M1xbfas6J9QbadGuEopFVTyWn14SoUs+kX3gMVlug7WtoWXqdsF62XJCajlHpX
5mXjUSrYoGUyzbMZRqSXSgSZu6QLxbhakeLxgcEdCSEvpTdbOIWhoeRbmFZqXpNb9Fcy20Bc
Rxf4V+vcmpi8lJLc75l+0R5dR1o6DX7dnxrqiZtd7NulxPYeQBvtAOpfQV0mpNVaTtev4Y6awthK3k
mbNFjl2wDbmJmA17Zb/BocFGacXxq3kPrcwVtOzWMdHtveSYhdbVxehO2dy5ytQPFe9R3XpHuZ
exmtoUBaAlodXgbahhqNmsfa+NsbV4Zbalt1ID8b2hVqUNrcRdxdjUpcqVzMiXec+B0vROYdZV1qg
Pdesh5ypCfYvZukYPcZVdvpncgZ6ZwwGpJaaFx9V1la5JzMFBBobXh0cEL2b0p1qzTQcPN22SWicj
d36BN+dDJ4kgPyed975o9nYft2X4LFYu520HYrZW13Rml2Z5J331y0aat4g0/Xa7R5MEKDbad52
zR8b2h6gCVxcLJ7GRN3cvV7igQdeQ19zl5EXhR+FoGuYNF96XUmY3d9wWiSZb99wFvzZ/l9zU8
6aiJ96UIKbDF+CjQpbgd+MyVKb1d+YROUcdt+pgR2eHV/il0IXAeF1oCqXuiFB3Q8YbKEPGfHZB
mDnFtGZnKDDk6taLmCIUGeauKCLDPfbMuB2SUqbiCBoBOwcOaBpgTJd+qBC4wGWjeNIH+kX

TSMH3NSYBiKrm6YpiJclqYZQmITE4gZ2qHqkExaamGVDOVa6WFiSUIbQOE7RPEcA+ESwU
Pd3OCV4rwWKeVOH6bW7STHnJZXqeRDWYeYTePQFndY7qNjk2IZiyL/EC9aIGKkDNGapGJVS
Tia/qlXBPQb1WGoAVKdw6Db4nfV0ic132eWI+aJnFxXVyXf2VSX/mVIFkuYoqS4Ez7ZQyQxkBSZ
3OO3TL9aZeNNCS+awyL4BPXbrWlowV9drmDtojDvhakaXy7WTehL3CtXD6d/GSmXuabBFicYY
OYLkyEZBSVhj/3ZouTHTLAaMGRCiSgakCO+BPebi+KUQWndnKD0osUcwtWj36BdBpZoXH2dS
VctGVodhdfvVjXdwdixkw6d/hlzT8xePRovTF4efdrhyKeeu9uBxCyfGZvugJ1fup1sonNb31eYX1icO
FgyXEBcj5jMWSWc29ImFgodJxoBkurdcVqe6/duls4TEheABvLyJlePlxPBCfeqpyoQKOcd4loicbE
9l5XxZbftnrXAgb5xpd2PZcQRrTVeLcmZtKUsrc8BvCz5ZdQlw7DDUdjVyvSlzdy10XBCOeSR1egK
lfMh6RYeEaX9tQntha2lucW9JbURvo2MkbtZw61b0cGFyO0qvceFzID34c0t07DCNdJB2OCIGdY1
3YxB/d9F4RQK4e+p8IYZ2Zvl0pHpvaRZ1OG5zayJ1z2JsbNp2hlZabol3SEoxcC14GT2XcbR46TB
Jcw95tiHhdA96chCCdqB7lwLheyJ90YVwZMI8B3lyZwx8BW2EaTx8B2GZaxV8KIWnbON8XUmhb
qR8pD0pcEZ88y//cbJ9SyHFcrN9pRCndYd+LQM6eot/ZYRsYtODEXiEZTeC2mytZ4aCPmDdaXu
ByVUHa2aBaUkfbUOBlzzGbvyA8S++cHmA1SGtcX2A0BDKdJOA8AOMegGAxoNIYReK73eQY5
SJrGvPZfulbWAZaAaHY1RgagiGdEiYa/uFpjxgbcqE9i97b1iEayGScGGEcXdkc76DXgPTeYqB9Y
JjX5qSUHaTYiWQbWraZJqOkI8/ZrKM/IOlaMKLiUgBasWKPDvubKeJFi8vbkmIlyFxb1uHbRD0cwa
FhQQOeSaCvoFeXk+ZrnWcYOKXOWn0Y16Uzl5yZYGSrIL1Z5uQskdzaayO5DuDa6CNSS7pbV
SL8CFRbnCKzxEAcmehYQRAeNGC4IBfXTWhAXS8X82eB2kxYk2bFV3FZHiYYIJhZp2V2Eb8aL
mThTsparyRcy6ubIKPtIE3baeNqxEKceOI7gRqelqC/IFmeUVWj3WFef9ZiWmxerRcg13he0RfhFIS
e9BiiEY7fFIkDn5fO1ohi0DfZFrWR7Xfkt3w3ef3hv7wFegKV2toBVdeNeA3SbduVgXWjtd+Bit108e
KtlHIGLeXFnjXOejBqCjmleupsfyzFe6Fu3B6yfFlw9w3Yfc5yvAF3f454239XcttlH3PAdBxm4mg2dV
NoqVyldlBqilEQd0dsc0VseDhuZzlZeRVwXCyOed1yRB6RepRz9g3TfF1ewGMfpl6vH5hcDRsAnL
scalTPWeEcxFufFwRdDRv2FCWdVFxQEUJdmhytDkNd2R0KSxaeEF1IR50ePp22w3PexR4KgGff
cR8X310bc5y/Xlab3FzpWbNcQR0Ult0ckl1J1ATc4h2C0SfdMB3ADi8ddh39ywkds46x5dd4F5yw3
bee167wHlFp1933yQa7Z6BXE5bXt6JWX0by56SFq0cJF6kU9wce167UQbc0B7YDhadHB73ivmd
W98ZR5LdiR86g4leNd90AlifGV/TXupadGBJ3Bja7WAr2UwbYaAO1oHbwN/807ccHp/xEOIced/sz
gCcyx/uCuudDp/0x48dO2AAg4xd+SAvgJ0e9qAjnq7aCeIVG+DaiOHO2RfbAuGKVINbZyFVE49b
yaEnUMlcKeEDDeicgCDmytxcx2DUB4pc9GDKw5RdxGCKQK7e2KBoXnOZr2Pb26SaMaNvWNv
arqMFIh1bFaKvE2DbeyJh0Kpb3qlfjczcOSHnysqchOG8x4NcsyGgA5mdlqEjAL3ev2CAXjaZYeWj
22IZ5aUUGKLaY6SHVeoazGQPuzSbM+Oh0IBbmiNBjbJb+KLuirmcSOKrx3yceSJrA52db2GRAM
peqiCl3fsZIGdr2zNZpGa6WHKAlqYL1b6ajOVwkw8a9mThkGJbXyRiDZvbWSPzSqtCFWOZh3bcR
2MUg6EdTqHsgNUemGCQHf6fFWnmzWf8hZqGHPf/Fcr1blgBZfq0u8gD5iq0CegG1lrzUEgLhoo
CiigS1rbhr2geJt7QswgnVwlQBBgod39Xb7fGddsmwHfNRgKGEhfUFinFY3fZtlCEtFffdnfkBAfldqA
DS+fsRsdhyf0du1Brhf/Vw7ws+gNlzVQBngXF513YWeYlkdGtEejJmYGCAetdoTFWye11qOkrae+
VsMT/rfHFuMjR+fP1wMShlfY5yHxrPjVz2QtJf3B2BwCJgH17fXVAdw9q/WqJd+ZsbV/geLVt3VUs
eV1vUEptegZwzj+UerJyWDQ9e1lz3Sgfe/11VRrAfKJ2qAtTfjV4qQCmf6l87XRtdM9xlWnMdc9ygl8
5dsZzb1Sed4t0ZUn3eFJ1aD81eRt2eTP4edt3hyf0eo54jRq0ey55ggtsfRZ7UgDZfuF+PXOnctB4P
GkEc/N4rl50dQp5HVPvdet5j0lidsx6ET6+d616qTOgelB7RSfBeT175hqwedN8iAulfAV93wE7fhh/e
nLacQZ/BmhEckd+513Dc3t+xINSdHR+r0jbdW1+rz5TdmV+yjNSd0p+9SeTeBF/MhqseJ5/hQvZex
eAGQGVfWeAj3IEb3GF5Wd5cMuFKF0EchWEbVKncyCD0UhJdCuDUj3edTWC9DL8diqCsCdfdv
uCjRqjd4WCIaWDeKaCEgHhfNKBRXExbhOMtWafb32LZlwpNaKHVHhceuJA0eecwCICz1VdBW
HPTKXdRiGkicidfeGFBqSdoGFwwwheZGD1QlifGWBcXBTbOKTjGXGbliRtlYb7qP51EkcNWOT0
b7cfKM3jzScxCLnjl3dB+KiybndQyJtBqAdZmlmww6ePWFXQJZfAiBlm91a9uaaGT/bVqYDFqnbsS

VuVCEb+WTnEZxcQiRrTxkci+P+DHmc0qOfSa1dEONSxpxdNaK+gxOeHOGowKHe7uBtW5vhih
WpmQyhdBZRVoDhXtcsk/lhShfr0WAhOBirzsbhKxlszAnhKhooyRUhOtrbRcehaBt5QjThRlxYAAAg
vV5jm2jgxddYGOBgwFf3lltgupiWk9Lgslk0kUbgrBnUjrNgqZp3C/wgsBsWSQ0gw1uuRccg7Nw1Qj
xg4l0EwAAgjV7EmzegGRjz2LZglRlzljigKFnzE7ZgKVpzETAgLnr1TqGgNBt5y++gQNv8iQYgVtx6
hcbgfVzrAkLgip2uAAAgY58ZmwkfhBqCGl0flrk1hUfpxtHU5jfr1urURgfudwRTTo7fx9x6C+lf2tzfyP7f
9J1BRcZgGR2ZwkigPl5SAAAgPx9jmtpe/VwRWGMfGNxWlAfMhyb03kfQNziUP3fUZ0rznPfZh14
S9Offd3CCPbfmp4lhcbfvJ5LAIGf+F7rAAGgl5+j2q2ehh2j2DZeqR3NVcPeyR32E1He3h4dkNxe9N
5lzl+fDp54i8CfKl6nyOzfSB7WxclfZZ8GwmJftR97ABvf8N/oWoDeGZ9AmAveQ99HIZweat9OUy5e
hV9V0L4eoZ9iTkfewF91C6+e35+JyOPe/p+hhcufGF+/wnHfep/5ADPfxGAjGIJduKDkV97d6WDFV
XFeFmCnEwfeNOCPUJ0eVSB9zi2ed6B0S5yemiBvSNmeuqBwxcwe0eB9wn5fRyBowEhfnIAw2i
QdZoke165dmaJD1T+dymIEUtqd6yHO0HXeDaGhDg5eMqF8y4YeV+FfiMyeeqFLxcqekWE4gog
fGiDNAFmffeA8mfKdG6Qol32dU2PHIRBdhqNokq+dqSMVEFCdzSLKjfCd8+KL3CeG6JWCMAe
QKlthcieWOHcQpAe82EkQGhfYqBGmcDc3GXN11DdFuVMIOhdTCTNkotdb+RbEDFdlaPzDdedv
eOYi15d6CNKiLWeD2MFBcbeKeJlApae0yFtAHSfS+BO2TwjOZWpFuWjBRZnVJli0Rcl0jcinBflz9
YibBinDWqiRFIpcTaliPolSARiMxrXRNOiX5t0Aa8h2tyTwAAgvV7HGRli5dCFsDiWhfglHLiNBh/Uh
0iChkgz8Gh5NnEjVthxlpqCszhttpsLSADhvBukRNgh41wrwboheZ09gAAgkh8ZWOoh3FjLFp6hw1l
LIFYhqNnNEgVhhxpSj66hahraDU1hVBtjysPhSdvqR/2hUNxqhNwhc1zdgcPhJB3jQAAGbF9hWMI
hT9pG1nrhQBqsFDdhLhsSkevhElt+D5og+1vrjT3g6txbSrmg5ZzGR/mg710rhN+hDt2lQcxg2d54g
AAgS5+gGJkg0hvCFIYgydwMVBbgvtXkdAgqNyoD4Ngl1z7DSxgjB1QSq4gix2gx/Tglp3shOPgsh
40wdggLZ8AAAgNZ/UmG+gZN0+Vi2gYZ1v0/AgW52hka1gSx3Uz2WgPt4LTRUgOF5Fip4gOh59
h+3gRZ6zhOogWI7rQetgU59/QAAGMd/8mEkf/h7F1ggAd7W081gAl7okY3f9x79z0rf798XjQAf7Z8
2io+f8d9Vx+ef/R92xO/gDN+egfygGZ/uAAggliAFmCKfoOBU1eJfq2BCE6dfsiAxEWufquAozy1fpyA
mjOjfqCArCn9frqAyR9/fumA+xPNfx6BSggsf5uBRAB3f++AUv/pfUWHh1bdfX6GwE3faaGAKULfZ
KFcTwofYue/DMzfZeEqimufbmEax9VfeuEThPSfiKD6ghafumCqADAF22Agl8+fDGNzFYxfHSMKE
1BfKSLXURwfJaKWjuhFJWJeJLHfKelwylhNCIKB8sfQeHuxPUfUaGOAiAfk+D3wD/fv+ArF6Qe0aU
GIWTe5CSY0yye8WQtkPte72PPzsve8KN8DJse9mM0ykhfAiL3R8KfESkyhPVfI6lJAigfc+E4gEyfQ
OAz1tak7xWnVLpkkxZpUqAkOZcq0HZj6NfqTkQjoBirDARjYtIsyZijO9ooxuljNlrZg+ujPxuDQTSiXh
zhgAAgvV8glrfkP5ctVJ6j8VfRUoejpNh1EGljXZkZDjRjHJm+y/li6BplyZMixRsIRurivtuiA/ViwZw3wU
Mh/x2lQAAGlp9lVpijpRij1lOjYdksknDjH5m1kE9i39o/TiWip1rKi+8ieNtXiY3iWdvhBuwiUtxjw/3iUJz
mAVAhq54fwAAgDJ+h1nhjIFoOVGYi5Rp+ElZiqhruUDkib5tfzhQiPBvTC+JiEhxHyYch9ty3huwh8V
0ghAVh652NgVshYx6jwAAgVt/WVljp9t4VEhidFvO0jriQJwl0CFiCxx+TgCh3FzYy9Phtl01CX5hnh
2LxurhmN3chA0hj142gWIhB8bAAAgQ+AAFjciP5ziFCaiEd0hUhn451gkAOhs52djechiZ3dS8BhZ
14gCXHhUJ5fxuehSV6cxBbhOl7ngX7g35+KAAAgQiAAfhch395VIAfhuN52kfxhkJ6Xz+ihZZ64jdA
hP97dS67hIV8GiWahDB8vRuShAp9ZRB+g8F+UAZlgpt/qwAAgQeAAffWhil/QU+YhaV/QUdshRt/
RD8phH1/WTbYg/J/gS5og4N/wyVigzKADBt9gwaAaRCWgriA0AaJgdWBBAAAgQeAAFDHhQSFK
U79hJCEt0bKhBSESj6Ug36D9TZWgvuDuS4BgpCDnCUagkKDjxtagHODoRCigceDIAa8gSeCOW
AsgLyAHlASHAuLJU5hg6CKRkYugyuJbj4FgpulrzXZghyIDi2cgbSHkiTTgWmHLhs3gTmG1BCqgP
WFKAbmgJCDSwBtgE6ASIYRgz2RLE3SgtaP2kWrgmOOkD2MgdiNZTVvgV2MXC1lgPiLgCSXgL
CKxhsZgH+JhRCwgEaG2QcJgBOELQCjf/GAb1HMmt1WdUpCmMxZkUK7ls5cpjralRZfpzLLk4piq
ip4kxIsiFnkVloohczkR9rYgx7j+lujgMji0F05AAAgV9u1GEmD9cSEn6lmRe8UJ2lJhhmDqhkv5kNT
KhkY9m1SpfkFtpeSFlj4BsCxdRjzZudwy1je9xVANqic13RwAAgml+oFEule5h30mVldtkJU11kpRma
TpskRhoozJ4j8Vq4SpGjgdtJCFHjdVvWRdqjX9xcAznjCp0AgOilZ5XgAAge5/aFDPk/NnSkITkl1pN

EHekNBrHjohj2ds/zl9jiRu5CofjRJwzSFTjElyoRd7i/N0VQ0Tip12IAPEh2l7MAAAgYOAafb6khxss0
j6kKluP0GEjzxvyznRjedxTzH8jLNy2Snxi610ZyE+iut13BeHio53Ng0/iTl5KgQfhnN81AAAgUCAAF
AaklZyFkiUjzFzTEEajeB0fzlwjJ91mzGpi312vymzioR37SEbicJ5DReliVN6IA1zh/R72AR9hWZ+Vg
AAgUGAAE+3jyF3m0gyjeV4X0C7jKt5ITkYi3550zFeimx6kSl6iX57XiD7iLx8KheOiD18+Q2ghth+R
QTShlh/pgAAgUeAAE9Fjet9OkfAjMR9gkBKi519zDiwin1+FjEDiXZ+cSkziJB+4SDNh81/WReDh0B
/4w3BhdyAdgUYg8aA0gAAgUyAAE7FjOOC10c0i8yCuT+2irOCnTgoiZqChDCMiJiCgCjTh7WClyC
KhvCCvRdmhliC+g3ThPeCewVPgxyB4AAAgVCAAE45jAiljUakivmICz8mieaHjDejiNKHEDAXh9O
Grih1hvCGbCBIhiqGPxdHhYmFzg3fhDCEQgV8goqCzAAAgVOAAE2si1iOU0YhikqNYz6siTiMeD
cziCeLmS+1hyqK2CglhkmKPCAPhYCJqBcshNilKg3og4qFvQWighCDkQAmgRaAGkflomhWUkE
ooAhZVjpxnbBcWzNrm2JfZSwpmVFicySPI5JlghwmlIhoeBKKldtrQQmKkn9vRwGWjOZ2PAAAgv
V+20fKn99b2kElnbddedjpPm49hFjNQmVljySwXl19mfCSMIbVpLxw7IHprzxLCk+VuSgnVklhyAQHo
i3h4VQAAGnZ/IEe3nY5hOEDtm5NjdjoumZVlujMzl39oECwElZ9qaCSHIAZswBxMkstvChLykiNxN
woWjs90oglxijd6KwAAggiAAEd/m59mbECvmcJoUzntl95qQTL5ldpsRivWIAZuTSRtknRwVRxOkTt
yQxMXklx0EgpPjU93KQJwiSB7xQAAGaeAAEdFmddrkEBvmB1tJDMqlluvzK7IGdwbyukkqFylSR
OkRVz1RxJj911bBM2jyF25QqHi/d5sAK5iB99MwAAAgW2AAEcHmFVwrkAqlrhx8zlelQ9zPDJ0kzJ0
iStmkXx13CQij/t3NRw3jr14fhNMjeV5uArGir98HQMehyd+gAAAgXWAAEbNlwR16j/qIX52xTkbk+t
3pDlzkij4jSsukHt5gCP6jwN6fxwnjcB7exNgjM98eQr+ibB+OgN5hk1/ogAAgYGAAEZ+leh7QD+UI
HR7pzjAkVJ8FjHdkTZ8nirjj5p9NCO/jiZ92xwEjN5+ihNii9d/SwsniL+AlgPFhY+ApAAAgYuAAEYSIQ
KAIT8ak5uAnjg8kiOAsTFkkG6A3ip3jtWBHCNojWKBbxvljBKB0BNNivaCGAs+h+WB5AP/hOmBjQ
AAgZOAAEWtIEiGCD6YkuaFtDe5kXKFazDqj8CFPIoMjimFJCMPjLWFJBuKi16FNhMzii+EjgtNhyi
DbgQwhFuCWAAAgZuAAEURk7SLlj4fkICK1TdKkNuKlZCEjyyJmymyjZaJKiLFjCGI2BtVisSIQBMd
iYaGnAtahouEuARYg+SC8AAAGaGAaj1IXltRu09rXxJU/IF1YYhYpHNrY8VcJGVYZgxfl1cwaGBi
90ivaq5mLTmVbNtpKClyboprzxZdcEFtcwR6eG9xNptkV8tadY22WtBdToAJXb5gJHI6YFpi5WRcY
vVloVZjZZFoVEgKaBlq6TkUamptTykbbB1vcRZFbjpwwQS3eAh0cZmXU49jOlwwVvplUn7lWkpnB
HEqXTVpgmN5YBtrklWqYvltm0d2ZbZvjigaCdxYSjOadxzABYwbHR0AgTtd613lpfuT9Nrw4q+U55
tHH2OV0lud3AiWnVv5mKfXZRxVFT5YKNyv0bqY4h0GDg3ZhZ1WSiLZ8p2ehYdauh3NgUbd196
WJZzTH90HYluUJZ0yXxrVII1eG8sV/F2PmHUW0V3CFRWXoF31EZqYY14ITfZZDV5SChVZex57
xYeaZR6cAVbdvN8xZUKSZR8RYg6Tex8R3tWUh18S24/Vbd8b2EPWTP8mFO4XKB8yEXzX9B89
zeJYpJ9Jyg0ZFZ9XRZEaH59rQXEdkF+4ZPPRu+Ea4ciS4CDwnphT+qDG21tU7KClmBgV2GCGI
MtWu6BqkWJXj+BRjdCYRuA9CgYYuyAvRZoZ4+A5QYkdZ+AtpK7RI2Mg4YSSUyLLXlyTeGJ2my
gUdClSf+2VaSHk1KkWVOGhkUjXMOFkjb/X7mEwif+YZuElRaFZryD+wZ2dRSCSJGfQnKUc4UI
R02Scnh8S/2QeGvOUAmOtF8HU/mNAFIXV8WLYUS6W1KJ5ja9XmOInSfmYFmHmBacZf+Grga
7dJ+Dm5COQlucUYQNRX6Zs3ebSkaXHGsnTmuUvV5mUnOScVGVVleQPkRaWf+OPDaBXSU
MfSfPXzaLGBavZVuJAAb2dDyEtY+PPtekE4MvQ+ag6HbcSMidwGprTQOavl3fUSGX01EoVRuV
BkQJWNqSezZOXB+QRye9XkCOfBa+ZNOK8Qcnc+mE1pQkYjBRr4bjZF1VOnmnZnlYw2xZaFpc
N18BakNfoFGWbDhi+UPCbjFmLzVHcBZpLiW5cZlr2BMbdBhteAM2epNyYJJWxVaQIVWYGRd
DnhbYu1f22tBZR5inl4aZ1JIXVDaaYxoFUMua7lqszTXbbttKSVybz1vVxMOcjBwqANtejZ1epC0Wc
BikYPvXMJkqncX6lmw2pFYiJo5F1LZJprAlAyZxNtGEKpaXBvIDRza45xDiU0bQ5ywxMCclVzzAO
deeV4R481VihqtlKIWYRsGnYZXL9tg2laX3dvBVyIYihwh0+VZNByCEluZ1VzfDQYaY903CT9aw5
2FxL4bxJ24APHeZ56t43dUvdysIF2VpdzcnUTWhR0OGh8XQV1GVvPX+1l/08AYr126kG8ZWh3z
DPHZ714pCTSaT55ahMBbcx5/gQDeTh84YymUCZ6goBUVAB6o3QKV7V6yGeaWtZ7DVsUXed7
W05rYON7sUFMY7F8CTN+ZiB8ZCS5Z6t8wRMvbLZ9KgRrell+y4t+TZSCVn9JUaSB0nMfVY6BU

2bSWNqA+FpvXBSAQ3nXzeAZ0DpYiaANDM/ZK6AEySkZkSACHNba8WATATJd+mAdlpiS0iKI
H5IT4KI9XI6U5SH0GYOVwSG11nMWmGF701IXaSFGUCJYLKEXjMBY1ODxiSQZPaDWRN+av
GDIQUad2GB4YIZSUIRxH1NTZuP9nFRUcaOL2VEVU6MpVkiWMSLLUzeXCGJzUAIX0qlkzLDYg
aHiSR6Y72GvxOZajWFnAVedu6DFIhUR3yZWHxgS+WW8nB8UCSUImlSLU8CSeFiHV0uQcUxh
Wr6Ohj/JXgGMzDKKYNaLVCRnYqqKMROwaZOHwAWYdoyDyldcReGg0XuKSmOd4W/ITrma92
PvUmeYQFgEVgWVo0v5WYyTKD98XOWQ7jJaX9KPCSRWYcWNZhPCaQyJiQXldjqD6YrqZ/VR
5H5eacdVWXHZa4IYzmVLbRFcNFiybp1fk0wCcDFi6D7ecdBmHjEEc2pplylKdMFr0BAdd6ZtmglS
flFzmolXY71Z/n0BZfcdwnCzaB9fhmRRafhiR1fia89IB0tabaNnxD5cb3JqbjCmcSNS9CHUcnNvLxA
bddlwrgJDfC12bYfnX+hh23vFYoBj92+qZQFmE2NwZx5oOlcoaTZqYUrFa0dsiD3obUFupjBSbwp
wrSGkcFRyeBAZdERztAJue+R45oaYXHVplXqjX2JrCm60YjNsg2KhZlpuFFZ+ZtZvqko8aRRxQz1
/azZy1DAIbRp0UyF7bmN1qRAXcuR2qgKUe6V7DYVIWWNxnHmSXJFyDW3IX59y6WHZYitz5F
XZZKd050m4ZxB18T0baVd29y/Da1V39CFbbJ542hAnca15rALNe0R8+oRNVqp4tHiKWg149WzT
XVB5OmEGYAd5n1UoYqx6EEkrZT16jjyzZ6Z7ES+Dab57mCFLaw58GxBbcJl8xgMzepd+toM9VD
CAOneUV8R/3Wv4WzV/hGBKXhV/T1SLYOF/K0iuY5d/GjxXZiB/GC9KaFB/KyE+aal/ThCLb6p/uQ
OQefqAOYIyUfuHuHaiVbWGu2shWUuFwl+PXEuE9IPvXzeEP0gzYguDojv8ZLKDIS8SZvqCwiEv
aGaCihCzbticuQpgeXOBhIE0UBOPFWxU+ONempBV42L6F7MWqOKkFNLXaaJUUEXyJKIMj
udY1OHOS7XZbWGcCEdZ0GF4BDQbiCEmQQjeQKCM4A7TmGWZnTNUkGUOWlyVfuSFV4Y
WSOQMFK0XDmOaUc5XzmMxztFYhOLVY6hZl6KJiEMZkCJQhDpbYKGkQRceKGC839QTOCdo
3QAUNOa7GjDVJ+YPF2AV9aVxFl1Wv2Tb0bUXg+RRzr7YP6PXi50Y5CNySD+ZWqMPBD9bP2I
NgSMeFGDE4GUbexSBnW3b1tVamnkcL5YzV4OcfCjVlwcyRfekY8dFxiyDnUdaBl/iywdudpBx5Z
eA5rtQ1meuFt4AEIfkF0zoA/adtZvHSQa65ce2jrbXFfOF06bvFh8IGAcGxkr0WvceJnbjlqc1NqICxnd
K5ssR41dcxu9Q1teStw1wE1ffZ3SX8lZi9hKnODaFtjS2glanNlbVx7bDFnmVDjbeppeyEuyb5pr+jkLc
TRuKCwncqRwQx4Wc7lyHQ10d6tzvwFcfbV5dn3uYuZob3KNZWFP+Wc2Z8JrhFvLabVtJVBsa59
uzES/bXxweTi0bzxlyvucMRzvx37cdZ1Kw16dl12lgF9fXt7W3zkX+9vqnGgYqpwn2ZnZUxlslbZ2x
yqU/BaYJzx0RLa4l07zhfbWt2GCu3bwt3OB3ocBp4PQ2QdTF5fAG0fR99EXvoXVF22XCvYD53P
mWGYwx3pVpYZVd4KU8dZ5R4vEPJacB5YDgBa8N6DSuAbXl6vx3hbo57aQ3JdCF8gAlYfHV+o
3rzWvJ+EW/QXgp93WS9YQN9rFmpY3N9nU6LZdR9o0NwaCN9wDetakN98StPbA9+Nh3cbTd+i
A3+czN/MwJ0e9qAAnn9WNWFR27uXBCEdWPYxyiDqFj4YbWDCK32ZDSChULgZp+CIDdXaNy
B2Ssdar+BtB3UbASBsQ4pcmSBIALDe1WBMHkOVwKMY24HWk+K+GMWXXqJIVg3YBmlcU1U
YquHbEJgZSyGjTb7Z4KF1SrlaX+FTR3GavCE+Q5Jca+DrgMFeuWCC3gfVWiTdm0nWMCRfWJl
W/aPjVeDXqSN40y8YUWMXEHpY9iLAjakZkaJ2iqxaF6l8R23af6lOg5jcROffwM+eoWCMXc5VA
WadmxeV2eX9mGbWqWVflbsXV+TSEw9YA+RPEGEYrOPYzZbZTaNyyqFZ2WMhB2raTWK+w5
5cJKHAWntejaCUXhXc/pSHG0jdRdVZmH9dilYsVbhdv9b/0u9d9RfTECFekpilzTYeY9lzihoJo3R
qte35rjQr0fctuTAAxf9l16XcycBhZdmwscY5cJWEycvZe1FY0dBRhiksvdS1kR0AVdkFnCzSHd09p
xig2eFRsZBqdeUNusAsFfCpxKABBf5J4F3YtbJpgfmtLbI9imGB2cBJktlWXcWtm6Uqwcr5plz+xdA
prYjQ/dT9toigKdlVv0RqPdzhxugsVerxz8wBlf1V6AHU7aYBnTmP7a4po4V/HbX1qeFUGbwVsKko
4clZt5T9Tcf5vqjP6c1dxbifhdlJzJRqDdV90pwsieX52rQCffx97qnRSZq9uKGmuaPJvLI8WaxxwOF
RvbM9xZkm7bnpyoT7wcBpz6jOzcZR1NSe4ctN2ehp9c7R3nAs/eF15eAC6fsZ9LXN0ZDB1AmjXZ
p51gl5LaPF2BIO8asp2rEkIbJl3ZD56blx4MDNfb/N5CCeLcUV55hqBcjd6tAt9d098SwEefh1+mXKZ
Yeh77WgOZH972l2UZvl7y1MbaPR75kidauV8Fz4PbMh8ZTMUbnt8ySdjb+B9QhqFcOx9wAu2dm
N+wwF6fYJ/13G5X9+C3Gc9YpeCLlZUZS+Bh1JzZ0WBGkgPaVCAyj2ga02AnTLFbRqAkCc2bpS
ApqxDb8eA1wvkdZaA8QHlfP2A53DbXiGJtGZgYOmldFv9Y5GHPiG1ZbWGUUdvZ9GFhz0iaeGE

6DJqa8aEcScCbVuEKhp6bsCEEQwHdOSC3wIKfI2BYW/9XJ2QjGWJX22OwVsxYhyNAIECZEUlI
UbYZnOKUTysaJOJPjIVapCIXybQbD+HvRpubdyHHwwkdEuEjAJCfC6Bh28pW1CXWWTJXiSVB
lqFYNeSwFBRyxCQzkZZZUSPCzxIZ3ONgjHOaYSMOiama0uLQBplbR2Jqww7c8yF8QJxe9+Bp2
86eilSNGTTERVVjlp6ezxY5VAfe6xcLUW0fCRfdDssfKhiuTAdfU9l6SQ2fiFo7hbrfy5rkgimgKRvGwA
AgCl3zG5AdmtZMGP7d1Fb+VnFeC9ewE+GeOJhfEU4eZdkPTrMelNnBC/aex5pvyQSe/1sWRbsf
PhunwjGfxFx5gAAgAB5kG1Xcxl4WMzdEtillkedXBkYU77dlhmokTld0Fo5zp2eC9rMi+feR1teCPz
eglvqRbtevVxkwjfbB0oQAAGAB7HGyDcDBmWmJ6cZ9oIFiAcv1p5U53dBbrrURddSJtfToidjdvVC
9ld0NxJSPVeEFy4xbueTJ0bAj8fHx3SgAAgAB8dWuxbYts1WHBby5uHVffcL5vZE3tcfhwsKpscZBy
CTnKdGhzbC8odZB0yiO2dp52Gxbzd553SQkje2R58gAAgAB9nWrvaytzU2EAbPp0llclbrN07k1Kc
A91u0NicWl2lzlfcSB3hC7edAJ4dSOSdR15ZhcDdjR6SAIqel18cgBUf3J+yGoqaQN562BKavp6LVZ
+bNI6bUy3blV6tkLmb857FDj/cUJ7ii6bcpt8DiNyc8N8oBcSdPh9OQmseXZ+oAC2fs1/2mlgZxeAkF
+NaS6AO1XPayt/6EwcbMB/sEJkblB/kziZb9t/IS5UcUp/ryNOcoN/5hcbC+CANwnieK2AjgEJfKAs2i
aZWYHI17EZ5eGQIUlaaaFZ0tta0iEtKHPbOeEJTglboODui4CcAeDcCMhcVODTxcacuSDTQoMd/
6CRgFQfciA42fOY/CNuF3/ZiuMU1RNaEiK9krHafSJy0FDa56lxje3bUmH7S20bt+HQCL3cECGyB
cYcgmGDgovd2eDxQGMfWOBDGcGYqKURF1MZO2SXFOuZxiQfUo7aM2O1EDNaoKNVjdabD
mMDi1zbeGK/SLTb1SKMRcWcVOIWgpMduqFAwG+fQ+BLWYhgHxSQVyPglTvmVMLgJhY7Ulzg
JZcNT/EgKVfezXrgM5ivyt4gTFI6yAVgeFo6BM1gvxrgQacgyhwBgAAgFN5hWVMfOhY41vWfU5bt
FJsa1eg0jvfehhSj9afi9kFTWbfopm4ytFfwtpoCABf8RsORNIgMpufAbKgaFywAAAgAB7BGSDecF
fQlspem5hIVHcexBj6Eh3e3tmOT76e/BojzVUfHdq6isXfRZtOh/vfdhvbXNZfttXQbzgEt1awAAgAB8
VGPJdwNlbVqEd+dnTIFOeL1pL0gBeU5rEj6beehs+zUNepBu6Srpe0pwyR/dfBlyjXNnfS10IQcWfy
F3/gAAgAB9eGMOldrklndZttAVC+dpxucEeFd1Fv4T43eA1xWjTAeNNy3Cq2eaV0Th/Jen91rBN
6e6t26gdGfhF6XgAAgAB+b2Jeck5xuFkxc4hyulAWdK5zuUbydYR0sT26dl51tTRgd0B2xyp2eCZ3
1B+veQd42BOXek150AeWfQ98IAAAgAB/MWGwcEV3/ViOcaZ4ek+AcvF49EZtc+N5cD1LdNI5/T
QKddV6nyo8ds97Rh+Yd7h78xOxeRx8qQfec5+gQAlf/J/3GD/bnJ+V1fmb/N+Qk7icVx+MEXgcmV
+MjzUc3F+SzOudIJ+fyn+dY1+wx98doB/GxPDeA1/jwgbe2iANwBhf12AQWBRbNyEo1cybnKEDE
4tb+2De0U/cQODCjXlchyCtDNDczyCgSm0dFeCZh9YdViCaxPNdxICZwhMeryBvwCrft+AdF+ba3
OK91aAbRqJ5E2CbqSI2ESnb8OH8zvKcOaHLzLechKGIClucz2GGh81dEyFzRPUDkOE5gh0eieD
FQDqfnSAnI7majSRR1Xea+2PtUzybYaOKKqnbq6Mzjtdb9uLmDKJcRGKICKzckqJux8Yc2eJChP
adZGG+wiWeauEMgEffhuAwI0Ehv5SPIRShphViUuphi9Y00LXhbFcHjkhU9faDC7hRdirybihTJI2
Rv+hcFozw/WhmZrlgTQhV5xDQAAgHd7CVxSg5tYilO1g4NbWESjg2NeJ0Jpgx1g/TmOgu5j1TB9
guFmria/gxZpcRv7g6dsCQ/5hEpufgUKg+NzuAAAgAB8S1urgJtemIMlgMZg9kqqgORjUkIFgMlluTI
AgMNoJDBFgN1qkSaggSps7Rv4gb9vJxAXgnFXTAU9gpZ2UAAAgAB9ZIsHfgNkflKUfmFmbkotfq
5oYUGdfrZqYjfttBsaDAJfwVucCZ9f2hwYhvygAZyMRAygNVz/QVpgXV4pgAAgAB+XFpke65qV1I
CfDVr3kmufKltaEEwfNjvAdiVfQlwnS/lfVhyQCZWfc5zyhvqfnN1NhBPf2N2sQWhgGx6wgAAgAB/
KFnAeaBwKlFmekVxT0kaetdyc0Ctex5zmTgme3F0yC90e9h2AiYgfF53LBvbfQN4RhB2fhN5gAX4f
298tgAAgAB/v1kqd7V2H1DXeH92xkiVeTN3bkA3eZV4HzfDegJ43i8oeoB5rSXwexV6eRvNe7t7Qx
CZfO58PwZHfpJ+aQAAGACAAFiVdfI8LFBFduR8SkglD7d8bj+4eC18rDdYeK58/i7WeT99ZSW5ee
F91Bu6eot+TxCOe+h+/waJfDb/7QAAGACAAf9dHqCMU+ldXmB2Uldl2Bhz8lduCBWDbad22BQ
S50eAuBRiV3eLmBWxueeWqBhxDHevqBkQa/fSaBSwAaf9SAEVdfcyulRk8HdDiHeUbKdSiGtT6a
dbSGGDZjdkqFly4XdvKFOiU4d6uE9RuDeGSE0xDVeiuD0wbtfJOCewBcf2WAPlbDcgqOXk54cyK
NFUZHdB2L1z4ldLCKyDX/dVCJ3S3ldgCJHCUDdsSlfRtsd4WHzhDheX2FtQcTfBmDegCTfweAY
1PLjZpSLkv/jJdViEQ3i5pY3zwliRfCkJPnifBfcytliWtiuClIiUtl3hfCicVozAzAiV5sBgMsh1JyUQAAGJh8

ZVNAimZYKkuAia5bEEPHiPdd9TvLiD5g1jOih6ljtys3h0dmlSIThzppWxfTh6xr8gzyh1pu3QNxheF0
7AAAgBR9cVK3h41d70sLhxJgbUNmhpRi6Tt6hgVIYzNjhZdn3isNhVVqVyIBhVpsvhfghcdu/A0ehZ
VxmQOuhJ53TAAAgAB+XFIrhRpjjEqNhM1lqEL3hHpnxTsdhAxp4TMbg7lr/ircg4xuGyHqg6BwHBf
phBBx9A1EhAt0OQPjg4V5XAAAgAB/KVGqgt9plUoUgr1q3EKlpgJsmDq9gkNuUjL0ggglwDyqlgFx
zSHOghNzaxftgoF05A1sgqp22gQjgoJ7OAAAgAB/zIEkgOxur0mRgOtwD0llgOBxbDpLgK1yuTJtgl
50DSpcglT1ZiGjgLZ2rBfrgRt32g2dgWh5kgSBgYt87QAAgACAAFCIfyN0XEkXf0V1RUGXf1p2LTnk
f0F3DDIWfzp39Coaf0x46SF7f3951hf0f9x6vQ3JgE98NgTVgLN+agAAgACAAFAifYp6lkiWfct6i0E
Yff169Tlyffd7ZzG0fgJ76CnPfiR8eiFLfI99EBfdrZ9rg3qf1V+rgUdf/V/vgAAgACAAE+bfCt/40gDfIF/3
0CAfMV/3jjnfMt/6DE9fOCABSIwfQyAOiELfU2AeRfFfaKAYw4BfnSA9QVWf1CA8QAAgACAAE8M
ev6FuEdxe2CFST/te6+E3zhie72EgTDKe9qEOikUfA2EECDNfFWD+BferfKqD6w4Sfa+C9QWGfs
GB/AAAgACAAE5/egCLlkbtemuKsj9yesCJ1DfyetWJDDbpevmIYCjHezOH1yCZe4CHaBeWe9aG
kQ4gfQyEoQWufkqC2wAWf9yAD0qPIIFR8kOqktVYzzEkUZyZDV0j+pcGy3qjsZfZyYOjfBirB1mjZ
Zl0hOEjfhOuAoJi/FsrAG8iQRztAAAgLV9lkoskXZXpENKkBxapzxrsdpdzUujZdgmS25jJdjhiXyi95m
ax1li41pNxOpi9xrgpligJvcglKh512GQAAgD9+cknljrxdl0L1jZtfyDwjJlBiaTTxi3hk+i2KipxnhyXWif1
qEB1jibVsgxPlifZuyAp/iFByHgJPhmJ4MQAAgAB/NEIzjGhifkKMi3NkzDvDioBnFzSfiZZpUS1LiNFr
iSWwiEBtvB1ZiAFvzhPgiD5xsQqvhtZ0rgKMhVB6BQAAGAB/3Ej9ikBn10ltiXppyjtjiLNruzRMh+Ztmi
0lhzhveCWDhrVxUh1Hhn1zCBPzhrB0kArhhYN3PQLShFR7qAAAgACAAEibiF9tJ0HGh8BuxDr6h
x5wXDPshm5x0yy2hdhzTCVfHwI0xx0nhTN2KRP/hVB3bwsbhE553wM2g2J9JgAAgACAAEg2hr
VykkFIhjdzvzqdhbR06DOWhR519SxshKB3CSUOhEJ4lx0JhA55NBQJhBd6OQtPg0F8RwORgo5
+cgAAgACAAEFghUF4FUD0hOB4yjothHZ5fTMvg/N6lywTg4V60yTlgzV7kBzfgwJ8ThQFgvt9EQt
2glN+dQPcgDV/mgAAgACAAEdLhAR9IUBsg7I95zmdg2N+NzKqgul+eyufgoV+zSRqgjt/MBYfggl/m
xPvgfWAFaUgPgX6AdwQYgTSApgAAgACAAEbHgvqDLj/hgryDHjkQgnGDDDIpgf6C7ystgZ+C4iQ
NgVuC6hxfgSiDABPWgQuC4wuhgMWCOgRKgKqBjwAAgACAAEZEgiKl1z9kgeilVziZgaGH2TG8
gTOHWCrOgNqG6yO/gJqGmRwqgGeGUhPCgEWFOWuxgCuDswR0gDeCUgAAgACAAEDom9F
RpDramdlVCDTMI+ZYai5UlfZbyCeSIFffICBrxNibhhjkm9ImA9xkfpoxQeJjk9tfQBpipR1EgAAgNB
+rkCqmPFXCTqal0FaCjSPIY1dDC4nk71gGSd3kjVjHCBikQ5mERh3kGho6Q+nj+JryAfVjHNwMg
DAiTZ3LQAAgGZ/X0CLlj5cTjp1IMxe9TRmk1Nhn14BkbJkUCdbkE1m+yBYjzlpmxihjpnslA/Wjf9us
AgXitJyzQENiAJ5BQAAGAp/+kBCk/9hcTonkrtjxjQVkwXmHi28j+dogicojpRq3yA+jYVtMRiMjOFvW
w/7jEhxiQhQiWd1TQFQhvd6nwAAgACAAD//kfFmjDndkNtojjPlj7ZqlC14jk1spCbxjQ5urSAdjAlwrhi
li2NygRAair90VwiKiCJ3ywGdhgF8DgAAgACAAD/AkCVrmTmWjzhtTjN6jjpvAy0sjO9wryavi8hyWB
/sitNz/hh0i1h1hBAziWZ3HQjNhvl6OQIHhRN9WQAAGACAAD+Ejo9wxDIUjcZyDzM1jOhzXCzqi7d0
oyZ0iqd16h/AicN3NBhiiRN4bxBJiDV5zkwJhfZ8VwJmhEN+eQAAGACAAD8xjTp2Bjj6jlt24DLWi8Z
3viyQiq14oyYkial5jB+AiNB6fBg9iBp7axBOhyJ8jgk1hRJ+QAK1g45/egAAgACAAD6+jCh7RDh7i4x
7xTJOith8SiwRicR80yWziMt9Yx8ih/Z9/Bf9hzp+mxA9hih/WwlQhEmAAwL0gvGAYQAAGACAAD4
+i0iApzf1iraAzDHGiguA9SuTiPyBliVDiAiBWR7EhzWBnBe7hnKB6BAnhU6B1Qljg5uBjgMogmyBK
wAAgACAAD3CipGGKjd9igOF3jFTiVuFmisoiFCFayTjh2CFSB50hpCFNBeDhceE9BAVhJmD5gl0
gwqC2QNTgfyB0wAAgAmAAJJBv7tMh4U+WhZQYngHxGxUP2rfXulYF12XYXVb5FA+ZBJfnUKI
ZqBjKjQvaPZmdCS7ao9pWBInb0lrDwKxe3RwEZBIUItVYoOQVTBYf3a+WAFbn2nCwupewly1Xdl
h30+LYMtk8UH7Y5xn4DPGZhdqmiR/Z6dtARI0bURucgMGeuNzYo6fTVheAlIVUKJgZnWLU+Ziz2
jGVzNIS1vqWn5nwE7pXcBqKkF+YM1sfTNpY2tuqiRKZO9wlhl/a4BxxwNsemR2mozdSPxmUICc
TK1oAnRaUFRpumfMU/JrlVsiV4VtbU5NWwJvPkEHXj5w/TMTYPZyniQbYn90FBJJafd1CwOTefV
5a4teRQhubX9QSQ1vf3NCTQdwlmBkUPJxzlpVMhzCE27WH10QECaW+Z1azLJXrl2hiP7YGt3i

hJmaKR4UgPjeW1744oWQYx2VX4nRdh2x3l7Shh3PmYHTkR32FmzUIZ4dk0vVkB5F0A1WdZ5u
DKOXMI6VyP0XsR69BKoZ497lQRaeKV+AojmPmF+OX0aQup+CHFTR2d93GVDS9F901kTUBx
90kyzVDd92T/dV/h97TJcWwp+EiPvXVh+SxLmZqF+zQTFd+9/2lfMO36GDXwdQDSFOHB0ROG
EaGSHSYCDv1h5TfyDIUw7UkOCkj+IViyCHTItWWGByyPsXAYBohMcZc2B2QUhd1SBblbQOOG
NtHsuPbOMPm+VQn2Kz2PJR0WJkVfdS+iIYkvBUFWHRT8zVGWGTzIBV7+FjCPsWteFBhNKZQ
2EggVvdtGCwIXhNniVPnpPO2WTN27KQEuRNWMDRTePYVdPSfyNnktRToqL8z7nUr+KfTHaVv
CJSiPsWcelbBNyZGaGzAWxdmKD2YUBNEqcnmJOVmaFW4fPI+XjGKMq2uVGlbYSE+Sv0r0T
PuQgz6mUVGOijG5VPSM6SPsWOOLsROTY9qItQXodgWD/omaXT9M8H0nX0VQtHC4YUpUe2
QvY3FYRIecZaRcBkrzZ+VftD3caiNjOzATbDxmhCEjbbpZg9OcwprUgGbfUlxW4fIWDIVU3ucWrN
YZW9zXSdbe2MkX6penFbFYjZht0pJZMhkyD1bZ0NnvC+4aXpqqSD2auhs7w9icSVulgHofMd0h4
YdU4Fdg3oyVmhf525LWUdiUGlyXCRk0IYEXwNnTkmxYeJpwjznZJVsjC9nZuluZiDOaFlwYQ90b
3xxywlsfFV3Y4SZT0Rld3jsUo9nMW1AVc9o82FUWPxq21VPXCNSwUkjXztupDx+Yh1weC8fZlty
MiCsZh9zuw+Dbgp07gJme/J54lNES3htPXfDTw5uX2xEUplvh2CCVhBw1FSIWXhyI0ifXMdzdTwc
X9Z0vC7gYmB19SCWZDV3EQ+kbMh4FwKxe3N8EoISSBZ01HarS+t1XmtKT7V17l+wU2t2pFP7
Vw53YEgcWpF4IjvAXcx45S6sYHd5qCCWYqJ6Yw/qa7d7RAMleq99/YDzRP98bHWrsQt8WWps
TQ18S173UP98ZVNIVNI8iUenWl18ujtuW/N8+C6AXr59RiCYYUZ9oxAqast+XwOOef1/p3/pQjKD9
nS8RmqDRGmZSpmCl15FTryCGFLUUsSBqEc3VqKBSjsgWjCBBY5XXRyA5CCaYAYa5RBhafqB
JwPoeWWBFH7+P7iLTXPeRAuKBGjLSFSIw12TTJmHtlJCUMSGvEbGVMWF2DrRWHeFGy4uW
4eEjSCcXumEOBCQaUCDIwQ0eOWCSH4jPX6SjXMRQeaQuWgPRkWO61zySqaNUFG8Tu2Ly
0ZeUwyKYjqKVuGJLC4KWhSINSCdXemHjhc3aJ6FsQR0eHiDA31XO4aZr3JcQASXVWdyRHiV
AFxqSPKS0IFLTVKQv0YHUY2OzTpOVYGNHS3sWNSLwCCfXRSKpxDYaBeHcgSqeB6DJ4DXY
uJNQHUPZJVQ72lQZkhUoV2EaBpYXVGoafNcEUWta9Zftjk4bcFjOSv/b5pmsg2ScP1pYgy+dm1r
tQChfvBysn89XgtVNXOvYCpYPGgqYkhhR1yMZHteYlDdZq5hekUOaN5kjDjCawFnhyu0bPFqVx1
1bk5sywzYdKRu2gDmfnt1kn3FWY9c93JvXBZfW2cgXplhw1utYR9kR1AoY6JmykSBZhtpSjhaHF
rvStxanluEx1ca/NwHQzvcxRx7wEjfhV4GHxuVYJkjXFLWGdmUGYvW0NoGlrnXhNqDk+IYNhsA0
QEY4lt+Tf/Zg5v5Ss2aDFxuR1GaedzVQ0Dcbd08gFXfbx6SXs5Ud5r+nA/VQhtK2VLWCpuY1ooW
z1vxU7uXj9xLUONYSdymTeqY9hz/ysDZhd1Vh07aB52iw0ocln3/QGefUV8PHorTpZzQ29DUfzp5
WRmVVV0jVImWKN1Xk5QW9x2OUMTXvZ3HjdVYdR4BirXZDN47h1EZpt5xw1xb3Z7FQIPflV9+
HknS5h6kG5bTy56m2ObUrx6q1i6VkB6503DWa17MUKnXPh7jDcLX/979yqyYn58cB1NZU188g2
zbox9+wJ2e9h/engvSOiB0m19TKSBQmLYUFiAuVgUVAqAY007V6KAIEI+WxZ/9TbDXkv/5SqNY
ON/9B1UZCCAIA3sbb+AhwLOe0OAxxdPRo2l4WyoSl6Hw2lTTiiGrFdrUfqFz0ywVbKFCUHUWU
aEYtZ6XJiD3SpoX1mDhx1YYw2DYg4bbQmCxAmyesWB3HZ9RHOP4GvkSFOOO2FfTC6MnV
bRUBmLOEwxU+yJ70FzV5ylyTY4Ww+H1CpHXfGHGx1bYhyGqA5CbGyEsgNXelqCQnW8Qpu
WxWs4RomUnGDISnOSeVZPTnaQh0vHUmKOt0EhViqNETYAWbqLqiorXLUKkh1dYVOJiA5ja+
mGTwOMegGCZnfwaKdNemzQagNRGWG8a2JUulajbORYZ0t6bmtcDEAxb/hfpjRqcZBjJCfZcyF
mbRn8dHNpSQpqeYBsNQAAGAB0K3aZY/9VE2umZcFYEmC+Z4VbE1XLaWZeJUrla0NhNz+nb
RtkRzQlbuZnRSegclxqHRnxcfrsIAqKd9FvOwAAgAB2iHvcX7pcZmqUYd1ezV/WY/5hOFUJZitjv0
oqaFNmRj8tanJozDOxbG5rTCdubihtshnob8VvxwqmdlZyMAAfy6x4vHQ5W91jkmmbXldY18IYM1
nO1RdYzxpOkmgZaJrPT7BZ/htRDNIaiFvRCdDa/NxMRngbd1y3gq+dQ11EwBhf1x6qXMwWFpqq
WiyWxlr615AXdNtNFO1YIBuqkkXYyBwJz5YZapxrTmcZ/5zMSccael0qBnhbDB19Qrmc+Z4AACjf
ux8Y3l5VTRxo2fNWCdyXV1vWxNzHVMCXfd0CEiEYmp1AD3mY4V2BzLQZgJ3FSb4aAt4JBnzar
l5HQsyctx6+wESfjF99HFPUI4oGb6VXR4yVy1WI54+FJiW6R5VkgAXqh5xj2CYZB6SzKMZDV64

ybaZlt7ixoEaXR8NAt2cfV9oQF3fYZ/UnBtT7x/l2YsUv9/Klv8Vj1+xlHDWXt+mkd+XKd+hT0eX7V+j
DJJYoF+sSa6ZMd+9hoQaFJ/TwuwcSp/9gHOfPSAfW+XTXqGaGVeUM2FcFs7VBuEglEeV3WD1
Ub3Wr2DRDy3XeqC1jIDYNmCjyaYY0OCdBoXZ0yCggvfcHiCBAIWfHmBaW7HS3uNLGSbTtWLs
VqIUiyKP1CGVZyJEUZ6WPYlBDxZXEOHIDHDX1SGbSZ4YeKF9RobZmiFqAwHb+CDywJUfBC
Bk24DSbyT1WPuTR2R2VnyUHuP51AGVACOM0YSV3aMpzWJWtSLSzGNXgOKLiZdYLSJXBofZ
aqIUQwob2CFRgKle7mBtm8dboVNpmShb5dRK1o0cKxUtU/QcdhYVUVbcwdb8DrDdDpfgY+pdXx
i/yO8dsZmSBZoeCtpHwhSfEds0gAAgAB16G30agdU6WOia3pX1VlfbO9axk8ZbnZd0UTDb/hg4D
pPcXRj8S9ccuZm9iOYdEBp1hZxddJsTwh4eq1vuWAAgAB3+GzkZflb1WK5Z8NeM1ibaY1gl050a1
pjJEQ9bSJlsjnobuFoQy8XclFq0SN4cextSBZ5c8NvZQiZeUZykQAAgAB5xmvxYltilGHqZG1kbFfvZ
ntmSk3kaIvOWEPlaoHqbDmObH5shS7bbkhunCNcb8lwoRaAcfhYXgi2eA51VgAAgAB7WWsQXv1
pUGEjYVBqn1dEY59r9k1TZefTgkNSaBpvGTkzakJwuy6ebDVyXiNBbcxz9haNcF91WQjidvN4JgA
AgAB8tGo3W/Vv/GBXXnhwylaHYPVxn0yuY2pyqELJZdNzWdJLaCl06i5aakN2HyMna/R3VhaqbvF
4bgkwDex67AAzf6p99mlmWTF2r1+ZW9128VXdXoV3OkwaYSh3vEJOY754VDhuZj95BS4daH55
zCMPakl6oxbEbbJ7cQl3dQd9VwCXfW/MWiUvrd9X17ZWYF9D1UvXEZ8ykuEXw58ykHSYcl85D
gQZG59Hy3fZtF9eyL1aLp99RbWbJl+ewmzdD9/eADsfngAQGfGVI6D714QV2KDGIryWjSCUUri
XROB10FPX+eBfDesYqiBSS2cZS+BPiLVZz6BXbga56Bnwnic5KBWwE0ffeA0Wb8Up6KdF1RV
XmJIVPBWFKH3EpLW0OG50DUXiyGFzdQYQaFdi1eY6+FBiK3ZeyEzxbnasWEkwoKcv2C/gFyfY
+A+mZAUOeQ3VyoU8uPEVMrVqyNUUnNWa+L3kBTXKuKlzcCX5qJhi0pYmOlSiKdZNOIjhsahC
HCworcoCEWgGlftmBHWZhdHJNx1yxdQBRXIMPdZdU9ElkdmNYjj+fdzxcITWteCdfrcsoeTdjHR
+4enFmWBLOfDNpIQZZfv9txgAAgAB3z2VecBpUqFvQcQtXs1JQcgNavUjBcyRdzj8adEpg4DVKd
XZj8SrpdrBm8h+id/ZpyhLmefdsPAaLfXRwmgAAgAB5j2R2bD9bSlSEbYFdZFGhbshgUEgvcCpi6D
6lcZBlgDTycvloGiyydFxqxx+Pda5tHxL8eAFvPAA3fBpzXQAAGAB7GGOhaM5ht1pLaIRjvVEEa9xl
x0erbXhn6T47bxRqDzSjck5sOSqAcjVuWR99c5hwXhMPdkxyHwbeeu12DQAAgAB8bmLUZaloFI
mWZ2dpo1BoaSZrNEcmavds3j3PbMRukDRSbotwSSpNcDZx+h9scatzlXmndMd1AgcSedx4tAAA
gAB9kmlVYtFuZljeZLpvgU+6ZqRwnUaOaKRxyz1Rap9zBDPzbl90TCORblx1IR9ab+N22BNNc2h3
/AdmeN17JwAAgAB+e2FaYDR0y1gyYkZ1X08eZFp19kYHZoV2pDzhaKI3ZDOeasF4OSnbbK55G
h9Jbkl6BBNwcjZ65Qezd/59SwAAgAB/Q2CfXdh7OVeEYAt7Pk5/Yj97SUV8Zlx7ezxvZtN7xDNIaQ
x8KSmjxh8pR81bMx9NXLcSh91Qf0dzt/LwBGf4qAH1/pW8aBjVbPXguBEk3PYE+AnUTIYrOA
WzvzZRGANTLrZ2OAMCImaY2ASx8ca2yAiBOecDWA1ggpdpKA3gCSfwqAYl8yWeuH11YgXD2
G5E0rXo6F+kRZYQKFQjuAY3OErDKTZdyEPiksaCOD+B8DajeD4BOsb2ODhQhWdgCCUwDSfp
2Ajl6CWESOCFWCWqaMoEygXQaLQEPiX4qKETsfYgyJCDJKZlilLyj8ZuqHiR7uaTWHIRO4brOF
wQh7dYaDiwElfkKAsl2yeoRN3FTReqBRcUv7esZVBUMleyZYnTn4e59cLzCtfDlftia7fRjJHxvDfjx
mTg+Pf9xpRAScgWZuzgAAgAB5fFzPd1UYFQKdtZXdEtSd1daiUJ+eAddpDmleMZgvjBceZpj1Ca
Mepdm1Ru8e8hpgg+2fbpsSgTZf+hxjgAAgAB691wlcq9asINac3RdSEq6dD9f4ElAdSxiiDkkdiZIMT
AUdzBn1yZjeFBqbxu3eYls4g/Xe9xvNAUOfpp0PQAAgAB8Q1tJb2hg01KzcG5i9koscXdlHUGJcpp
nWTjGc8Rply/QdPlr1iY8djluARuwd35wCA/0ejtyAAU9fXZ20gAAgAB9Y1qRbHJm4FIQbapolkmfbu
VqT0EQcDlSHdhkcZFt7i+Jcu5vwyYSdExxhBupdZxzJxAVeMh0ywV4fG55MAAAgAB+VlnpacRs3l
FsayRuLEkBbIhve0CGbghw0jfbz4pyMi81cQ1zmyXfcod0+xufc+N2ShBCd3p3qwXSe3V7XAAAgA
B/D1IFZ01y9IDTaNVzxeh0amF0IEALbAp1czeNbbJ2Xy7pb1l3XCWxcO94WhuWcl15VxBrdld6eA
Ykepx9QAAAgAB/rlifZRJ5HIA2Zrx5Y0fjaGh5rD+MajF6FTcka/l6kS6abb17JCWAb217wxuKcPx8b
hCMdVR9TgZqed5+7gAAgACAAf+Yx9/Mk+TZnt+/0dCZpl+0j8BaHZ+zDaxalR+3S5DbC9/CiVlbf
h/Trt3b7d/pxCmdGqAFgakeTiAbgACf/2AAVdaYWKFRE70YyyEoEarZPaEAj5+ZuODjDZFaNKD

Mi3xasKC+yUTbKKC4RtkbpmC6hC7c5+ChgbVeKmBvQBff4uAL1a8X9uLQ05IYbOKLUYsY4qJH
T4QZYWINDXrZ4SHbS2taYWG0CTna3qGWhtVbauGChDMcvWEjgb9eDGC1AB9fyyAVVUFgMR
N3kz3gHJRZkTzgCtU8DzAgBZYjDRigCZcICu9gGlfpYJcgQhjDBfWgiRmLwyygxNpkwMTg4Nv6gA
AgAB691Q8fNdUCEXJfNdXG0Rhfn9aMTxKfRRdXTQGfWRghCt/fdpjoil+fphmpRfgf7VpcAzkgQls
gwnZghFymAAAgAB8NFOQeVRaB0uyeZpcqUPgeedfUDvgeLZiDjO0ettkyitGe4FngSljfFxqlhfpfXx
slQ0Qfz9vWAOXgM11MQAAgAB9SILhdjJf4EsbdriGUNhd0FkVzt1d+JmsjNheJRpDCsNeV1rYil
HelJtmxfve3hvpQ02fbFyDgPMf7J3iAAAgAB+PFI/c2JlpkqLdBhneULjdNBpUTsJdZ9rRTMLdnttOir
Rd2dvLyHoeHRxAxfzeaByrA1ffE50wgQOfR5owAAgAB/A1GucNtrW0n6cbdszUJWcpZuQzqOc49
vxzKIdJfXUSqHdaBy3yG9dsR0Vxfyd/11sw2Sew93iARufb97jgAAgAB/kIEfbohXKEIzb4tylkHYcJBz
IDofca90MzJKctZ1TypEdAV2diGXdT93kxfydpJ4pA3Aefl6OgTEfOt9OgAAgACAAFCNbHN3B0jnb
ZR3gEFVbrh4ATmsb/V4pDHqcTh5Vin8coN6FyFsc9F62xfsdUp7oA3neQF87AUOfDF+twAAgACA
AE/8aqR82UhTa9Z85kDCbQh8+jkrblI9MzF+b7B9fymqcQ993yE5cnF+ShfgdBt+ww4GeCB/cQVM
e46ADwAAgACAAE9saQ6CtUfEakiCWEA3a4SCAjjxbOOB0DEYbkqBtSldb7qBtSElcS6BxxfTcxK
B8Q4gd1uBqQWAewGBOgAAgACAAE7IZ66li0dFaO+Huz/CajKG9ThKa56GVTDDBRSF0SkbbpS
FcCDfcBiFKhflcjSE3A41dreDggWseoyCNAADf/uAAkwqhx9NyET9hjNRYT3VhVtU+DZdhOBYIS6t
hJhcKSaqhJhfrh3ahQtjDRPMhiVmlwoahd9qMQGvhWBxOgAAgAB8S0t9g2pTmkRlgs1WyT1Ugk
JZ+TX4ggddMi5jgfhgYyZ7giNjhh3LgqxmihPng71pTwpWg+9tDAH+g/pz1AAAgAB9UUrsgA9ZSkP
of7ZcFDzqf2xe4DWgf21hsy4ff5FkgSZPf+RnRB29glJp7BP+gYxsXgqKgjtyvyJDgsB2NQAAGAB+
NkpafRFe3UNkfPdhSDx2fOtjtDU+fRxmLC3UfWdonyYgfdZrCh2sfodtTxQRf5JvWwq4gMFybQJ/g
a94RwAAgAB+/knMemhkX0LkenxmbTwEep5ofDTcev1qlC2He25sqiXse/luuB2VfLtnwRQgfdNyT
Qrnf251CwLGgLV6lgAAgAB/nklQeANp1UJneEBriDuJelptOzRveRJu6C0realwISWpelNyQh1yeyR
z0RQpfEx1Ogsifjx3uAMrf8h70QAAGACAAEjVdddVYEHYdjpwjsbdqpx4zQNd1ZzlyzZeBB0aCVse
Nh1sR1Rebh26xQyevp4EAtWfTJ6UAOGfvl9SQAAGACAAEHUc+p0/kF0dGt1xzqjdPd2kTOhdcF3
Zyx+dpZ4RSUnd3d5LR0oeGR6EhQxect68guAfEZ8vwPTfkN+lwAAgACAAEFtckJ6kkDqctR6+Do
Uc3F7YDMhdEx70CwRdTN8TCTSDiZ81xzydx99ZhQleLh9+wuge3J+/gQTfaZ/xgAAgACAAEdQc
NGAOEBicW2APDmLchOAQjKncvyATCupc/CAZSR/dPKAkhy+dfaAyhQXd8iBCAu5erqA+ARlfr6
AzQAAGACAAEbSb5aF5j/pcDiFejkYcOSFETJBcdiEsytSctmEaSQ6c+iENxyRdPeEGBQLdwCDp
QvPeiCCngR1fK2BqQAAGACAAEM6jcdNeDz3jDhRLDaxishU2i/4id1Yfyj5iTiCfyGWiPjfnhlWiTli+
RARicVmPAfSiEtq+AB3hv5yoQAAGAB9dEKripS9jx2iQhWSzZBh+NZni+khzBc6yi/hrxgKiF2hpljV
hIXhutmXRA5h2lpUQgWhnNtvgnDhN1BwAAgAB+Skl+hw9YXjwXhhBbWTXwhS1eUS9dhLhhP
yilhHVkiSFXhHNm9BIWhNFpphBchUtsSwhShNRwaQEYhHN3lQAAGAB/BUHLhDBdsDumg3Rg
VDWEgtBi9i8DgotlkShGgmpoISEygnlqohlQguJs9hB4g29vNQIGg2xy9wFZg2t49gAAgAB/p0FTgal
i9Ds2gR1lQzUdgKdnkC6rgl5p0SgBgJBsCiEGgLVuNRIDgSdwLhCQgctyEwi8gih1gQGlgNh6mAA
AgACAAEDxf2FoMDrPfwJqKjS1frhsHy5LfsL+Sevfu9vzSDIfzJxmhl6lZQhCjgF905wj9gQJ4FgIPg
Y98EgAAgACAAECNfVhtgjpvr9vDzRYfPpwly32fTByCCdlfXlZdyCRfdh04xkKflV2PBCzfyN3pAk2
gAJ6ewJugMR9WwAAgACAAEAfe5Ry5ToAe3d0ADPqe291GC2Re8F2licNfCR3LiBMfJl4PRjifRx
5RRC3fgt6bQljfyF8pQK+gBN+gQAAGACAAD+qehR4QDI7egt5ATNcehV5vi0Pend6Zyaceup7FR/
ve257yRime/V8fhCrRF9XgmDflp+owL/f3t/iQAAGACAAD8teMp9tDj1eM1+GjLReON+fCyReVB+x
CYtec9/Ex+Vel5/bBhpeup/yhCafDiAKwmbfa6AZAM1fviAcAAAgACAAD6vd7GDPDh7d7yDNTJcd
9mDLSwmeFGDEiXQeNqDAR9leXOC/hg2egSDBBCNe4KCgAmvfR6B2wNifouBMQAAGACAAD
m/IOFM8TRnkvZQri8lkSNUZykmj6VYICLrjoZbxxw3jeNfWBSQjfBitQykjPlmhgW6ioZr4AAAh2B0d
gAAgAB+hjIRkaVSljP+kAFVhy6qjnBY7SjljSfCXCLEjChftRwpi6Fi8RSji65mAgzYirdphQYGiMNuk

QAAhqt2VAAAgAB/LzkjJnRXSzPNjRtaXS53i9NdbiizishghCKeigFjhxwaiZJmcxSziaBpNQ0FiLtsaQ
ZlhzdxJwAGhgN3+gAAgAB/wzjKi71cXDNTiqVfIC4ViZph5ShiiL1ksCJkiBJnaRv/h6pqCxS6h7lscw0
qhvtvQAaChd9zogBMhQN5lgAAgACAADhoiVNhZzMOiG5j2y27h5RmUSgUhuNoySlohlZrMhvchf
xthBS3hgtvlw1LhXFyCQa+hKp2FgCchBZ7BQAAGACAADhAhxVmbzLPhmpokS1phcZqsyfFhUBs
xyHhhNluzRulhJtwxBShhKVyjA1mhB10xQcEg454gwELgzJ8TgAAgACAADgHhSprjDKNhKZtRC0
hhClu/Sd+g8lwqiGhg4RyTtxtg2Fz6BSNg2h1Zw19gvZ3aAdDgpd6ogFugmp9bQAAGACAADetg5
RwujltgyhyBCy9gsRzTycggoB0myFMglV13xsugkZ3HhRIgkp4UQ2FgfR6FwdzgcB8jAHAgb1+bAA
AgACAADc4gkl13zGpge521SwugZp3zSaagWV4vCDUgUh5phrlgUV6jBQhgUR7bg14gRB84QeR
gQV+UAlCgSp/UgAAgACAADavgS57JDEbgN97xiufgJd8aCYVgGx8+iBbgFI9iRpigF5+FxPbgFI+p
Q1mgEx/XweogGV/3Al4gKuAGwAAgACAADYhgDmAjDCYf/WAwismf7eA+yWkf5WBMB/2f4qBY
xoMf5eBlxOhf5SBrg1Xf6iBcwe7f9+BjWJmgEKAwwAAgACAAAAA//8AAP//AAD//wAAbWZ0MgAA
AAAEAwSAAAEAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAQAAAQAAAgAAA
QsCFgMjBDEFQAZQB2ElcQmACo4LnAyqDbkOyQ/ZEOkR+RMKFBovKhY6F0kYVhlgmwbdbxy
CHY0elx+hIKohsyK7I8lkyCXMJs4n0CjRKdAqzivMLMktxi7EL8lwwDG+MroztzS2Nbc2uzfAOMY5z
jrYO+I87T35PwVAEEEEaQiVDL0Q4RUJGS0dTSFxJZEptS3VMfU2ETojPgVB/UX5SfFN7VHpvVeV
Z4V3hYeFI3Wndbd1x2XXNecF9tYGphaGJIY2NkYVWfZl5nXGhbaVpqWWtYbFNtSm5CbzpwMnE
qciNzG3QVdQ52CHcDd/54+Xn0evB77Xzqfeh+5n/kgOSB44Lkg+WE5oXohuuH7ojyifeK/lwBjQeO
Do8lkAKQ/JH3kvOT8JTtleqW6ZfomOiZ6Zrrm+2c8J31nvqgAKEHog6jF6QgpSqMnadBqE6pXKpr
q3usi62drq+vw7DXseyzArQZtTG2QLdDuEe5TLpSu1m8Yb1qvnS/f8CLwZnCp8O3xMjF28buyAHJ
Fcoqyz/MT81azmXPb9B40YHSidOP1JXVmtae16LYpNmm2qfbqNyn3abepd+j4KHhnuKa45bkkeV
55l3nQugm6Qrp7urR67Tsl+167lzpVpAg8QHx4vLD86P0g/Vi9kH3IPf/+N/5v/qg+4H8Zf1Ji//Fv//AA
AA7gHcAssDugSnBZUGgQdwCF8JUApDCzYMKQ0cDhAPBA/5EO8R5hLeE9cU0RXLFr8XtBipG
Z8alRuLHIEdeB5vH2YgXSFTlkjgPCQxJSUmGScOKAlo+CnuKuUr3SzWLcwuvy+0MKwxpjKiM6E
0oTWkNqg3rTizObs6vzvDPMc9zD7QP9RA10HbQt9D4kTmRepG7kfsOdJ40rgS9xM2k3YTtZP1
VDUUDRS1VPWVNdV11bXV9ZY11nYWtlb21zdXeBe5F/nYOxh8GL2Y/tk/WX7Zvpn+Wj4aflq+mv
7bP1uAG8DcAdx3lRcxd0HnUldi53N3hBeUx6WhtlfHN9gn6Sf6OAtYHlgt2D8oUlhiCHMYg7iUaK
U4tgjG6Nfo6Oj5+QspHFktmT75UGlh2XNphQmWuahJucnLWdzp7poAWhlqJAo1+kf6WhpsOn5qk
Kqi+rVax8raSuza/wsQeyHrM2tE+1abaDt564urnXuvS8E70yviO/dcCYwb3C48QLxTTGX8elyKrJzc
rwzBPNNs5Yz3rQmtG50tbT8dUJ1iDXNdhH2VjaZ9tz3H/diN6Q35bgm+Ge4prjeuRa5TjmFebw58v
opOI96lTrKuv/7NPtp+5570rwGvDq8bjyhvNT9CD06/W29oH3SvgU+N35pvpw+zn8A/zN/Zj+ZP8x//
8AAADpAdMCvQOoBJIFewZIB04lOQkkCg8K+wvnDNMNvw6uD54QkBGGEn4TeBRzFW8Waxd
oGGQZXhpZG1lcSh1CHjofMSAnlR0iEyMJl/8k9SxRJuEn1yjNKcMquCutLKEtlC6HL3kwaTFYMkcz
NDQhNQ01+TbIN9E4vTmoOpQ7gTxuPVs+Sj85QCIBGEllQvhD6ETZRctGvUewSKNJl0qLS39Md
E1pTi5PUIBGUTtSMVMnVB1VFFYLVwNX/Fj0We5a6FviXN5d2l7XX9Rg0mHRYtBj0GTQZdBm0
WfTaNVp2Grba99s423obu1v83D5cf9zBnQNdRR2HHckeC15NXo+e0d8UX1Yfl9/ZoBsgXOCe4O
ChlqFkYaZ6Glqomyirqlw4zMjdaO4l/pkO+R9pL9IAWVDpYXlyGYLJk4mkWbU5xhnXGegZ+SoK
Sht6LLo+Ck9qYlpxqoLqlCqlirbqyFrZ6ut6/SsO6yCrMotEe1aLaKt6641Ln7uyW8UL1+vqq/2MElwjr
DbsSixdjHDshEyXnKrsvizRPOQ89v0JnRwNLI1AfVKNZG12LYfNmV2qzbvdy73bfest+t4Kbhn+KW
443kguV35mrnXehP6UDqMOsf7A3s+u3m7tLvfvCn8ZDyefNh9En1L/YW9vv34fjG+az6kft3/F39Q/4
r/xT//wAAAOEBwgKIA4gEbAVRBjUHGQf7CN0JvwqhC4MMZw1LDi8PFA/4ENwRvxKiE4UUaBVK
FiwXDxfxGNQZthqZG3wcXx1CHiUfCB/rIM4hsSKUI3ckWiU9JiAnAyflKMgpqSqLK2wsTS0tLgwu6
y/JMKcxhDjiMz80HDT5NdY2sjePOGw5STonOwQ74jy/PZ0+ez9YQDZBFEHxQs9DrESKRWDGR

EchR/5l20m4SpVLc0xQTS1OC07pT8dQpIGEUmNTQ1QjVQNV5FbFV6dYilltWIFbNIwbXQFd6F7
PX7Zgn2GHYnBjWmREZS5mGWcEZ+9o22nGarJrnmyJbXVvYW9NcDhxJHlPcvpz5XTQdbt2pne
ReHx5aHpUe0B8LH0afgd+9n/lgNWBxoK4g6uEolWVhoyHhYh/iXqKelt3jHeNeY58j4GQh5GPkpe
ToZSSlbiWxZfUmOKZ8psDnBSdJZ43n0qgXKFvooKjlaSopbqmzaqqPKqBKsXrCmtO65Nr1+wcbG
DspWzp7S6tc6247f5uRC6KrtFvGO9g76nv87A+Mllw1XEh8W8xvTILslqyqfL5c0kzmPPotDh0iDTX
9Sf1d/XH9hf2aDa4twk3Wfeq9/v4TXifOPE5Q3mWOek6PLqQ+uV7OruQe+c8PjyV/O29Rb2dvfV+
TH6ivvi/Tz+mv///wCAAIAA6a1/zX/Z1Fp/nX+3vwR/jn+0qa5/f3+1IFl/bn+6fwN/aX/Paa1/cn/yVFd/i4A
WPwF/0YA1KayANYBr/LI/DYnH54R+44i50IF+uoevvSR+rYbWp/9+p4YEkuF+qYU+fcN+soSNaKd
+x4PvU45+6oNNPnd/OIKiKXN/mYHo+rJ+FpOQ5Zd99JGZ0IB91I+mu3h91Y3ypn193oxMkZB98Y
q3fKZ+CYk9Z79+K4fZUtx+WYZsPf1+r4T3KUB/DoNM+OI9P51e4+F9Hpp9ztx8/pegufN9ApUVpR
x9EpKdkF19MZA8e6N9XI3wZu99IYu0UkB914luPZJ+NYcuKRR+k4SW9zp8f6cu4kh8ZKNtzVt8R5
+vul18Tpxlo9h8Y5j6j0J8jJXGerR8xZKiZi59EI+LUbF9YlXqPTV9w4lcKPP+HoXW9cZ76LEE4NZ71
qxiiy+97wafCtzx7yaODOqZ74Z9cjjV8D5tPec98TpdTZXZ8n5NkUSt8849qPOp9V4uJKP59qYcZ9F
p7gLrd34B7Z7Vqyq57Ta/zthJ7VqrUoZd7cKXJjUd7oqDZeQV75pv6ZNR8PJctULV8kpJTPKl8+l2a
KQV9QYhC8vV7PMS43ix7F75wyWx68bggtOI6+LIholp7Eqw0jFt7RKZeeDx7iqCcZDN745rqUEB8
O5UrPGI8o4+RKQt86YIN8aZ7CM6U3Nh63sdhyBd6ssAns616tLIUn256ybKQi2J696vdd2p7OqU6
Y4p7IJ6iT8Z77Zf3PCV8WJFtKQ98nYo38EV64thw24h6s9Bvxtp6gshksol6fsC0nmZ6jbkNint6tLFxd
qd69aneYu57TqJRT1V7p5qvO+d8F5MqKRN8XYsF7tp6xuJM2k96kdmxxc56WND5sZR6T8hUnYI
6Wb+vibp6fLcNdgN6uq5zYmt7EqXgTvZ7bJ07O7J74JS8KRd8Klu28QeHM35y3N+GYH53yL2Fj3
6DtJ2E336soH2EO37TjF2DpX74eDaDF38xZAmCI398T+KCKH/HO7aB7YADJ8qBx4Be7yKGMlfi
2xiFaYcLxxGEOYY2swiD+IWOonwqDXoTqixqC2YRKdyKCW4PDYyWB6YNTTzGBhYLdOzuBVIJ
WJ5SBK4HL7VeFRJFC2WeEhl+JxXyDw43XsZSDI4xinbyCIYrzifeCHomMdiuBrohDYlqBSocTTp
SA84XZOs+AyoSPJ2WAoIMf67SEc5qZ19SDtJgCw/uC9ZVxsDOCWpMynH+B05D9iOKBZo7TdU
KBBozBYaCAT4rBTgqAcYi4OnCAT4asJzuAJYRZ6hmDxKpV1IKDB6CDwpOCSJ0drumBsZoQm1
WBL5cPh96Ax5QcdGeAc5E8YPKAM45rTYp/+ouPOh5/3li+JyN/roWK6KuDPq1A1OmChakFwTK
ByaTOraWBM6D0mjGAsp0lht6ATpljc49//pW0YEd/wplWTRB/io5rOdt/blrQJyt/Moa+50mC2bal05u
CHLGgv/mBXKybrlaAxafqmS+ARaNDhfx/5J6nctF/lpohX7B/XpWuTKV/KJEvOaJ/CIzHJzR+xYfZ5
eqCkMAU0k6By7o9vr+BA7Rjq2aAaq7cmCx/6aldhRp/hqPnchJ/Op6IXxl/Bjk8TDI+z5PiOWH+sl6jJ
zx+Z4jW5JaCXMI70PqBksLDvXCAXLwKqjGAJLW4lxd/nq9rhCl/NqkjcUh+6KLtXnl+spzES8Z+fpaK
OSI+YpBIJ0J+F4m04zuCMtLhz7CBZMtlvDiAkMPkqRN/67y+lhZ/XrWXgOl+8a5wclx+oKdWXeR+a
qBDS1x+NpkdOPB+HplJJ0h904p34eWCDdxGzoOBPNQ0uzCAZcwLqCJ/usP4IT1/J7vfgo1+tbPA
b+5+YqusXWZ+LaOfSwN9+puEOL995ZOEJ0x9m4sf41OOHx0b0FWNEH1KvV+Lnn18qmWKXH
26l3KJJ336hlelAn4+cY6G436ZXoqF0H8HS5GEz391OlyECX/QJgSDS4BR4beNclY3ztWMCiWN
u/iKooTmqRKJblRbljyIRIPXg3mHLoNdcKaGIIl/XceFHYK3SveEKoJrOCGDboIKJdGCr4Gu4DCM
hl80zWSLI421uqCJxlw5p9illorolSOHelmigoSGb4hpb9WFb4dPXRqEfIzPSm+DloVGN8KC4oQpJ
aWCJILy3rqLxZgezACKY5XMuU+JA5N9ppqGH2ZFrlAmGwl9mgY2Fvl1zbwaEyYuYXHeD6lnSSfi
DEogCN2+CZ4YrJX6BqIQd3UyLJKEJyqiJwp3puAyIYZrNpXeHOZf2kvyGlpUvgKKFI5J8bkGEOY/
dW92DZY1OSYiCmlq2NyeB8YgiJWmBL4VA2/yKqqnoyViJSqYEtsGH6alhpEaGwJ6GkemFqpr6f
7KErJeCbXiDxJQeWz+C8pDMSRuCJo1vNu2BfIoaJXOArYZm2q6KRrLZyB2I5K40tZqHgamlLoza
GWKUjkPSFQqDKftyERJyCbMaDXZhSWrScjZQ2SLqBwJAONruBFIv3JX6AOodz2V2J/LvSxt6lIL
ZjtG2HK7DvoiKF/6u7j/yE56aUfgWD56F+bBKDAJx/WieCMZeVSFeBZZKeNoiAul26JYh/14hk2Ba
JyMTDxZSIWr5+syWG6rg1oPWFuLJAju+EmqxVfRyDlaZ2a1CCrKCsWY+B3ZrvR+yBEZUiNk+A

Z49kJZB/g4k21sSJmc2yxFGIKMaxsfOGtL+pn92FfbjqjfSEWLIwfEODTat+apuCYaTaWQGBk54/R
4iAyJeSNhqAIJDwJZd/O4nv1XGJaNaewyiH+s8LsPCGhsdgnvCFSr/AjSGEILgge4yDELCDagOCI
ajyWlqBVKFqRzSAiZnVNe1/5JJUJZx/AlqO1d2V/XwyxAaT3HylsjmRwHzeoF6P530tjpCOIX1+fM+
McX3TawCKw342WSSJHX6oR1mHh38gNX2GKX+cJFWEw4BF1liU7YThwsmS2oRosRGQzIPu
n0SPAIN/jYyNRoMVe+yLoYK1aj2KAIJpWIGlalluRtiG3YH2NSGFi4G9JCaEJ4GS0zyUC41uwZC
R/owor+2P9YrjnnyOMYm5jKCMgliXex6K5oeAaYyJUlaDV++HxIWbRmaGRYSwNNCE/YPCI/yDm
4LH0e6TUZXuwFiRRZPersqPPJHQnS+Ne4/ti62LzY4WekqKNoxOaNulq4qYV2WHL4jzRgGFvod
KNImEf4Wql9eDIIPk0LCSuZ5ovyuQq5uVra+OoJjEnCqM4ZYmisGLNJOWeXyJoJEYaDCIG46nV
uCGqoxCRaKFQonbNEuEBoell8WCplT5z4KSPabSvfQM6NHrIOOKp+7mxSma5xiiceKvpkXeK
OJKZXdZ3uHppKxVIOGNY+TRUGEy4xyNBqDjllnI9GCHIYRzIOR2K9NvN6PzqsKq3aNxKbBmh6
MBKKmiOmKV56Xd+KIwpqZZtqHPpasVdaFzZLPROuEYo7vM/CDH4sql9+Bo4cRzRyRjrfPu7WPf
7LLql2Nca2/mRyLr6jiiAKJ/6QRdxmIZp9OZjSG4ZqeVWVFcJX/RJGEA5FbM8OCv4zUI+uBO4f2y+
WRW8BlunmPSLp5qSGNM7Sjl/qLa68Ohv+JtamCdjulFaQBZXuGjJ6QVMWFGpkrRC6DrJO+M4
+Cao5nl/SA44i9yp6RMcjBuUCPGsl+p/eNAbuxlumLMrVbhgyJda8KdWmHz6jBZM6GQaKCVD6E
z5xLQ9GDYJYLM1+CH4/dl/2AI4lsyVaRCtE2uB+O8soipvuM18L2lgOLArvKhT6JP7SedLmHkq14Z
DyGAqZaU82Ejp9FQ4KDH5gtMzeB4JEsJASAWYoCyL+dmHuXt/ma4Xvxp0OYLHxNloaVsXyrhd
uTTn0NdUSRBn10ZJyOvn3fU+eMe35SQ0eKRn7UMo+ISH9zlr6GM4BDx7Sck4PTtwGZ54Nqplu
XPoMDIaaUz4KvhQqSd4JgdI2QOllaY/6N+oHeU2OLwYGxQt+JIYGMMkOHplF8lpOFIIGCxpmbtl
vutfyZD4rDpWyWa4mclNCUBYiYhEyRtYeac+WPfoamY2+NSIXEUu2LGYTzQoOI+IQnMgCHEo
Nplm6FB4KqxX+a/pQCtPaYWZiYpHqVtZayk++TUo5/g4CRBlzWczSOzos5Yt2MoYmnUn2KgYgh
QjKlIblagMcaGkYU4Ik2Ei4O5xG2aZ5wHs/SXwJloo4eVGpbLkw+SupRngraQbJIPcoSON4/EYkuM
D42AUg6J+ItEQeSH64kPMZSGFYb8lj6EDYTAw2eZ6qP+suiXSKCyonuUpZ1mkhiSRJpQgdiP9p
dGccSNwJRJYayLI5FVUZKJf45qQZCHb4uEMWyFIYjClk+DfoXKwliZhawFsemW46gJoYuUPqQK
kTuR3aA7gRKPj5x3cRmNV5jBYR2LLZUWUSSJFJF4QUWHAo3fMUqFllopsll+C/4a9wTmZN7QS
sNeWk69doleT7aqjkE2RiaYcgD2PNqGgcGGM+50wYlaKz5jOUK+ItJR5QPWGN5ApMSSEvYv+I
m6CkYeVwBSY+7wPr6qWWbacn1mTsrEljziRSKvvf0aO7qbCb42Mq6GfX9eKeZyGUCilXJd3QJm
GRZJqMPaEZI15lnqCNYhRvt+YzsQMrn6WLL3wnjiTg7fNji+RErHffluOsav3bsSMZaYXXzKKLqA8
T6mIDppoQEOF9ZSVMMuEFo7YIoSB5oj1vamYrMwHrWqWB8VfnUSTW76hjVGQ5LfofZSOfbEw
bhuMKqqAXqeJ7qPVTz+HzZ0xP/uFspaUMKeD1JARloyBplmCvCKIYHtbrHSh4XvSnNaea3xGjS
SbVHypfYeYV30RbgSVeH2AXm+Snn30Ts2Pz35xP0WNDn8CL6OKhn+5IRGH5oCqu3SkVIM4q8
Kg6IL0nCGdgoKujHOadoJofOOXg4lobXSUq4HxXfSR2IHGTmiPEIGpPviMVoGbL3CJ2IGqIPyHO
oHcupyjdIrl1qvWgEYn3m2Cctlj5i8iZslgJfEuWw4cibPCT8YZGXYSRJoV8Tg6OZYTDPrSLsoQVL0
OJPYN/IOqGoYL4ua6iu5KUqhmfW5DmmpWcAl83iwuY/o2je6GWE4wZbF2TQYqcXQ2QfYkqTb
WNyYfGPniLIYZsLxultYU2INqGG4P7uMqiJJoIqUCew5fNmciBzPv0ik2YZpM4evaVe5EJa8mSqY
7nXJWP6IzMTVyNPIq7Pj2Km4i3LvqIMYbelNqFk4Tt/GhqaGmqF2eS56tmN+a8JuwiXeX75jLeja
VApXzaySSLpMpXA2PbZBmTPWMv42vPfqKGosDLuOHqYiBiPiE+YXgtwihQak6p4Cd5KWYmA+
aiaHviLiXh55aeYuUmZrSapGRw5dYW5OPAJPpTJiMUZCJPb+Jp40zLs6HL4oIRSEcla5tgag67D
Tpoqdj6x9lyiaNKgfh+SXL6PdeM6UPp+lae6RZJt8Ww6OnpdgTDOL7ZNTPX2JP49QLrWGw4t2IS
2D+od4tPigprhXpXadTbNMlhGZ8q46huaW6KIRd+uT8KRxaS2RD5+fWnCORJrWS7uLkJYZPS6I
4JFjLpKGY4zOIUGDI4gds9mgbb/YpF+dF7oqlQaZvLRxhfKWq67adxOTrKIKaHSQw6PEWdmN85
5FS0iLPZjOPOKli5NeLnGGD44KIVKDQoitsrugQMdWo2Gc6sEZICWZjrrHhSSWeLRwdlyTcq4dZ
9iQhKfUWVqNr6GSSuiK9ptaPKKIQ5UuLIWFyI8iIWGC+4knr+WtQ3tWoTepIXvIkp6IBHw7g/ahQ3

yidWidnX0RZvqaFX2KWHeWmn4ISeeTMH6QO3WP1X8xLOOMtn//H4WJfoEOr3asOILTOMCoKI
KUkh+kG4JXg3igY4ledPKcxYHuZpSZRIHIWCGVzoGuSaKSaYGjO0OPE4GrLMiL/YHaH4alxll1rt
erWYotoCenUYIBkY+jS4hVgvmfm4d6dlScA4aqZjSYh4XoV9KVF4U1SWORt4SVOxeOZoQFLLC
LWIOYH4alH4NErhuqoJFen3immY/NkO2ill49gmOe5ozHc/2bToteZcGX0ooDV3aUaYixSSORFYd
uOvCNzoY8LJuKx4U2H4aHj4Q7rWOqBZh+nsml/ZZOkEah9pQfgcmeSpIMc3KaspAGZUqXNY4O
VxiT0lwdSOCQg4o3OsiNQYhjLlyKO4bDH5SG/YUhrLSphZ+Mng2lgJzHj4Ohe5n9gRWdzZdKctK
aNJSkZMCWtZINVqiTTto99SI6QAlz5OpeMuYqHLISJqYhFH7+GV4X3q/CpGaasnVOIFaNFjtShD5
/YgHWdYZx+ckKZx5kyZEWWRpX1VksS3JLDSESPjl+gOmyMQlyNLH6JJ4mpH+aFxYa2qw6ova
3PnH2kuqm7jgmgtKWhf7ydA6GjcZ2ZZZ2xY7iV4ZnPvdGSdZX7R++Plpl2OjiL0o5/LHCIs4r1IAiFR
4deqhqcobTYm4KkcLAZjQygaatVftScs6a5cNCZD6lpYwqVhJ2nVUOSFJkwR4WOvpTEOfSLbJB
mLFmiTlwrICSE3YfuqRKoMbvamoKkMbaBjBegKrEfffWcbqvecAqYxKanYmGVM6F9VLuRvpxaR
x6OZZdAOboLEZI0LECH8o1HIDyEgohqqAin+8LVmZej/Lzwi0if9bb5fTicM7EDb2OYhKsWYdOU7
aU2VEeRdZ9eRsiOGpmROXyKxJPYLCyHpo5BIFCEN4jSpAa1QHUQlkSwkHvhiJ6r3nw5ev2naXy
jbXyjDn0ZYCCe0n2cUq6aqn4kRS2WmX63N86SmH9mKkuO2IBJHhqK+4Fwo7u0P4KnlfyvmYJW
iFeq74lMerOmgoHfbTailYG9X+Wd+IGpUn6Z1IGfRQmVx4GIN7iRyoHDKkiOEIIOH6KM4KMo1C
zZlmclZeuwlrh/mqGYfAemOls4b4bPKhZYY9X6ydNIWUUI CZFYT5ROiVDIRxN6WRE4P+KkaNYI
O1HkCJgoOPosqyqpBklRiuA47Zh4KpWY1TefWk9Yv6bl+gqlqvX1icell0UhGYX4hDRMGUYYchN
5SQcoYVKkOMxYU9Hk+I54R6oj2yB5cXIjGtX5T3hwKotJLceX+kUpDvbCagBY8RXv+b041DUc2
Xvot6RJSTx4m+N4CP3lgaKkWMmlavHmulS4VPobOxf520k/ys2psJhmaoL5heeO+jzZXaa6aff5N
kXpKbS5D/UXaXNY6hRFeTPlxRN2GPTloWKkyLk4gPHqKHmoYNoRSxDKRek2asZ6EbhdmnvJ
3WeG6jWZq0azOfCpeiXjCa1ZShUSeWvZGrRCCSwo7FN0WOzlvzKISLCYISHtOG/oa0oFSwrKsl
kq+sBqchhS2nWKM5d9Gi8598aqeeoZvPXbmaaZgzUMiWTPSkQ9uSUZEnNx6OV428KIKKjYp9H
v6GeldGn36wWbGWkdCrs60PhEqnA6ildwKimaQ3afCeQZ/zXR6aBJvBUEqV5peZQ3+R5pOANu
eN6Y93KkWKH4uTHyKGB4fBno6wE7gYkOirbLMAg2umuq3ldjeiSqj4aT2d7aQYXlaZq59HT9CVi
ZqAQyWRh5XDNrCNiJEZKjWJvoyQH0CFqlgqnZav176KkA+rMLjsgq+mfLNEdY2iB62vaKadpqgl
XAaZX6KsT2mVOp09QtmRNpfbNoGNNpKSKiiJbY1tH1mFWliCmFy9S3wji4m313xcftOyaHyeciKt
WHz+ZZioX31tWTujgn3uTNGevH55QGKaDn8TNByVan/IJ7uQ9ICtHMiMZIHRmD28TYLLi3K25IJ
2fsKxfYlocguscoH3ZYGnfYHWWSuipYHITMmd4YHJQGGZMYHdNCWUjYIIJ9KQHoJdHO+LjYL
hmAm7cYIWioC2CohvfpSwpYeNceeroYbLZWams4YaWReh34V+TL2dHIT1QF+YbYSENC2TyO
QmJ+aPYYPtHRGKz4PZI7q6sY+4iu61Ro5CfkCv3ozRcZuq3YuGZSSI8lpMWOOhHlknTJucX4gQ
QFKXuocNNDSTH4YfJ/eOvIVdHS+KKYS4I1i6C5YBio60npQBfeOvMplFcUeqMpAvZNqlRY5qWK
agcYy5THCbt4sTQD6XGII/NDSSgYgBKAmOHYa1HViJg4V9lvG5eZwvih20EZmtfWyuqZcscNmp
qJTGZHMkupJxWFWf45AyTDKbKI3/QBaWhovfNCiR6lnSKCGNdofyHZqlYlYjlnO4/qJiaizmZ9TfQ
CuMpxAcHepMJIGZCKkQJZeWAYfaJOMS/qaqpDLP/KWBY4eNByRYluEKDaM4okSHdalloyldS
4lqiQiRGzMqTofHOTyqE9b/aoxJ2wY7Cj0Zo2V6ye9pbTS66aNpODP76VjpBHNASQ5I0fKECMX4o
aHgmHllctRy4QK6jiFCy2apke62tbaYjb0KoYqlKYxGjap4CVyWei5oRS0KZyJYwP3CVHZJjM9mQ
cl6oKD2L6YsNHjOHHYeVIEe387Sph4OyjK/eeumtHKsPbpGoDKZiYnWjDqHGVqCeK51BSteZZJj
KPyOUt5RmM6yQCJAWKDeLgovnHliGulftk2i3sLqbhsGySLVKekOs2K/wbfunw6qkYfGiwaVrVjCd
2qBKS6ZEJs5PuKUYZY9M4aPsZFeKDGLLIylHnaGZlg2jQjFZ3zsgRi/PH0HdUi5GH0saX6zbX1
7XeKt033eUn+oT35aRxqi4n7IO7ydhH+EMI+YOIA7JVGS/YEaG5SNsYlvjSXEa4MrgT2+SoLJdXC
4L4JuaZyyhll0Xf2s7IIOUpynbllCRzuh/olLO+CcoolrMLmXSoJgJYGSF4K0G86MyoMzjS3DiIlNgU
C9aohndXK3UYeGaaWxr4bDXgysHYVUq+molWCR1OhMoUKO/6b0oStMN2WeYRgJauRTYQ

uHACMAIQfjRXCvY9LgRu8m43idUK2fox8aXyw34s4XeurT4oKUpql04j1R06gaYf3PA6bE4cQMPy
VwoY7JdCQnYWIHCuLUITzjN/CDpUrgOG76ZNDdQi1ypFeaUmwK4+XXcGqm43kUnulH4xMRz6
fulrHPBGaZ4IXMRKVGoF7JfOP84bFHGCKoIWpjJ/BdZrugJe7VpiMdLa1OZYoaP+vmJPZXYCqB5
GgUkakio+BRxqfIY13PACZzluEMRiUdomiJhuPQYffHK2J2oY2jEPA8aCsgEW61p3FdGy0vJrbaL
uvGZgAXUWphpU8UhekB5KURveenJAFO+6ZQ42OMR2T5osoJj2OpljdHPKJK4aui8PAgaZgf826
Z6Lqc/60TJ9xaFiupZwMXO6pD5i/Uc6jj5WPRsCeIZJ6O8uYxY9+MRKTYoyTJIOOGInDHSyllocUi
yjAlqv4fyq6Bqf0c1iz5qPtZ8KuOqADXGqon5wwUV6jG5h7Rmadq5TfO42YTJFbMPOS543pJlqNm
4qSHV6lGldrinC/zbGBfnm5sKz2cq+zjKhIZyqt2qPwW+WoOp+SUOyitJtTRgydQQZcrO02X4JMcMN
GSel8lJl2NLYtMHYeHroe0iaq/gLb2fdC5ZLHiciKzQKzIZqutiae+W3Wn5aLPUi2iXJ4ARcCc55lLOx
eXhJSyMLSSG5A7Jl6M0ovrHaqHVofwgZDNd34NdmLHGH3la17Aqn3OYli6SH4JVe2z7X5fS5Wt
n37VQUenan9gNwyhSYACLQebJYC5lvaVDIGOGnCO8YKNgh3MioQEEdt7GJoNea8m/s4LIYOW5
T4KAVkCy9YJTS+asrYJIQZamdIjYN1mgSoKFLVOaloLBI0CUFIMRGruN+4OGgnXLnom/dyXFO
lifbAO+w4eNYSO4aYbGVoCyF4YbTCWr04WQQdalmIUoN5ufaYTgLaZPoSjl4GTOoR0Gv2NJI
Rmgo/KxY9WdyzEWY2+a/u93YwzYSG3horzVoWxN4nPTDOq9ljLQfOkv4flN8iemYcbLc+YeIZfI7
mSflW3GzWMaYUsgoHKCJTDdxvDI5K7a+m9GJC+YRS2wo8DVn6wc41jTDSqMovjQgGkAlp/N+
ad34k1LfyXwof7l++Rx4bYG3WLR4XPgnvJdJohdwPC/pega8O8eJUmyPG2H5LsVmKv0JDNTCC
pj47QQfijXlzwN+ydOlSrLhWXEol0JCaRCIfPG82K3oZDgknl759tdtjCd5xra5+77ZlvYNG1k5ayVkev
QpQSTA2pAZGXQe6izY88N/CcpYz+LiuWelrNJFaQYYiqHBuKJ4amgeXld6SjdoLB/aEZa1S7cJ2
XYI+1EZpaVg2uvpc7S96ofZRCQcyiR5FrN9+cHY6zLi+V64wJJHiPzltHF2Jiob6gWDIDam2dfvBjq
WoatC6+6GjYBm0lp3oVaauQJpNS4Wn/pbXQYehx5OEN7CbmpBPLhyVZo0rJImPR4oaHJSJBod
AgL7Hra64dWXBK6ooake6k6WhX560KKFkVTutzp1JSyqni5lUQT6hU5WBN36bJZHNLgWU8l4zJ
JSO0oqzHMOIl4d8gBLHWLOgdNfA1a6KadO6O6l+Xzizy6S5VOGtbqAZSt2nKpukQQCg8ZdRN1
SawpMjLfGUjY8aJJ2OcYs1HOqlOoeu9ER80HqD39Z84Xrdy2t89ns9tv19KHu0opJ9WXwojip9iXy
Zebx9xn0TZUh+FH2WUNx+bn4ZPHd+5X6UKEl/en8n8ih7alRO3dh7j4PayYp7u4M3tTt8B4LSol
8UYJyjMV8mYIYelx87IHNZFB9TIGPUB59t4FNO/J+PIEEKA1+1YCx8DJ6No4C2/56a4yNx856pos
cs6N7A4nin4p7Yliwi4Z7vYeJd358JIZ2Y3N8l4VzT3V9EoRpO319pINbJ9h+QYli7ml5KZeq2k15aJ
VOxjZ5rJL2sip6FJDonjZ6fY7oimB66lZ5doZ7Y4sZYq177olETuB8fodoOxZ9HIWUJ6p9voN47Lx4S
6FL2LI4j54UxLp41prhsM15Rpf8nP5uZUpiU96MJJsdaF6u4+5YfZ7WY0PTII7+lpeOr58n4fDJ499
QoTF6zZ3nqrf1zx356bTw0t4NKLJr354qJ8Qm9B5lZtriEh5n5fadMN6MpRUyUV62ZDZTdI7f41WO
nd8KonxJ5V8yIYU6b93FbSJ1d13XK+swgV3pqrMrlV4HqY3msZ4m6G0h2B5lJ1FdAB5upjkYKl6Z
5SQTWI7E5A2Ojp7wowCJ518Xldl6E12rr471lJ27riGwMJ3MrLKrS93q61amb94Kqf5hnt4sqKscz9
5UJ1tYA56Apg8TPI6spMEOf57Zo35J6N7/4hd5ux2Ycfg0yZ2ncFEv2523Lqkq/p3U7RkmKx30K4x
hYx4VqgMcnV49KHx215qZvhTIV6W5XFOb57Fo/TJ6l7sYlQ5YV2JtGB0dV2XcobvjR2l8Ktqt13C
7uVI613hbSFhK94Bq18cbp4o6Z6Xth5Wp97TBI6D5hxOYN6z5GPJ617boon5CZ19tsd0KJ2KNMc
vSt2XMsCqex2zcl6ltd3RLryg/R3wrLucR54XqrtXlp5FqLyS795zprwOVJ6lJMgJ7F7Norh5vSDrnlR
06KDK3nKwFaCq3pJrQaCTHrdmb6B9Htthn+Bonv4czKBXnyRX9yBKX03TJSBAX3dOU6A/n5zJn
qBEH8r5TCCUYKnofyB4oJEvsuBdoHmq4+BLoGrmGaA64FyhVSAslE7cjKAgYEYXweAYIEFS+y
AS4DtONmAVYDHJkKAbICh43yBIYvf0GCAwlqgvUqAY4lnqjKALlhlly9//ldmhEV//lZrcUx/uYWJXk
h/q4S7S1h/poPkOHF/vYMAJhF/2YH/4c2AK5URzsl/05L3u85/fZDkqNI/Tl8Zlft/Jl1Vgzl/CluccGx+/o
n0XZh/BohdStV/Fla9OBZ/N4UcJeZ/VoNC4DJ/Y55CzUI/DJtWuml+tphwp5V+i5XYINp+apNMgj5+
VpDNb5p+WY5dXPJ+cov5Sl1+jomNN8h+uYcuJc5+2YR+3rx+wKdpy95+baOzuQx+G6AApIV99J
yek7t92JIHgUR9ypX+bsl905LEXE1984+XSel+E4xiN4p+QlIBJdd+WYW93Vp+OrCmYPJ95awnt9d

9kqenpTp9cKNykr19WJ9HgGd9T5spbhB9XZceW7x9gpMiSYN9po8eN1R91Is4JeF954bi2/t907nn
yUV9erSYtp59JK9GpB19Aqo/kb5866VBf4t846BPbVd89JtwWyt9HZagSRx9RJHJNx59dY0UJep9
hYfp2qN9iMMXx+59LbzstUx80rbAout8rbDzkK98kqstfqB8h6VubJR8mJ+9WpB8wpoYSK9865Ro
NuR9lo7WJfJ9MojQ2T99UMxBxp588MVVtBF8kb5loc58ZrfKj7J8R7ExfcZ8N6qaa958RqQMWgF8
cp2ESEp8nZbwNq582ZB6Jfh87lmb1999ItVlxWp8vs3jswh8WMZLoN58Kr7Mjt58BrdlfQ9786/Ba0V
8AahCWYl8LqDKR/Z8W5IMNoF8m5H0Jf18sYpK2eOKs3hCx6iJl3jYtXalgXlxozmHnXoSkQ2GvXq
1fvaF4ntbbMyFEnwSWpOEUHzXSG6DnX2dNkeDFH5RJMaCl38u2GSJV4EuxkSIUIDytCyHToC4
ogWGfoCUj/aFsoB3fgOE64Bja/qEMYBIWeCDhoB6R96C5YCLNeCCa4CJJKB84CR1vmIL4n3x
O+HNojrsu2GQofhoOWFf4b+jveEw4YjfSWEDoVUazuDZ4ShWUCCz4QDR16CQYNeNYaB04Km
JGSBYIHd1XaHRZK1w4OGU5DasZqFZY8En7KEql1hjeWD84vLfDaDR4pGanWCr4jWWKaCK4d
4Ru6BroYRNTeBTISIJdyA34MP0/6GiZt4wiSF15jUsFaEqJYynouD8JPNjN2DQJF4e1KCm484abe
CDY0IWBObmIrmRoaBJ4i8NPOAzYaaJCeAYIQ50qWF9KQ5wNKFA6DRrw6EFZ1qnV2DYZpDi8
6CtJctemeCEpQsaPSBiZE8V3uBGi5aRh2AqYtwNL2AT4iUJDn/2YVq0VeFda0lv5mEg6jgreqDla
S1nFGC5KDFit2COpzleZaBmpkZaEeBFZVgVvWAp5G4RcGAOo4KNI9/4IpxJEB/YoaB0AeFD7X
YvlqEHLdprL6DLKv1mz6Ce6c9ieeB0aKUeMKBMZ3+Z5eArZl8Vm2AQZULRWR/1pCTNGB/fYw1
JEt++4d6zrqExr6ZvQyD0bjXq3KC3rMTmhGCKK2eiNyBeqg1d9yA1qLcZtqAUJ2TVdp/5ZhXRP5/e
5MQNCt/J43gJFV+plhVzV+EjMdUu8GDlcDaquCnrmPeB47Qdh+KBL63rdwaAhqfEZil//qGpVV
J/IJuWRKB/K5V3M/p+249sJF1+WYkVzAWEXdAGupKDY8j6qTSCaMHXmAmBqLq/hxCA8LotdIK
AQqyjZZV/uKWIVN9/T56tRFF+55exM9F+m5DQJGN+G4m6zPaR8Hd6u9GQMngtqreOfHjgmYuN
CXmQiHeLnnpCd4CKPHr4Znel4Hu6VWGHjnyGRGKGSH1ZM1mFLX4ulymEEH8xy7WQnoABuq
eO93/rqaONVn/WmluL9H/Hh4+Km3/AdraJSn/FZcilAX/bVMYgwX//Q+qFjYAlMwKEgoBKlvmDbY
CCynqPgYhYuYKN5od/qJWMUYall5qK/oXghrqJs4Umdf2lcYR4ZSmHOIPhVEeGCoNcQ4CE5oL
VMrWD6lJLIs6C24G8ySuOp5CguEqND48Gp3SLfY1slpWKL4vzhdKI6oqHdTSHroksZIOGgofhU8
aFZYajQyOEUYVkmnKDYIqulqmCWolcx+WN75j0txmMWZaYplmKyJQ7lZSJfplKhO2IPY/qdG2
HBo3dY9+F4ovbU0iE0onjQsqDylfrMjmC34YllpeB2YP3xqyNXaFEteOLyp4vpSuKO5sXIh6l9Jgvg/
WHtZVWc5eGgJKSYy6FX4/ZUsCEUY0rQm6DR4p+MgyCXlflqaBTIUZxXaM4amitMKLT6XUpB
2JwKlck4ale55ZgxeHPprBcteGC5c8YpCE65PGUkeD35BeQhyC1Yz3MeWB6lmlraAz4YixDyMg
LIBs5aK7K10owOJXKjgkoSIF6R5gjCG2KAichGFpJvdYe2EhJepUcqDeJODQciCbo9eMbyBgotalS
OAY4cQwwWMObpRslmKpLT7ocWJEa+hkWKHx6qFgS6GhKV3cTSFSqB5YTiEJ5uJUT+DGpak
QWqCEZG8MY2BKlZuls6AClffwbyMAcKasR2Ka7yUoJil1baGkFCHhbCsgDyGO6rdcGWE+6UdY
l+D1Z9mUL6CyJm2QRKBvpQCMWGA2Y5llth/uoiUwHGL1MrZr/qKOsREn5qlob2Zj2qHTLbtf2+F/
rBJb7eEuKmyZACDj6MIUFGCgpygQMmBeJYcMTyAlo+zluB/eYkwwFqZVHbqsDaXB3eeoCOUv
3hVKAiSsnkUgAyQsHnWcDWOU3qeYEyMx3tqUFaK2nw9QHul+H0eMlyHQ34UIaSFgh8/v02YF3
8Or0KV238En0WTO378jzuRpX8Df1OPsn8Ub5ONy38yX8CL539aT9+KCn+NQBylOH/KMEWGIIA
WIXmE24B/vj2XAob6rkuU0IY2nmmSo4V0jnmQtITPfqioZ4Q1bv6M9IOpX0CLHoMsT3WJUILAP8
eHjIJYMAaF94H9IVOESYgnvSaWLI7RrUqT/Y1YnXyR04vhjaSP6lqSfeyOB4IRbluMMoggXruKZ4
b4Tw+lqIXbP32G9ITDL9CFbYPEITKDx4K4vBGVe5ayrEaTTJSBnlyRlpJUjMqPOpBXfSuNXY5rb
bSLioySXjKJxqo8TqeIE4juPzWGalcoL6KE6IWEISODRIPDuwCU556PqzOSvpuxm3qQmJjVi8+Ot
JYqfEuM15OPbPSLBpEIXZSJQo6GTi6Hj4wMPuWF44maL36EYldMITaCsITWueeUbKZ5qiuSRq
LtmoSQLZ9fiu6OPpv+e4GMY5itbEeKkpVwXQalzpl8TcOHG48TPp6FbYvyL1+D6lj5lUmCLYXSuM
SUDK5mqRWR5aojmX2Pv6Xdif2N3KHHequL/53Ba4+KK5nOXHCIZ5XnTVGGspIMPIOFAo45Lz
2DfYqNIVmBvlayt6GTw7ZBp+mRnrFCmE6PeKxBiOiNj6d+ebSLrKLKar2J0p4pW8WICZmQTNC

GU5UBPf6EopB3LxSDHowKIWeBXYd1tm2Ti74Wpr6RZrhxly2PPbLFh+CNTq1NeMqLZafiafaJhK
KIWyOHtp00TFaF/pfnPa2ETZKdLu2Czl1rIXOBC4ggTmTX8XjpaqROL+xljqPDLIshwKNGLMteAS
LKKz4aU6JQabUWpuHcKC3S++FtpqjPWmEBJSXLsyCh46lIX2Ax4iytDmg0Xa4pRidyXeHlqqazX
hWhvaYK3kXeAOVInnfaTeTDnqwWliQjnuFS22OGnxjPKKLSn1TLcGJd35jIAuHL3+us2ufon58pFic
qn6VIVmZvH6thlGXJ37Cd26Un37iaLeSJH8QWeuPsH9ISxKNSH+NPFuK7H/hLZGLv4BOH/WGfo
DhsouemoYlo46brlVwlKSYylITYhbOWP4RJduWTwoPGaECRUYNswYeO6YLtSsCMjIKZPByKPI
JPLWalHllcH+GF4YH9saqdx416orqa3ow3k9+X/Yr0hQKVd4nEdkms/YiiZ7qQkleRWRqOMYaJS
nCL44WLO+WJoISaLUGHi4PMH8+FV4MAsMmdFJTqoeSaL5L9kxWXUJEPHeuUzI87daaSVI14
Zy6P6lvHWKmNkloZShyLS4hzO6+JEIbbLSKHAYVwH8+Ey4P6r+acgZxSoPqZopnFkiiWx5c0g3K
URZS6dOWRzZJRZomPYo/6WCONCY2pSbmKxYtgO3GIhokILQ+GcYcWH++ELTYtrvWcBKPJo
BeZKKCVkVSWUJ1bgrCT0Jo0dDmRWJceZfaO7JQcV6uMk5EiSWCKTo4zOzqIDltTLP6F8oiglA2
DoYXTTrfKbnatCnx+YxKdekGqV7aNygdUTa5+gc3uQ8pveZVSOG5gxVyiMKJSOSP6J4ZD3OvyHn
o1uLOiFgYoQICeDKYabrOibTLKmnng2Ydq4Qj1WVoKlygN2TGaT8cpqQmqCVZJWOJpxCVo6Lx5
f3SluJfZO2OrOH0I9/LMqFHYtrID2Cw4dHq86bDboEnPqYOLTMjkyVYa+Lf+qS1KppccKQTqVvY
9+N06BTVfuLcJtXSB6JJZiOm2G3pF4LKyExYyqIFCCbYfdqrOa3MFYm/6YBbuTjWyVK7W5fx6S
mK/bcQ2QDaoJY0WNJaRKvX+LJp6SR8KI2ZjjOjKGkpNELJSEfI3EIF+CJYhdqFSodXa0mjCkz3e
AjCOhMnhNfhWd7XkScCyatHniYm6Xinq9VJmUb3udRreRZXyHOPiOaH2IKyGLnH6xHpOlw4Abp
72nUX4WmaOjuX42i6CgKH5WfZyc7X57b8CZv36sYhGWOH7sVEuTj382RniQjn+NOMqNm3/4Kw
eK24CFHpCIBYFCpxOmVIVCmQiiXlS8ixWfOoQ3fSWcCYO/b1qY5YNVYbqVzoL6VAOSxYKuRj+
PzoJzOKKM5IJHKvCKLoI9Ho2HXYJQpmSlhXrMGOH9ospinuebooDfJqbQljzbt+YHofyYVGVCoc
DU7GSCoYaRgWPIYU9OH6MQoRzKtuJl4PVHouGyYNGpbOk0JNVI7ihRpGPididwl/JfAaak44fbly
XcoyEYOKUYIr9U1mRZol3RceOhYf6OFmLrYaSKs2JBYVeHpiGNyQvpP2kO5pMlvigtprfxiROdM5
WTe1GaCJNLbbyW55ETyFqT1I7vUuyQ2ozPRXmN+Iq4OC2LHYi0KsmIbYbjHsSFjYUOpDSju6F
QljqqOp5XiGGcuZtYeq+Zj5hsbSyWbpWRX9+TWpLLUoqQYJAMRTKNfY1ZOAaKnoq4KsaH5oh
KHu2E+IXWo1OjUKhVIWOf0aSzH5acUqELefaZJp19bleWA5oAX1SS7ZaZUhqP8JM7ROGNDI/p
N9eKKoyqKr2Hb4maHxCEelaGomWi+q9CIG2ffar4hpyb/qapeRGYzKJ/a7yVpJ5nXqmSiZpjUZGPi
JZmRH2MopJzN5mJvo6QKqmHBIrUHy6EDYcdoWSitLYmk3KfN7FDhaqbtqxYeDOYf6eMaviVUa
LQXgOSMJ4oUQyPLJmERByMQ5TpN16JX5BdKpSGp4vzH0eDsYegoF6ierz+koue/bePhN6berl
Pd3mYPqyTalOVC6cmXXiR5aHNUJ2O3Zx6Q8qL85cyNyyJDpH/KoOGWlZvH1uDZYgNnK+wQ3
bkj4GsDXeTgnGn2XhJdWij5HkSalef/HnpW9ScI3rPTwiYYHu6Qi2UtnyxNXmRGX3CKKiNsH8CH
TqKO4CFnEqvJ33hjyaq+n3xgh+mz34HdR2i5H47aESfBX58W5ybNn7OTtuXe38qQguT13+UNW
SQQYAWKKSM44DBHUmJclGfm9iuMYSujr2qCYQkgcCl4oOgdM6h/4M+aASeKILrW2aaYIKqTr
GWrYJ3QeyTEIJVNVPKgoJIKKCMLYJiHVelvYKhm1mtYotZjkapOYo4gVKIEYkedGmhMlgtZ6qd
W4dMWxyZI4Z9TnmV7IW1QcqSXIT6NUOO2YRVKJ2LjYPjHWOIIIOKmtKsrZH1jcWohZBBgNikXI
6Rc/qgfY0OZ0ecqYudWsmY54o+TjqVQYjhQaGRuoennTCOPYZTKJ+K84VTHXyHgoRimkqsFZi
AjTCn8JZAqDqjyZQCC2qf6pHqZsmcF4/jWl6YVI3xTeaUr4wBQWIRJ4ocNRaNpohOKKqKUoa3Hb
SG0IUpmamrkp8RjJinbpw8f6yjSJlmcumfa5azZlabmJQTWf2X1JGITZqULo8EQTSQpYyLNP6Nllo
qKLWJw4f9HeeGMOXZmOurJKWgi+OnAKlrfwCi2Z61ckye+ptqZcybJpgyWYmXX5UQTT6TuJH2
QPSQLo7pNN2MpovyKLeJRIksHhOFq4ZzmByqyawWiwqmpagEfiKifKPycYCemKAUZRIav5xIWP
GW9JiTtMaTSpTkQJ6PvZE/NKuMNI2vKK2I04pHHjeFOob2IzSqe7J/iimV63efUmiK6k6cLmeQ6T
DZGeaZKBdWF2WIZwNTFCS55fCQEmPWZOANHqLzo9SKKKlcltIHlaE2YdmlkGqOLjWiVSmFL
OufJGh6K59cBGd+6lhY9OaGKRXV+GWRJ9jS+2Sk5p1QAOPA5WTNFCLeJDLKJilHlwoHnCEiY

fDkUW4HXdlhQ6zJ3gFePauOXisbN2prXlsYO+IKno9VTWgtHsjSWmcV3wTPZWYEX0RMfGT1X4
pJjmPvX9tG/mLnYDukRS3BX33hOayGn4leNWtNX4fbL2or35SYNSkM36XVSKfxn7wSV6bbX9YP
ZOXXKX/RMfmS8IBhJkyO44EWHBqKxYH7kN62EYRihK+xK4PleKCsSoNtbJWny4MSYLejV4LKV
Qye8IKZSVGanIJ5PZGWXIJuMgCSJ4J2Jl2OlOkGhDeKBoLwkl61QlqnhGKwWlmheFardoigbFG
m+le/YHyihYbwVN6eIYY4STSZ1YWLPYavoiTuMgeRdoRnJmyNeYQLHFCJX4PNkDG0iZDehAi
voo9Pd/+qvo3DbAKmQYxYYDahz4sBVKadbYnASQ2ZJYiGPXSU+IdYMgiQ0oZDjN6M1oVgHhAl
ulSVj8yz65cAg5evCJTsd4aqJ5LZa5SlqpDhX9ahOI79VFac1Y0xSNGYjYtsPVCUXom1Mf+QM4g
UJpiMK4aiHLmH/IVDj1KzZZ0fgyKug5p+dxipo5fbazCIJ5VSX3+gtZLeVA2cUZCDSJqYCI4yPS6T1
4vxMfePp4nGJrCLiYfHHPaHVoXdjriy86M1go6uEZ//doupMJzHarCkspmuXw6gPparU6+b2ZPCSF
KXjpDIPP6TW44YMeOPJ4thJr6LD4jTHSqGylZhjgmyk6kygdStsaVpdcuozKGfagGkSp37XnSf0ppt
Uy6baJb5R+yXGpOPPLeS5ZA0Mb2Or4ztJr6KI4nLHVWGUyBqJTuyPq8igQ2tXKrQdQ2odqZ7aV
Oj7qJGXdufcZ4nUq6bA5oiR4eWspYmPHCSe5l3MZAORl5gJruKLoqrHXmF64cuJF6x8rT/gFCTeRa
mdGyoLatGaMKjoaZ3XVqfH6G+UkOarp0iRzKWWpiPPDSSIZQNMXWN6o+pJrmJ1ottHZiFI4d8hj
LAD3gUeue6Y3ifb7q0w3kzZlivmnknWYegeHqsTr+IYHuPQ/CgX3yAOSSbc32ELo6Win6fl++RtX/f
GtaM44FShkK++H5Aevi5Vn5Lb8uzv35dZJaumn6JWZWpfn7MTtOkbX8qRAmfbX+bOUSaf4AiLr
WVloC8JBqQzIfzGwiL/YJThki+AIRJevW4ZIPSb8Oy0YNgZJattIMHWZyon4LFTt+jk4KfRByeloKR
OV6Zp4KbLtaUwlK2JECQAILnGzOLM4M7him9J4osetK3i4k6b5yx94hLZHOs2odzWYCnxYayTs2
ivlYMRBmdxoV4OW6Y4oT3LvOUBYSJJGCPTIQ8G1mKg4QMhfi8ao//ep62zo6Ob2ixOY0fZEOsH
lvGWVenClqETq+iAYldRAudDohDOXKYMlc5LwiTWIZFJIKOoIV3G4qJ0oTEhba7xZW5elG2LpP
KbxOwm5HZY/erfo/8WRamao42TnuhY4yMQ+ecclrvOWKXj4lkLxGSsIfsJKuN7laXG9iJDYVchW
G7OZtnef+1o5jxbsWwEZZ6Y7Kq9ZQUWNql4ZHHTkqg2o+XQ8Wb5Y15OVOXAottLxiSHllzJNCN
TYeaHB2IX4XehOy6wKEFeY21Kp4Fblivl5sCY0+qeZgVWISIZJVBtGsgW5KNQ5GbZY/rOTOWf4
1dLxKRIYriJOiMwliGHFiHyoZNhFq6WqaLePK0wqMBbbevK591Yr2qCZwFWAWk75ivTZmf45V4
Qz6a6pJROPuWA489LviRF4w9JPOMQYlchIqHTYaqg6e5/awDeEe0Zqf1bRauzaPjYiypPZ/rV4W
kh5wNTS+fd5hPQuqafJSfOMGVk5EBLtgQp41+JPiL04obHLOG44b6guO5qbFnd6K0FKzSbl6ufa
g3YbGpUaOvVxqkL59DTNWfHJr5QqSaH5a/OJCVNZKaLsKQSY6ZJPYld4rAHNaGioc8ew7IBHk
dcILCGnloZiG8J3nEW+a2RXpkUeKwZHsiSCCqh3wDPI+kwHz3NKufDX4BKzKZWn8elbOTsYBX
GcGOHIG2e3/G9H7ycODBCn7DZm67Fn6jXCm1OH7HUiGvXH8HSF+phn9qPqGjwH/mNPCeB4
B8K3iYU4EglfeSt4HUGgSNKIKqe8DF/YSZcRTAEYPYzpi6G4NYXFe0RYL/UiGucYLDsJKooYKq
Ptei24KvNSudlILQK7aXbIL8ljKR3IMyGj6MUoOHe93FGYogcSO/K4kGZpu5Mof4XFqzXlcpUlitiYZ
1SJ+nvYXkPvKh/YVqNVWcTIUHK+uWpISylmWRHoRwGnCLmIRLe9jEVY+IcRm+ZI38Zo64Zox6
XE6yklSyUIGsvoogSKCm9Yj6Pv+hOogCNXKbj4ceLBaV64ZLIpiQZ4WRGqyK34Txe8bDrJTOcP
m9uZLNZmK3uZDTXCix5Y8QUjGsFI1oSlqmTlvjPvWgklpxNXma44kVLDGVOIfHIs+Pp4aOGwO
KD4Vwe5TDGZofcMu9JZeFZjq3l5UMXAWxT5LKUhSrgJCISHSlu6kPuuf/Yy7NX2aTlRnLEiUm4k
ilwGO/4dvG1GJWYXcezvCmZ8tcHm8opwnZe+2nJkpW8KwyJZqUduq95PJSEelMJFOPsufc47rN
W+ZwYyflE+UDlpijlyWOaog6G5OlVYY4esbCKqQ4b/68LqCxZXG2lp0yW1KwSZn4UXqqdZbdR/a
krJPoPo2e75EJNUeZPY5BLEGTh4uNIzmN5ljvG8ulO4aFejDBx6kwb3O7x6UrZPC1t6EuWt+v2J1
1URaqAZncR6KkNpZqPkyyeJMNNRuYxo/HLC6TElydl0aNclmPG/qHzlbHeYrBb64Mbuu7bqmEZI
S1XKUFWn6veaDIUMKpn5yuR1yj0pi9PhWeE5TjNPAYYpEkLB+SrY2KI1KND4oYHCCHclb+6Z9
5a3Ug1hJ5x3X5wox6JXbWrwt6mHe6m417DHibiBN7g3l5dIx8CnpbYPP8ontATXI9Q3wlOgJ9830A
JvZ+wX3254l3tH7n1B54Ln7XwLt4q37JrV15Pn7Rmgl50X7fhsF6Zn70c217Cn8RYBB7vH82TMV8
dX9YOYN9OH9yJrh+EX+M5aF2QliG0ld20leQvxZ3Yoadq9p4CIXVmK54soUUhZN5YoRecm16ll

O0X0B67IMVTCV7u4JxORJ8kYHmJoJ9dIEI4+N1DplH0LF1rJAsvYI2TI5Wqm52/ozAl2d3tos2hHI
4d4m7cYF5S4hIXoR6L4bdS5h7FYVtOK97+4QIJlJ86lJq4jp0DZt5zyN0s5jlvBZ1WpYaqRp2FpOzlj
d22pFcg3J3qY8XcKZ4jozXXdV5h4qdSxh6fohfOFt7coY4JjZ8ZoPC4LBzPaTdadz6qFauqt0l53Yp
9B1W5qklRF2KJd/gnN3ApRsb9B38pFfXSt49o5ZSp5594tPOBl6ohmJj175oUb3zVyka5RzEhZPq
oAuWhz7KWupqt0t6GilAx1i52mgZF2bpm7bxJ3aJXZXJV4dplASjR5gl4mN+F6gop3JkZ7doZa3cJ
yB7fKyU5ys7KluChzX619pYp0Laibkwt1BaPlgLN17p8Eblh27ppLXAF4BJWcScp5FZDrN6h6HoxTJ
k17FYd43F5xnME0yZFySbsvtthy9bUppFtzwK9+kgB0mKnef8x1gaRIbZd2hJ65W2d3n5kvSVt4tZ
OjN2x5x45lJlN6w4h12vRxT8qbyEBx98PQtaFynb0Ao0RzZraCkQp0O7ALfvh1lqmYbOV2J6MmWt
l3RZy3SPV4YZZENzV5e5ADJlI6foIU2ZJxG9P9xw1xt8yUtJpyU8UVollzGL2ukDpz7LZHfkZ00q7hb
FB12Kd7WmF2+qAXSJ54GZi3Nwd5O5GTJl16Q4oU3OCAFXQ0ymJ/43Uht+9/s3YSpX1/mXcOkx
V/iHgGgLd/gnj6bkV/jXn1W8V/qXr4SVx/z3v7NvyACXzwJTqAW34K2zF+an1zyNF+Vn2Etnp+Q32
ZpCF+RX3Kkdh+UX37f6N+Z34wbVl+jn5xWv9+xxX6+SL9/A38lN0t/UH9EJQF/rn+J2ZR8/YaHx098
/oXAtRR9AIT/ot19F4RskLh9OYPcfql9aINQbIN9qILWWk5994JrSDN+S4H7Nih+qoF+JM5/EoDw1
+574Y+MxcF77Y3vs6B7+YxZoY18GosDj458R4mzfad8hlhra61814cxWaZ9PoYCR7h9p4TNNdF+
FoOaJKF+iYl81lZ68piSxEV7BJYnskJ7FZPBoE97PZGjjnR7dl+QfLR7vo2HauR8llulWQl8mYmSR
Od9EoeYNYd9jYWsJlI+BYOA1ON6MKGvwt56SJ5gsOl6XpsvnxR6jJhLjVt6ypVxe8F7HJKiahp7il/f
WGt8Co0lRtl8i4pnNU59C4e/JJN9f4TK0355jKqpwZJ5paatr7d5vKKynf158J7+jGF6NjtUeul6jZe3a
WZ7AZQnV997i5CjRnh8Eo0cNRx8l4m3JJ99CYX40hh5CbO7wEN5la7yroB5N6oonON5baWni2d
5tKEwehJ6EZzEaLR6iphIV1R7GpQTRhd7p4+/NOp8MYuUJKl8o4cl0LZ4pry2vuh4vbcWrTB40bF
3m7R5Bqw2ilx5S6b8eS15qKHIZ/d6lZyeVsB6tZd8Ra97RpJWNLR7141WJLJ8TYf3z014WsWovZ
V4bb9Jq/V4fbjqmpp4r7LiiWR48qzbeFI5TKbVZ0d5xqDUVjd6XJrXRVB68JTWNIJ7iY76JLI8BljKze
x4Hc6MvGB4K8eXqux4NcCTma14ZbmuiJN4prLEd6d4/qvXZrR5eKTuVcN6EZ4KRP96qJcoNFh7
R5BzJL97x4mA0F6G4nNhvueGIHRfrX6FYnVhnBSExxZnir2EMHdveXqDpnh6aB+DK3mPVrOCw
XquRWKCYXvONBaCG3zgl5mB5X4czweFOnwovamElXxRFeD83yJmwKDC3zOicWC+n0aeKO
Cin1uZ2SCK33TVhCB3X5GRNuBlN62M7SBY38VI2OBOX+GzbSD2oS/vGyDSYQkqzOCu4ONmf
2CS4MXiN+B5YKqd9uBjIJIZreBRYH8VX2BEIHCRGSA34GCM16AvoEwlzSAAn4DYzDqCx41Juw
qCQYvnqeqBu4qJmNKBVolbh9WA+4g4dvWAr4cjZfqAeYYfVOyAWYUpQ/yAPIQtMxOAK4Mtlwu
AF4lQyr6B5ZXjua2BYpO5qKuA4JGSI7GAg4+ohtSAMY3LdhD/7ov/ZUR/xYo/VGB/toiLQ5p/p4bR
MtR/ooUhlvZ/koNByVyBMp6HuFWAspuYp2KAM5itloV/25YFhcl/jpNrdTF/UZDiZlZ/MI5mU85/J4v1
Qzd/HomAMqJ/G4cclwR/B4R7yAiAl6c1txiAGKOHpJx/mp/alXp/R5xshNt+/5kMdGV+yJW8Y95+rJJ
9U01+qY9KQuB+plwVMnh+plj7lxJ+i4WbxrOAGK/ctdZ/m6trpQ5/Hqb4lGd+y6LGg+h+hZ6hc5V+U
JqMYzN+N5alUsp+OZKRQod+N46YMkx+O4rAlx9+lladxWB/ubhttpN/PrMuo8F+wa3wkzp+a6kEgt
5+laQkcrN96Z9SYnp90ZqMUj191JXQQih91ZEPmHx93oxrlyp9xod/xAJ/csD0szZ+9bsAooH+dbUJ
kiB+Gq9cgeV9zKm3cd99kKQdYc59dp6MUbl9e5kAQc99fpNvMe99jo34lzn9eYhGwqh/PMltsgd+u
sLnoYB+NLxPkTJ91bXMgRR9hK9PcS59RajaYT19KaJtUUt9L5wGQYR9NJWhMcl9So9clzp9Oljy
w/aN6XK9s4SMj3POoyOLOnThkr6KFXXygnOI+ncHckaH7XghYgSG6XIDUbOF8HpsQYGFAXuc
MUmELnzNIg2DYX4vwtSMUHsbsneLEntqoiqJ2nu5kdqIznwQgaaHzHxwcZWG13zcYWqF631UU
S2FDH3ZQRKENH5hMPeDdX7lIdyCtn+DwbSK+4M8sW6J0YLPoTilqoJjkQKHr4lKgOmGwIG7cP
CF3oF5YNYfCIfJULWEP4EoQLGDfIEHMK+C0lDilbGCHYDAwHSJ94tMsESI1lonoCWHtYkDkA
qGw4f8gA2F3YcBcDOFBoYVYEGEP4U1UD6DiYRgQFuC2IOLMG+CPYLAIYuBloHlvyiJlZN3rxSI
AZGRnxGG44+sjxCF+Y3zfy+FGoxJb3SESoqvX6WDjokcT8eC5oeSQAiCQoYKMDqBsoSXIXmB
EIMEveSldZutrdaHWJkNnd6GPpZujfmFWJQAfjeEfpGhbqCDs49UXveC/lOPT0SCWlrTP7GBtoiZ

MBGBJ4Z6IYqAfoQvvKeH4qPurK6Gx6CWnMyFrp09jP+EzJoTfVmD9Zb3beKDLpPuXlyCepDwTs
6B2o39P2SBOosPL+6ArIhBIZx//YVBu2iHa6wrq3+GUagWm66FOKP+i/uEVqAYfHODf5xBbR2Cu
Zh8XbyCB5TFTIWBaZEaPxSAy41zL8mAP4nvlat/jYY2ui2HDbRTqj+F9q95mm+E3qqeitmD+KYG
e3KDHqF8bEOCVJ0DXQuBoZiUTdCBA5QvPryAZo/LL55/34uElbh/L4cMuOaGw7xxqQSFrbom
UKEILFbiceDqawJeoCCyqbDa3aB/KGNXGWBRRpxdTVSAqZc1PmqADpINL3Z/jlz8lcN+34flt6KGh
sR+p+SFcL5mmESEVbg8iOCDZrleebSCg6wKasmBsKYEw9mA+aAGTOuAXJoQPiv/wpQgL1V/
Ro5Mlcx+m4hqt9CVFnJBqFeTNHNPmPGRVnRiiZCPnnV/elGN83aiazqMVnfNXBCKvXj8TNqJK3
oxPcWWho3tyLp2GPHzDIJmE0X5KttOTm3pBp3KRznqWmCWQA3ruiNiOYXtXea6My3vNaq2LRH
xRW5aJwXzeTHGIRX11PW6G0n4ULluFgX7AIG2EJn+LteSSU4H8ppmQIYGdl2OO2YFCiCyNSI
EHeRiLxIDXaiuKToC1WyeI3oCgTBOhd4CZPSGGF4CXLiCE2oCgIEaDjYC1tOGRU4mYpamPnY
iOloeN6Yelh2iMYlapeGyK5IXXaZeJd4UUWq6IFYRXS7eGv4OjPN6FcoL2Le2ERYJhICWDBoHH
s8qQgpFQpKqOz4+QIZ+NHY3UhpilM4xNd7SKJlrVaPqlvYluWi6HZYgIS1aGHlaoPJue2oVRLcO
DulQclBeCfoLWsrCP3ZkXo4+OLpaolleMgJQ9hZmLAZIKdtGJjo/naDiIKo3VWZGG04vHSuGFJlnA
PFCEs4fBLaODKIXnlCyB5oPxsZCPTaDmon+Nop3Kk4qL9pqvhLKKepfJdgWJCZTyZ4qHqJluW
QSGVI9ySniFDYzAPA6DzloYLYiCqYeXIEKBX4T1sGaO1qizoWSNLqTjkn+LhKETg7+KCJ16dS2I
I5nxZtKHNNZZ8WG+F4ZMRsgqEm4+yO8eDWoxcLWmCOIktIFSA6oXcrz+OeLBroDeM1KvjvVCL
LKdbgqmJrKMxdDaIN57kZgGG0ZrEV8iFepatSY6ENJKfO3iC8o6XLUWB1YqtlGSailamrg6OLrgb
nw6MjLLukDGK4629gaGJXqjHc0yH5KPhZTyGeZ8NVymFH5pASRmD2JV6Oy2CI5C6LSKBfowP
IHGANldXrOGN9b+/nf6MULoEjz+KpbQ5gMOJHK6CcoeHnqjXZJaGLqNAVqSE0Z2xSLaDipgrOu
6CSpKvLQWBNY1KIHX/7ofurCKcWHldnZ2ZxHNEjzGXN3RrgM+U8HWKcpGStnazZH2QjXfnVla
ObXkfSCKMWXpfOhOKUHuwK/KlaH0bHxSGfX7Dq1ea9Hm+nOSYc3o0joqV+HqqgDmTw3sgcg
2Rm3ukZAYPg3w4VfWNdXzVR9GLc31+OdKJe340K8SHpn8AHvuFx3/3qouZx4EZn6CXT4Dnjeq
U3oC0f6ySt4CNcZOQnoBzY6SOloBpVZ2MmlBrR4iKpoB8OZqlvoCYK5yG+YDJHuaFJYESqa+Y
z4hZm2SWYleCjTKT94atfwmr1YXrcQSPwoU3YyuNwYSTVTyL0IP1Rz+J74NgOWilFoLYK3mG
YIJyHtOEI4IWqM2YBI+ImpGVml4ljG6TMYynflmRfItGcGqPBYn0YqiNCYi0VNOLId1RvKJS4Y7O
TaHfIURK16FzoQTHtKEB4MSP+aXWpbxmaKU95Tui36SlpK1fX6Qe5Cxb6iOb469YgOMdYzbVE
+KjYr7RpKluYkiOPyG6YdZK06FOIW7HvSDZoQTpvGWyZ5QmLuUa5uNiqWSD5jFfLmP9pYXbv
mN65N6YW+L85DwU9iKDI5rRjyIOlvwOMmGZ4mFK0CEsodIHxSC1oT8peyWT6Wxl8KT9ql9ibq
RnZ7Fe+KPhJtubjuNeJgoYM6LfpT3U1eJmJHNRd+Hxl6uOJGF8oufKy6EPLi8HzCCW4XLpOaV7
az2lrOTmKjTiKaRQaSseuWPJKCzbVqNFZzLYBGLF5j4UsGJLZUrRXGHWJFmOE6Fho2uKxSD1
loZH0eB9IZ+o86Vn7QzlaKTTK9wh5+Q9KqnefOO06YGBISmvqF0X12Ku5z2UjKlz5h9RQmG+JQ
LOA2FJo+IKvuDeYtbH1uBm4caorKVYrtnIKWTDbyThsCQs7CweSeOjatXa8+MdaYNXsWKbaDY
UbmlfpupRLKGp5aDN9eE1ZFvKuaDLix3H2uBUoefoJ6jxHldkxCglXNAhaCda3RneD2ahnWKawK
Xr3a5XfSU6Xf5UM+SM3k8Q5uPj3qJNpGM93vsKXGKg31yHa6IDX85oAOidHlkoOfVnnchSGcPH
pbd8yZZHrgap+Wm3tzXaGT5HwaUImRPXzJQ2OOqH2FNmaMHX5TKViJuH9BHaiHTIBfn2OhW
YBgfWeQIA+hKWbLYAdd2GYyoAMakOVpYAKXVOS+4AZUEmQYYA0QzCN2YBfNkGLW4CZ
KUGJBIDyHaKGolFtnrKga4dCkVSdWYafhBOaSoXLdt+XgYUnadOUyYSTXPWSJoQQUACPml
OSQvyNH4MdNiGksIK7KS6IZoKEHZ2GCYJingKfoY4IkKyckozMg3WZhot0dIKWwlo7aViUC4kSX
I2RbYf6T66O5obiQsOMeIXQNgcKEYTTKSGHzYQLHamFcYNNnUye95UBj+ub75MWgq+Y6JEr
dZ2WJY9WaLiTco2TXAaQ1lviT0OOT4oyQniL4oilNdiJeob0KSGHMIWShdeEx4Q0niKeZJvujyyb
YplmgfyYX5badPqVnZRIaCeS65ICW4yQTo+zTuSNy41nQjWLXoskNbWI84j3KSGGplb9HgKEMI
UEm6Kd6KLejlaa6p+sgTGX6Zx3dECVJ5ljZ4OSdZZhWwOP1pN1TneNUpCNQemK5Y2wNYuleY

rnKRuGKlHqHiaDroW7mrqdg6mvjWSaiKXZgDmXiKlAc12Uwp5SZrqSDJq3WlqPaZcyTfKM4pOw
QYqKdJA1NVOIClZNKQyFuomNHkWDQYZZmb2dMLB3jGyaNawJf0mXNKeWcoGUa6NJZfeRsJ
8OWbiPCZroTXKMf5bEQS+KD5KmNR2Ho46bKPuFWYqwHl+C5lbimLmc7Lc2i4aZ8LI3fn+W7q0
ucceUlag1ZVSRyqNNWS+Ot557TQeMKpmuQOKJuZTpNPCHTpA8KO2FCIuvHnSCI4dVIUerXX
JGiLSnnnNRfEGj33Rjb96gUXWLY6Scz3bDV5qZYHgOS3aWCXleP0GSyXq5MziPI3wtJxeMi33L
HGeJgH+rINyqH3koiFSmbXmYe+yuXoOb5OfNHqiY2Sbv3tHV2WYXHwAS0yVEXzDPyKR3X2T
MyWOtX55JxKLtn+EHHGIs4DDIHGpDH/Sh/OlXn+re5ehsH+Mb0qeNn+QYyeay3+mVzOXdH/QS
ySUM4AFPwWRCoBJMxSN7ICiJw2K+oEfHHuH/YHdk/CoJoZXh32keoWgeyugzITxbuydU4RrYte
Z7IP3VvGWnIOWSvOTZoM4PueQS4LkMwWNO4KnJwmKVYKbHIOHXYKr2ynW4zThv6jsouMe
rSgBYpMboGcjok5YnqZKog4VqWV3odKSrqSsIZZPsKPolVvMvWMmlSeJwuJtoQIHJuGvYOFkuS
msZNGhmujDJF3ehmfYY+rbfOb7Y4GYf2YioxzVjyVP4r0SmmSE4lzPo2PBI5Mt+L+YaZJxqJEIVv
HNSGCoRUkkamHZnChdWiepdeeYye0pT8bXObX5K/YYuX/pCWVd2Utl6CSiCRiYxvPI2OeoplMs
yLboh0JyilfYa6HQiFa4UMkYqlnqA8hSKh/Z05eOOeVpo4bNea45dmYQKXgZSoVWuUNplASceR
C49bPiGN/Iy/Mq+K7oo9Jy2H+4fsHTWE44WukL+IOKaYhEyhlqL9eAid7Z9kbA6ad5wBYFCXEZiy
VNeTw5V6SVSQIZJDPdGNhY8TMoSkd4v6JyiHhokLHVqEcYY4j96k3qzqg2+hPajBdzGdkqSXa0
maGKCeX6KWr5y4VEeTXJjpSOSQLJUaPYONGpFRMliKDI2fJyCHlloRHXqEElatjvikjrMugqOg76
56dnydRanBaqOZyKUnXxCWW6CgU86TBZwxSlap0pfFPUGMwJNhMjSJs48ZJxmGy4r1HZSDv
4cPii2zB3K8fpGuinO8cxeqFHTFZ6Kl8nXmXFih23cbUUCd03hnRhOZ5Hm8Ot2WDHsfL9mSPXya
JMeOiX49GzaK3YAbifGx0Hk1fl6tYHmocuu09nojZ3uk2nq6XDigzXtlUSqc0XwoRgeY6nz3OtqVGX
3WL+GRUH7LJNiNqX/gG1KKA4Elib2wwH+BfiqsV39qcrqn8n9aZ1Oj339oXBmf23+KUROb6X/ER
fmYC4AOOteUQIBqL+eQf4DYJOaM5IFmG2qJQ4IYiWuv3IWpfd2rc4UOcnKnDoR5ZxSi/IQFW+S
e+oOIUOqbDoNbReCXOYMYOs+TflLhL+2PyILBJPOMNoLLG36lm4LziRGvFlvDfYSqrYqjchymSI
mlZsiiNoiPW6OeNoeqULeaTYbaRb2WglYLOr6Sy4VFL++PHoSYJQSLklQeG6KH84O8iK6uZ5H
MfRWqBJAscaWloo6OZlqhkY0LW0Odk4ucUGiZq4pDRYOV34jtOp2SKoefL+qOeYZqJSGK4oVk
G+WHNoRziDat0ZfXfKKpcpWqcTilEpN+ZfehApFtWuydBI9xUCCZHY2NRU6VUYuuOn+RnlnZL+
aN6lgcJTyKSlaNHCOGklUUH5+tUZ3cfBG09JsacK6klJhZZXmvg5W8Wn2chJM0T8WYnJDFRQm
U0l5cOIORGov+L9aNY4m5JUyJv4eeHFeGAYWhhvOs56PCe1uoiaBzb/KkJP0lZM2gE5n+WeacE
JbuT0iYJZP3RKmUVpEEOhKQn44aL7aM54tIJVGJRYibHIOFioYWhi2siKmfepqoKqXJbzijx6HxZ
COfsJ49WVGbqZqfTs6XupcaREqT6JOXOdGQMJAeL5SMeYy9JVGI24mAHKiFJIZ7hV+sMq9ree
an16sMbpyjdqapY5SfXKJfWNSbUp4qTmaXYJoSQ/qTjJX9OZqP05HzL3iMHI4KJVGlgopHHMeE
0lbOf2W6ynNWdLW1l3RJaiiwcXVFX5urt3ZbVTynBXeKSxGiX3jWQNmdz3otNp+ZU3uWLKCU2n
0WIp2QcX61GiOMHICf2K5IXlmdLS0cXnaaiivVXpVX5uqn3rtVT+l9XucSxuhWXxkQOuczn0+Nru
YVH4sLMST4H8slsOPhYBEGk6LN4GDf2W4hX9XdK6zZn9MahyuUX9IX5Spo39cVT2lAX+HSyC
gbn/PQPmb6oArNtKXc4CeLOOTBIEfluWOtYGzGnOKbYJof0W3m4Uldluyf4SiafetaoQkX3Sou4O
9VSSkG4NtSxGfjoM3QPibEoMNNuGWp4LzLP2SRILslwKN/4MDGpSJvIM2fxW20oridFixt4nkacS
soYjpX0an8YgFVP6jU4c3SvaeyYaCQOuaUoXTNuaV74UwLRKRkoSJlyKNUIQ8GsKJC4Pwfte2IJ
CBdA2xC48KaW6r+l2VXvmnSYwvVLuiq4rgSsCelomrQMeZrIh8NtiVR4daLR6Q5YZMI02MmYVh
Gw+IRoSQfoO1h5Ycc7uwdZQhaSCrZZIkXrKmtA6VH6iGI5oSpCdkiywQKaZGosCNsuUtlHLSiQ
TofUI3OL94ZpG1WHmlUbfhC1BJuuc0uv85kmaLWq4paeXlCmMpQwVCqhk5HbSkudCo+iQHSYI
l1zNq+ULYtSLSSPxllGI46LaYdaG5CHA4WSfYS0lKEicravgp4WaBqqbpsJXcSlu5gbU6+hGZVFS
eeczZKLQCiyFY/aNn2TrY01LRCPRlqml5yK6og3G8KGh4X2fNq0L6aLchOvHaL+Z4CqCZ9vXTiIU
pv7UzSgrJigSYKcHJVhP9uXoplpNkmTO479LPiO0ovsl6SKe4j8G+yGHYZLfCkz06vhcXiuxafNZv

6ps6O2XMKk+p+2Us6gUJvQSS6bvZgJP5uXQZRKNh6S2pCZLOSOco0MI6uKHomnHA+FxYaSd
IDCIHQ6api9IHT3YN23onXFV0uyMXbRTeisw3f8RL2nXXII04qiDHqIMl2czHwVKW+Xj32YIIGSW
n8zGR2NTODwdNbBaXnpatu7+XovYRC2fnqFV3exEHsfThKrQHvSROumTHykO76g/32LMpqbvX
6JKbCWg3+Uil6RX4CrGVIMXIHgdQ3AWH+FawW66n9UYTC1cH8yV5awC39MTjOqrH+CRRCI
WX/aO+ugD4BMMs6az4DXKemVmYFqIPOQg4IEGYqLiIK4dSe/aoT/axO5+oReYTS0fYPKV5qvG
INxTjqu4MzRR2kboMTPASfLIMGmvaZ94MKKhqUzYMBISGPw4M+GbaKzoN5dSe+mYpbawy5
KIIYSezqYhCV46uQ4d0TjKo6lbARR2jn4YpPBCEy4WdMxGZNIUfKkKUEISxIVGPCoRdGe6KFY
QidRO96Y+Laum4d44IYPmy9oyOV2Wtj4tIThCoNoobRQSi74kNPAWds4gKMxmYhlcVKmCTW4
YwIYqOSYVgGkWJR4SqdOK9TZSvaru325KvYM6yWpC4Vz+s8473TfCnmo1RROyiVYvKO/qdGo
pRMx6X6YjoKniSu4eOlbyNolZIGpOlkoUfdl28xZnJamy3UZdFYIWxzZTLVv6sZZKQTbinC5BxRM
Ghx45yO92cjlyCMxKXW4qiKoKSK4jUleKNCOcYgtWH94WDdB+8UZ7Lafm215vJYBCxTjJRVpSr
4pYeTV2mhpOHRHehP5ERO6acBI6mMvCW1lxLKNmRpYoFlfmMhYfUGw2HdYXWc5a756O+aX
i2aqBAX5mw2pzMViirbZmcTQCmDpaKRCugxZOYO2ybipCwMsuWXI3YKmuRLosclgmMEYh8G
zyHCIYdcwC7iKiSaP22CqSSXzSweqCeVc6rC5zuTLKlqZldQ+ygXpXvOzybl5KMMquV9o87KI+Qy
owQlHeLsYkKG2OGrlZZ3v5152/OzFB2mHEXua13SXKUpxB3/HPjIHl4s3Usgep5b3Zxb1J6Pne0X
LV7IXj2Six8Cno6N6J8+Xt4JbV+CHzX3Opz3nmSymN0u3oGt+d1I3p5pW52cnrskwN3TntjgKZ4LX
vhhbkF5HXxiW9h6HnznSYJ7I31sNyh8LH3tJXR9Tn532wpyKIMZyKZzJYKltkx0H4lzo/d1E4HbkbN2
C4GJf4R3CIE+bU14FoD8WxJ5M4DASOt6U4CENrx7dlBJJT8p3/+2VhwxIx0xw9x1YsftNFy4onO
op1z6liskH508oeVfnP2BlaJbHB3KYWCWmJ4YISASGd5mIN8NI16z4KHJQp8E4Fp17dvk5W6xYt
wsZOTs2xxy5FvoVdy4l+Bj11z+o2hfYF1HovPa6F2V4oAWb13o4g1R+148IZqNg16OIS6JO17iYL
M1jNulp7vxBZvv5v5sghw5JkEoBdyBZZNjkFzK5Okfl10XZEKatV1pY50WRx3AlvjR3p4WolUNdB5
r4boJPV7BIQv1Lhtv6guwrpu7aRusM1wGKCrnvtxQp0jjUZydJmne7VzspY7aiB1B5LWWI12cl98Rx
V31owkNZt5NYj6JP96j4V30z5tEbFrwV9uQazbr5JvbKhHneFwnKPujE5x1Z+ieuBzHJtjaW90e5cv
WAB175MGRrF3YI7hNWd4yorwJQd6Koaf0c5sibqUv/9tu7UnrkRu56+6nLhwFqqei0txUaWKegJy
m6CBaLV0AJt/V2x1fZaHRkh29pGQNS94blzKJQ551YeJ0FtsJMO0vqptUL2DrQ9ud7dPm6ZvpbF
milxw36uDeTVyLKWkaAtzlZ/LVuV1GJn5ReZ2mZQnNPt4Go6FJRR5jliKzvRr28zlvXVs+8X2rApu
GL8Tmr9vRLhJiZJwfrF/ellxzKq4Z3tzOaP4VnJ0wp1BRZV2SpaQNNB31ZAUJRI5T4IS0rx8VW8iw
RB8iHCKr3N8uHH0ndx85XNUjFB9GnSwetJ9XXYlaUd9tHdjV7V+HnjCRjt+kHokNL5/DXt6JAt/qHz
60R96bHhQv5B6wnjarg57FnlnnI97ZXn8iyN7vHqUec58IHsvaG8InvUVvt9H3yBRad9rX0uNFJ+R
H3PI89+8X6Bz5N4woFAvh55N4D2rLI5p4Cwm1t6D4CHihJ6gYBheOJ7AoA+Z6F7IIArVIV8OIAIRS
R84YAcM/R9j4AJI5t+TX/wzfl3cloZvJ939oj+q1Z4elfpmhx48lcBiPI5c4Yfd/F6BYVFZtp6roR2Vbl7ao
OyRLF8KoLqM6F87YIII219vIFEzGZ2V5L3uy125pEQqgV3b48smO139I2Bh+54hYvgdwx5J4pJZh
154Yi7VSV6soc2REZ7hIWvM1t8WIQ3I1R9MoKQyvl1a5vTucx2A5kjqLR2lpZ1I7x3JJQlht93wJGm
diN4bo9OZVx5NY0BVI96Eoq9Q95674h5MyZ7zlZMI2B8p4PjyZR0oKS5uIR1PqFEp4l11p3OIqx2b
JqXhe93EZdqdVV3ypRJZLJ4nJE1VAp5hl4uQ4N6a4soMvh7UIhEI218LIUbyC1z+q2atzd0mqlZplh
1NaUYIZp1z6EYhP12ep0jdld3Opk5ZAd4FJVdU4V5BZGQQyd59Y3FMsp64ooil3h7woYzxspszebZ
gtd90GrFKpQ10tKw3IHJ1T6d8g/p1+qLlc6p2vJ4cY1J3m5I9Uvh4IJTpQsV5i5BUMph6govkl4J7aYc
pxWJzFr8Xtl9zs7IEo9h0SLNyk1904q30gwp1jah6ct52T6MCYqp3MZ2WUnZ4MJgyQmt5LZLMMm
l6L42II4p7HYgDxARyy8e5s19zX8FOotNz7LrWknd0hLSBgj91L64rcjJ18qfXYhx216GOUgh325tQ
Qh943pUVMkn56I8BI5F63Yi/xsmC626DthKCMG/xpWyCRHFglM6B9XLHhEaBrXQvc9mBbnWbY
1yBQ3cPUtWBKniLQmuBGnoJMfiBHXt6lnqBNn0cxX6BC3cutOKA2XfNpFaApXhvk86AeXkWg2G
AUnnEcXOAM3p9YrCAJntFUj+ALXwZQe6AOnzsMZyAVn2vlkSAGX6LxER/cH+Vs7x/Wn9xo0Z/Q

H9Pkt/Kn9Cgox/HH88clx/F39CYhN/J39eUbh/SH+KQYB/cH+yMUt/o3/IlhN/4H/iwtZ+Kofusmd+JY
cJogx+HIYmkcF+FYVlgZN+FYSvcYd+IYQGYWR+RINvUTJ+foLIQR9+vIJWMQR/A4HDIel/UYEfw
WF9IJBdsRR9IY6yoNt9H40KkK59I4uRgKB9MlomcLd9SYjKYLp9fYd8ULB9yYY5QMN+GIT0MM
h+b4O2IdR+xoJXwAZ8RJbr8N8TZZvn5h8UpQFj4p8XpHTf598dl+ub9t8I42aYAV81YuUUCZ9K4
maQGZ9goefMJt934WyleN+NYOavq97h6Flrod7IJ46nnd7nZsPjoZ7sJgafrl7zZU0bxd7+ZJeX2V8
Ql+aT6t8n4zkQBR8/YovMHR9YleTIfR9tYTCvVd66qnrrUZ6+qX6nU97BalljXt7G55Sfc57PZqrbk9
7bpcWXsF7u5OTTy58IZAfp8B8hoytME188IIZlgJ9R4XMvAh6brJSq/p6gK2UnAl6jKjXjFZ6oqRsfM
x6w6AOBXR69ZvAXhB7RZeCTqh7sJNRP2d8G48fMCB8jYsFIg586Ya1urJ6DLqoqrV6HbU0mth6
J6/Ai0J6OqqXe9p6WqV6bKh6iqBqXWt63JtpTip7TZZyPxN7vZF4L/d8OIyTIhh8moeDuWF5wMLlq
Yx5zLzemdz50LbJilp54rDrew95/6ria/x6L6UAXOB6g58uTcF6+JloPs17bZOjL9V77433liF8V4g0u
uqJwm4Kqyil3G9+m3qH+HD0i9KHKHJhfEqGYHPTbOiFonVOXXyE8HbQTgmES3haPraDsHnqL
02DK3t5IQCCtH08udiH8XY+qiqHLnb2mpGGa3euiwCFunhke5CFEHkbbEmEbn0XPGD2HrRTZ
CDT3u7PICCz3ynLwCCZH2OIM6CAn6UuM6GZH4qqTWFu34umbGFEn4yijqEd34+euSD435Va7
WDWn57XHKC3X6zTSOCbX79PfaCBX9FLryBsn+HIKKB3Y/Ut6WFKoYEqB6EkoVXmK6D94Spi
VODaYQNehqC44N9awiCaLL9W+aB/oKLTLeBpYIIPaiBUoHBL0CBFIFhHuA1YD7tmCEKY3/pve
DI4yWI6aDBlssiGeCgInjeUuCBYinaleBlod+W1aBOoZfTEqA8YVKPVuAroQ4Lk+Af4M2IGmASolg
tRqDWZYXpbyCzpP1InmCQZHT1eBxl/XeFqBUI3raYuA6YwSWq+AIYpFS82AVliDPQmAF4bELi
p/7IUbIH1/s4NTs9eCp547pJOCH5thIWqBIZiFhmKBHZXQd4KAsJMraNOAUJCaWhqABI4YS1x/y
YulPMF/kYk3Lgt/aYbkIJB/LYRtspGCE6Zco2KBjaLCIFCBBJ8mhWKAkJu4dqGAJphbaBR/ypUuW
X9/gZHeSul/TI64PHZ/GluXLep+9oiTIKF+uoVpsVSBmq5joiOBGaoCkxSAkqWihEOAHKGFdaN/sJ
15Zz5/U5mDWNr/DZWcSmp+25HDPCR+rl3sLcN+kloqILB+WYZGsBKBN7ZZoO2AuLFikeuAMa
w0gzR/uadidLR/S6KfZnV+7J3wWDN+ppIQSfN+d5S9O9d+TJApLZ9+OlujLx+Boclrta5r40n9OA
ZriSkO9/37Lhgk9/ZK1lc+p+9Ke8Zcx+k6JGV6x+TpzhSY9+lpeKO5d9+5I3LYF97oz0IMZ9wleur0a
QuG2soG6PUm8Wka6N7HCFgv2MknH5dHWLP3N4Zh+J9nUDV8OIsnaUSWSHdnguOySGRnn
SLMGFM3t/H5yEJn1IrlGPCnV/n5WNvnY2kPGMcHbwgleLMHe1c+WJ9niJZaSlx3lxV1iHnHpkSQe
GentkOtWfY3xrLIOEa312H3CDdH6nrW6Nk3z9nsuMWn0LkD+LH30bgbyJ9H1Cc2CI0X13ZTKHt
328VvaGpH4TSLOFmX55OpCEmH7jLE2DuH9RH0iC1X/SrIOMYYRhne+LN4PLj3SKCYM3gQql6
oLAcsaH04JXZK+Gx4IAVoyFxyG0SGKE0IFzOISD5IE4LB2DGoEMHyaCSYDIq3KLZovnnPiKQ4
qgjpeJHllcgEelBYhCchyG+lc4ZB6F94ZBVheFAoVRSAmEHIRpOheDQIOKK/aChLEHxmBvIH2ql
GKnJOVm9uJgJGgjYKIYY+uf0uHUI3qcT6GSYw4Y2KFToqaVYCEX4kFR5qDf4d7OdGCpoX5K9
uB7lSSHZGBH4MaqS6J8JtRmsul2ZisjleHvZYIfmiGsZOUCHWFr5ExYriEuY7IVPeD0lymRzWC84
p2OZKCHohOK8OBZYZEHOiAlIqIqAKJYKMHmbCITJ+si32HM5xRfXWKGZksb5+FKZYbYgOEN
pMiVGeDT5A5RsyCdo1hOVcBo4qRK6mA7ofdH12AHIUTptml6KqjmlGH2qaNik6GxaJ6fF6Fup6t
bqeEuJr1YTSdw5dVU8KC3JPGRISCBBFOQeBNlZJK4mAholeH25/t4Xjpailhrul1eHfK1ziS2Gaa
i4e1WFW6Q7bbyEVp/UYG+DXpuGUyeCdpdIReSBoJMXOMGA047pK2qAK4rDH3x/YlaZpIClOL
mflkqHL7RaiDqGG68lenWFCqnNbPWEAqSnX8mDCJ+cUqSCH5qkRYaBS5W8OleAgJDaK1B/3
ov/H4h/Glc2o/yXx22TihWVtG8UiEqTpnCWepGRwXINbQOP5XORX6iOEnUkUkSMTHa/RN2KIhh
kN5ml53oZKjaHV3vhHiqF0H3oozWWN3UNIWOuo3Xjh62SQXa4egmQcXeJbl6OqnhrX0WM7nlj
UfCLPnmpRJWJmnt3N1+IA3yUKguGin3BHg+FFX8bonaU2Xw1ILqS7HxphxqRA3yeeYqPRXzYb
CGNkH0iXumL6H1+UaKKS33rRFSIvH5oNyuHN37vKeWF03+EHfiEboA3oaeTvoM5k/uR24LUhm
2P+oJvePOOQ4IUa6CMmYHIXn2K/YGPUUyJclFgRBSH9IE9Nv+Gg4ElKcSFMoEoHeSD2oE6o
MWSzYpZkyyQ8YIPhBGPfYhFeE2NZYdPaxCLwoZrXgOKMIWaUOulr4TOQ82HQoQLNtGF3YN

WKauEmILFHeODR4l5n9uSA5GMkj6QMo/mhMOOX449d3WMtlynalKLFosjXWOJilm0UGyIDlhL
Q3OG04buNp2FQYWdKZ+D+4RxHgeCooNCnuaRVpjUkVePjJaHg+yNv5Q2drGMGJH5aaaKfo/O
XNOI9l26T/qHfYuxQyGGGI2Nm6Et4fIKZWDclYAHimCEIQzneGQxaAckF+PAJ0fgwKNN5oddd
yLkpc5aOmJ+ZRpXDWlCZGyT32G/l8JQSiFmYxuNjqEOonhKYeC9ld3HkaBkoUlnNiQTadlj06Oja
Oage+MyJ/pdN6LIZxmaAmJhpj3W3qH/JWjTuuGhpJaQmCFJl8gNf2Dx4vvKXGCh4jXHI+BKlXBm
8OP5q5pjj+OLKoZgOiMaaXFc+yKv6GbZzOJlZ2HWsiHlJmNTl+GHZWfQf6EvJG+NcKDY03lKVy
CKlobHnSAzoZimrKPkLV2jUiN2bCVgAmMGKuncyCKbKbKZn+lzKIDWjKHPJ1XTeqFxJi7QauEZ
JQtNZGDDI+sKUqB2ls5HoWAg4bsmNWe/22Vi+WcVm8VfxSZr3CWclSXXMIRZcGUu3OdWWO
SUHU8TPiP+XbjQleNt3iXND2LgHpeJ9OJaXxBHNaHXXH5mmD+djHS4i1ya9HWUfpmYXHZxceyV
7ndRZWmTinhDWRqRM3INTLuO73pjQFWMvnulNBeKmXy9J7ulIX4LHM2GI3+Ll6ecRHuMitaZu
XvPfiaXLnwUcYmUznxkZRaSenzEWNWQNH06TIKOAHA3AQCeL335WM/WJy376J6WH2X+3HM
SF54CYlvqbOI1ijmYsoHpfZmWLIGdcRCT0oFgZK+RhYEzWH+PSOEbTECNJoELP/iLGlEHM9iJ
FYEUJ5KHM4FEHL2FTYGNlkaaT4jxiZGXz4gKfP+VTYclcliS94ZWZDqQsIWZWB6OfoTwS/SMZ
YRKP8SKZIOuM7qIbYMIJ4eGILLHMiEsoJ5lY6Zh4+xiNGXEI49fDuUl4zGb9WSRYtfY52QAooLV
5uN1ljMS4+Lv4eSP3+Jw4ZiM5eHzYVFJ4uF8YRSHPeEBYNolMeY2ZaHiBSWapR1e4mT95Jfbz
ORqJBbYw6PaY5qVyWNP4yQSZOLLoq/P0CJNlj5M3iHP4dHJ46FYIXAHSODblRAk+qYRp1hh0
CV3JqlesCTbZfmbnuRIJVEYmyO4pK4Vp6MuZBFSSuKqY3cPvmIsouAM1OGvok3J4yE4lcWHU
mC6IT+kwOXzaQdhk+VZ6C7ecqS+Z1YbZiQq5ofYaWOapb9VfmMP5P1SkqKL5D1PqCIOY4CMY
GGR4sdJ4GEbohXHWmCeowikguXY6rOhVuVAabSeN2SlaLUB6QRJ7/YOSOaps/VVmL05ebS
c+JwpP/PkqHzJBuMvGF3YzrJ3SECol9HYSCHlYwkRKXB7FuhHuUqazheBSSQKhMbAaP7aPO
YEGNqJ9nVNOLD5sdSWeJZZbdPgKHcJKrMsmFg46LJ2mDtop/HZqBzYanjdSmZ222gd+jL28hdg
2f9HCSakic0HITXrCZtXOOu06WpnVVR9qTs3cKPF2Q23jNMQqOEhqnJZeLZnyjG6ClY37gjXGIC
3SDgYKh3nVSdbeernYpagCblncYXnWYingdUx6VjXk6R7SSqXpkPEGP33udMPmNI3zrJZGKi35
XG6aH+3/2jQaj1nsFgSOgsntEdWWdinuKabyafnvWxj2XfnxqUvGUkXz7R5GRu32aPCeO/35KMO
mMUH8MJYUjYH/tG6yHQ4D1jICiz4FYgKufslEQdPuci4DPaWCZg4CvXfGWioCjUrKTpoCsR2OQ
4lC8PAuONyDYMNyLi4EKJYajHlFkG7CGoYHbi/qh7lexgCye0lbbdlSbrYYMaPiYp4VkXZeVs4T
OUmmS2IRORy2QHlPQO+mNfoNaMM6K64L7JYmlelLOG8eF/oK3i22hJ44Ff5aeElytc+qa84tXa
GqX8YofXRuVAYj7UgSSKYfuRuGPcobJO7mM2lXhML2KRYT1JZuHzYQ5HAGFSYONitOgeJRsf
wOdaJJ/c16aT5CTZ+qXUY7HXKuUZI0QUaiRkltxRpyO3lnXO46MRYhIMK2Js4bQJayHN0WHHD
aEqYRNiiCf4prVflWc15hHcreZwZW7Z1CWxJNcXCST2ZETUTmRBY7lRkiOU4y8O1eLv4qgMJW
JLYiZJbWGsla9HGSElIT1iVqfZ6EffYScXJ34cd+ZRprUzoqWSZfnW3WTW5UQUKqQhpJURdqN
04+eOw2LQlzyMHClsYpZJbOGYOYfgHlqDrYWFilCe+qdefKyb8aOocQyY25/0ZciV3JxyWsqS7Zkl
UB2QFZW5RW6NYZJvOsSKz48uMEqlQov/Ja+F0ljpHKqDS4YAh6Kem62Le+SblallcFmYgaUEZ
SSVgaDIWjqSj5zeT6ePtJj0RRSNAJUROoiKb5E7MCuH5l16JayFeYnRHMWC+YZngwWt2m4Xe
Aup52+AbTKl+HDvYluiRHJsV7CemHP/TTqa93WvQrWXcXdsOCyUBHk5LdWQonscl2aNV30dGn
+Kl39XgtSsiHSDd9qooHVdbQOkunY9YjOhD3cxV5Gdb3g9TSSZ3nlIQmWZXqdOCqTA3voLduP
rH1FI3SMcn67GpWJSIBfgpyrW3qvd6SnensCbNKjnHtaYguf+nvMV3KcZnxUTQ2Y4nz2QpyVdX2r
OCeSHn5zLeGO039MI4CLp4A7GqilH4FPgkKqWoCvd1Kmf4CEbliiplBeYcyfA4BSVz6bc4BbTOa
X+IB8QoSUmICoOB+RUIDhLeeOFIEul4uK9YGcGriH3YlogeKpflaodvelo4X8bDKhyoVUYYCeK4
THVv6anoROTLsXKoPtQmKT04OQOBCQl4M9LemNY4MAI5yKSoLtGtmHM4LygX+ou4yYdoik5
4tva7uhEYpiYROddYk0Vp+Z64g1TGeWeodOQiyTJ4ZsN/OP7YWVLeiMulTSI7uJmlQ3Gx6GdoO
xgQqoEJKPdhakQpDba0+gcl8nYLCc1o2lVkiZTov/TCKV4lqQQfqSj4koN9iPV4fMLeeMIYaEl9il+4

VkG1yFz4RagHinfJiEdYmjsZY5asaf4pPxYDKcSZHIVduYwo+3S8qVVI3AQbmSBIvSN7GOzonwL
dyLmlgjl+yIclZ5G5GFQITtf9Km/55cdNejNZuEag6fZpiuX4mbzJX/VUeYQ5NoS1GU05DsQV6Rgo
53N3aOTYwMLcKLGym1I/SH9ld8G72EylVpfxWmkaQqdByiyaDKaVie+p1qXuObXpowVLaX05c
OStuUYJQIQQWRDPeHNzqN2Y4QLaaKp4sul/eHilhmG+OEYOXTfICmL6nlc26ia6X6aL+enqIOXI
ebAZ5BVDuXdJqNSneT/pb4QLmQrJNoNwmNel/mLY6KSIX+I/uHLokwHAKEDoYrelO1a26Tbnew
wW/4ZI+sInFjWqGn2XLZUN+jInRqR1GfW3YbPbmbOnfeNCOXLnm1KsWTKHueIVqPMX2dGXuL
XX/Ieli0GnSgbnafHWBZliq5nZoWpumo3dgUN2ianhxR1aePnmiPceaJ3roNDyWi3xEKuWSJ32ul
XyOQH8mGZ+Kd4DCelWY7Hp3bmuuV3raZHipyHtDWo+ljXu/UNehX3xSR1edQH0EPdKZMn3O
NFCVNN6vKwKRRX+aIZqNbYCRGb6JrYGkeFyx7IAnbkOtXYAWZFKo0oAKWm+kl4AOUL2ga4A
qR0ecVIBhPdCYUYCmNF6UY4D9KxqQfoFhIbOMs4HcGdql+4JweCSxDoXFBgysg4U+ZB2n+oS
6WkCjv4RGUJefloPnRyubhIOiPcOXiYNINGOTPlM0Ky+Px4MVIIdKMAYMUGgSISoMqd+uwUltTb
cOrylpWY8inQYIZWfGjCIhmUFOe4YeIRvia0obFPaKW2YYLNFIS9YVcKz2PFoS/If+LR4Q9GIKHh
YPSd5ivp5DcbXKrJI9cY3qmol3bWaqialxmUBaeQ4slRsiaNonGPYOWP4iONE2SXldjK0mOeoZJI
ieKpIVKGpiG2IRldyavFJZebQOqk5RTYxCmEZJIWUmh2ZBST8Kdtl51RoaZqlyzPVWVvor9NDW
Rz4lUK0qN7Ye9IkWKE4ZAGtSGQ4TkdP6ulZvHbHKqFZk3YnmlkpapWMChWZQ1T0udMpHZRia
ZI4+bPQ2VLY1kNAiRS4s6KzqNaokjllWJk4ciGwaFx4VOdf6uJKEma9Wppp4UYeKII5sCWdeg6Z
gMTtScv5UuRcaYrZJuPMaUto+0M9qQ1Y0GKyeM94pulmGJJftGzGFXYWodVOtwaZwa0GpRqL
VYWOkxZ86V8Sgipu7TnCcXphXRXWYSZUSPImUUZHUM7SQco6kKxiMIYuSImlqIx4icG1SFBYX
zbfG9Cm9TZKi4F3CJW5GzGnHNUqCuG3M/SdupH3TQQUekKnaGOKyFTXhSMBeahXo2J76Vw
nwmH16RCX4hGISMiYA4bjK7uXUHZN22zXXBW7qx1naJUsGs3HeGSfmn6XigQWmjA3ndONieL
3s0ME2Za3yIJ/qUs34cH5SQDH+UGLaLmIEkbly6kHqHZP21pnrNW9GwsXsgUterv3unShCm1nx
KQYWh/H0POPydMH3yMHyYc37uKC6TxH/tH8OPLYDpGOGKxIH4bnC5jH/nZQi0qH/CW9SvtH+
pUtiqwn/AShSI3X/yQY+hCoBCORKcR4CmMKCXIoEbKFuS84GZH+yOa4leGQeKC4K3bnK4poU
wZQCzxoScW8Wu1oQTUsqp5IO5SgqlAYN4QY2gNINSOR2beIM5MLqWz4MtKIGSM4MslBmNsl
M8GTyJU4NgbmK364pTZn+zCYIRW5euFohXUp+pJoeHSeakRYbQQXWfe4Y1OROawYWnMM
OWGYUIKKCreoStIFOM7oRFGZKlhoPxbjS3RI9sZLKyYo3uW2utbox4Uneof4sxScWjoloDQV2e2
YjxOQmalYfuMMqVeYb5KLuQ14YOIleMRIUxGeGH0YRubeK2spR8ZGWxzpJ6WyOs2JCduJen6
Y7FSY6jCo0gQTOeRYuaOO6ZjoogMMGU54i2KMeQRYdYIK+LrYYIGiOHN4TZbXa2NZI0Y/SxTJ
b2WrGsUJSEUdCnYJJQSTaigJA2QO6duY46OLyZA4xJMKSUX4pkKMOPvoiQIMiLKlBLGlqGtoU
zbPK1xZ5cY3iw2ZtkWj2r2Z3UWem55XKSnyiBZM3QKedPJDEOliYh45XMIST5Yv3KLmPR4mul
NuKtYd4GomGSYV/bGe1YaMmYwKwdJ+sWdurdJw/URCmgZkVSJGhnZYIQGuc05McOFyYHpA
2MGmTfo1fKLKO44qolOuKVYgNGrCF74W+1DZyOGqkwm1zLmxwsLF0JW49nvt1IW/5jU92InGv
e693LXNdag14TnUFWGt5hHapRt56vXhRNUF783n0JG59TXvA0jFv5HRNwlxxDHUvrvZyNXyRn
WZzYnbyi+Z0kHfXenl1wnjCaQp3CHmsV5p4YHqWRj55vHuCNM57E3xtJC18h31r0Flt5n2uvthvN
n2urWZwhn2xm/tx033OiqNzl33ueWR0eH4TaCN14H48VuF3WH5pRbF4036YNGd6SH7MI/V71n
78zq9sQobnvU5tr4Ylq/1vG4UvmrJwgYSFiYBx6oPkeGtzV4NMZ1R02YK1Vjx2bllhRTV4AoGPNA
55k4ENI8N7OYByzSFq3ZAJu9tsW45aaqZt2lyvmXxvU4s9iG5w0YnXd4FyVYh9ZpJz7YckVaJ1m
YXMRMR3RoR5M8N47oNBI6d6poHdy6NprJkOum1rOpaTqUtsxpQcmEduUZHmh19v4l+8dppxd
42eZdJzlouDVQp04oluRFh2oIddM4t4WoVwl7F6HINKyitopqlbuRZqPJ7YqBdr0ZuWlZVta5iThm9v
CpWddc1wsZKzZSdybo/QVIN0P4z3Q/p2DoolM1t31oeBI715ooSbyLFnzaset75paKcOpuJrAaL/liN
spJ8yhYFuTptwdQJwA5e6ZH9xzpQPU/5zrZBwQ5x1i4zaMyx3Yol3l8d5OYXKx0FnHrQAtl9ova8b
pZhqWqo6lQJsAKWvhldtr6EsdCxva5yyY85xQJhDU3JzLJPfQzl1Fo+AMvh2/YtQl9B44YbWxc5ml

bzPtQxoMbctpGZpy7GMk/Vrcqw8g59tJKbwc2hu5qGpYytwxJxtUvFyupc8Qt50sJIOMsl2pY0KI9d4
lYfDxGlmK8WGS9lnvr9Jo2NpULkAkxNq+LLWgtxrayucsJudKaKYqJwWaB0UoVyWZptQpF0WZ
RsMqF2W46YI914VoiQyH14hmoqt6J4/2vyptt5eG2/liF582+HhXZ6eXFLdNx7DnMJZD57uHTJU59
8eXaKQxl9P3hOMn9+CXoJltd+83vzxut2UnM7ti92+XQspYV3nnUglON4Q3YdhFh48Xcec+h5qngj
Y296eXkvUvJ7XHpCQo98Q3tWMhp9LXxflpx+Mn2CxWN0aHwJtMh1NHwopD51/XxNk7x2wXyR
g1R3j3zYcwp4an0jYrR5WX17Ulh6Wn3fQhZ7YH5BMcF8Z36almd9hX75w9ly2IS7s1lvYQOou10
n4NnkpF1e4Lwgg52X4J/cip3UYIWYfx4WYG3Uch5dYFhQat6lYEKMXN7t4C3ljp864BUwldxg41vs
fdyelv4oatzaoqlkXJ0WYISgVN1UogmcVR2V4cFYUx3dYXsUT54qlTbQUh53YPLMTJ7FYlJliJ8W
YGpwPJwZ5YbsJ5xapPhoGNyaJGrkEtzZY+5gFB0bY3QcHh1g4vzYJd2sYogULF39YhXQOd5OY
aPMQJ6foTfljB7yYMEv49vcp7Qr1pwfZvUnz9xhJbj0VvjJYjf2pzoZN2b7R0xpDVX/Z2A45BUDR3V
ou7QJF4qlk5MNI5+lbYIkB7SoRDvihupKd8rhJvtKO4nhVwwJ/3jjxx0Jx9foJy7pkMbu90HpWoX1R1
aJJTT7d2x48PQDx4JovPMLB5g4i1Ik163lVjvMtt+7AlrL9vEKt0nNjW6bnjR1xMqK1fYlyU56Lbhxz
ipprXqh03JZaTzJ2R5JYP+B3so5YMIN5Hlp3llh6f4Zgu2ptdLh9q3ZuiLMzm6Nvla3rjBFwqKj7fKFxz
KQQbVlzB58sXgh0YZpXTTrZ11pWPP4x3S5DHMF1w4walMj6MYc/uhRtCsDUqkxUF7r1mqRvH7
UMizBwM69Ke95xWKmMbLRyl6PUXYFz+J4vTk91dpiZP0V29ZMHMDZ4d42SImp574gAvRV+7G
m4rSB+52t/nUF+4m1KjXB+628Ufbt++XDgbih/D3KwXo9/OnSITvR/e3ZmP3V/w3hHL9yAF3obIVm
AhXwju9F8xnJBq/p863M/nDZ9EXRBjH59QXVNF0V9dXZhbXJ9r3eCXfJ9/nixTmt+YnnsPwJ+y3s
mL4Z/P3xPISN/yH2Zup9663qDquB7MnrAmzd7eXsCi5t7yHtefB98GnvEbMp8dHw2XWJ843y8Te
99ZX1UPp197X3nLzp+fH5oIPN/Hn71uUZ5boKvqZ95zllxmg96KoG3ipV6jlflezt684EebAd7YoDIX
MN76IC9TXV8hICiPKR9JoCFLvd9z4BhIMI+h4A4t954LlrwqF14mYmxmOt5Blh3iZF5d4dyelh574Z6
a0Z6clWSXCZ7CoS4TP17u4PrPe98cYMDLSB9L4JTILV99oF2top3IJNFpw93mZFJl7F4EI9SiHh4j
42XeWJ5FIVqanZ5plpOW3x6TljBTHt7DldCPZh7zoXDLph8loRRIMd9YILAtTd2OZukpdl2upjtlpd3
OZY4h3x3wpO/elZ4U5FUabt47478WuR5ply3TAd6clqCPUx7PIhPLnd8DoYyINt824Pws+J1dqP5
pJ51+6CCIXd2f50Lhnp3D5nVd6R3qJauaPx4TZObWkp5DZCdS5J5442xPP56uorLIN7lof4lOt8aY
UAsPZ01Kww01h1XqfVldt15aOyhV12d5/Jdqt3E5vwaCt3vZgrWaF4hJR4SxN5ZZDWPKt6Ro00Lix
7LYmIIPI8CYXvsUI0TrRPoht02a9ckxN1YKprhFN186XJdcJ2kKE2Z2h3PZy1WQR4CphISp149ZP
pPF554l+HLgd60osylQV7tobCsAZz4LxNoP50abblkh07rE6g3J1gavPdP52IKZxZsN20KEmWIB3
o5vxSjp4lZbOPB15iZGrLel6hoyWIQ57cod4scSFj2lkorKE/2sqk7iEdGz0hM+EBG6/dg2DmHCSZ3
yDMXJwWOuC2HRXSl2Cj3ZGO+uCT3g7LVKCIxorH/CCBnxSsLKDfnF3obuDGNkEkyCuHOTHA
2Ca3SpdWWCIXXOZu2B23cEWHCBpHhKSfGBenmdO4+BWnrzLQuBS3w+H7+BTH2ur7WBs3k
2oNSBcHmUkgyBL3n0g1eA/npndMmA0XrmZmiAqXt3V/6Aj3wcSY+AgnzUOz6AfX2KLMyAi340H
5OApX7yrqGARIDbn8+AF4CQkRp/6YBHgoN/yoAWdBN/sH/zZc5/nH/iV4F/mH/iSS5/pH/vOvd/uX/
9LJV/4YALH22AEYAdrWp/DYivnrZ+7YeokB9+zoamgaZ+vYXMc1N+soUDZSx+r4RMVv5+voOkS
Mt+3oMHOFR/BoJuLGh/QYHeH11/glFFrCt+C5CgnYJ9+Y7njvh95Y0wgJ5934upcmt934oyZGh96
YjQVmB+Bld9SFR+MIY5OmV+YoT3LEh+pIPEH3N+5IJ/qup9L5idnFx9JJYxje99F5PEf7B9GpGG
cZt9JY9aY7h9OY1FVdJ9X4tDR+p9IYIToiJ90ldmLC1+GoWPH4p+WoOgqaZ8daCWmy98b51tjN
p8Z5pEfrZ8cJdRcL98gZRXYwB8nZGqVT58y475R319CoxcOd19TYnDLBF9n4c/H5595ISiqGx726
hymfV72qSGi6R71aCdfZt74Jz9b8d785lyYjB8EpYBVJl8RpKmRwV8jY9dOZF82lwUK+99NljXH65
9f4WEpy97WbA3mMJ7W6uein57WacGfJB7Y6K0btt7dp52YWx7l5pTU/57z5ZFRpV8HpJKOUt8c
Y5MK9B814pQH7x9KoZLpf166rfal65677KyiYd67q1/e656+KhrbhV7DKNsYMD7Lp6lU3x7a5m+Rj
Z7wJUJORB8GpBUK7V8ilugH8d84ob1ppaMTGkhmGiLRmrZiISKQmyYfFSJUG5nboelXnBDYPi
Hb3lxU26GiHQoReyFrHYqOISE23g1KueEl3pCHp2DeXyJpa+KY3DPI5yJfnHYiaSlmXLle7+HxHQ

HbmgG8XU9YI2GlnaLUxKFWHfpRZuEmXIVOD+D5XrHKq+DTXw1HnKCwH3NpNyIs3gfluCH53i
CiQCHHHjrezWGYnl2bZWfQnoRYCqE9nrBUr6ESHuGRVKDonxdOAKDCX03Kn2Cjn4LHkyCG3
75o/WHUX9Hlg6Gln8RiEWF3H7gepWFMX7SbQ6Ein7UX7mD6n7sUmKDUUn8TRQuCxx9IN82C
RX+CKIKB5X/CHiqBiYAOovSGloanISWFdYXBh3SEx4Thed+EKIQ0bHKDKlOaXzaC/4MWUfmCe
YKeRLyCAIlyN5aBk4HNKi+BRyF3Hh6A9oEioeSFKI42IBKEiYymhmWD54sceOmDUonla5qCw4i
JXoGCPYdhUWmBwoZIRFKBU4U+N1WA7YQ6KHiApYNGHjmAVYJMoMOEVJXRkwaDvJOVhW
6DlpFdeAuCIY9bateCD41vXd2BkoudUOaBlInfQ/OAuYgyNxuAWoaMKgWAF4T6HIR/xoNdn5WD
oZ1lkeuDD5p2hGaCepeJdxuB8pTaagSBcpJDXSyA+4/JUFqAj41IQ4+AL4sWNt9/14jNKe9/mYaV
Hmt/S4RRnmyDDKTdkLyCgqE2gzeB8p2TdgOBapo+aQ2A65cDXF+AdZPnT7qADZDjQx5/so3yN
px/YIsCKdR/K4gYHn5+44UlnT+CkKxCj5SCCqf2ghaBfaOsdPiA9Z+raCGAdZvFW5yAAJf/TyN/mp
RTQrR/RJC6NI1+940fKbp+zll9Ho9+ioXgnBqCLLOlloiBp66wgSGBGanQdBeAkaUVZ1qAEaB2Wv
d/m5v6TqN/N5ecQlp+5pNUNih+n48MKaR+e4q6Hpx+QlaAm7KTLmkSjnGRfmrigU+P1my0dEGO
XW6FZ2aM5nBmWsmLc3JcTi6KDnReQZiluXZsNSKHcHiHKHyGQXquHT+Fln0UmviRY3BjqcP
OnGSgLuORnK7c8aM5XPpZwCLiXUtWnOKMnaKTeWI53f3QVmHq3l0NO6Ge3r7KFWFaHyKHS
SEYH5lmkaP03dojSyOV3fygDKM4Hh/c1OLIXkdZqCKUHNMWiOJEXqSTaKH3ntsQSGGuXxZNM
GFoH1OKDKEpn5JHqyDtH9lmXmOiH4zjHONGn4vf5CLr34scsmKcH45Zi2JOH5YWcOIC36MTV
aG7X7PQOI4H8gNJmE3398KBOD+3/oHPiDHIBpmKKNaoUqi62MB4SEftyKpoPgciyJclNXZaelR
YLhWVSHJoKBTP6GGYIsQKmFHoHjNHGELYGmJ/6DWYGCHPiChIFql8GMd4w6isWllor+ffKJz
onCcVilolibZO6HfYelWLqGalaOTIaFY4WgQFSEcYTANEKDh4PrJ/aCtIMwHR+B3lJ5Is+LpZNkie
GKWpGfR6JEI+icJmH6Y3XZEaGzYwiWC6FwlqHTBmEw4j8QAiD2leCNBeC84YUJ/GCI4TBHU
OBR4NvlcuK85qTiOuJsJgCfDalbJVvb8WHSpMBY4yGM5CqV5SFKo5wS6GEM4xJP7SDToo1M
+mCbogrJ+eBooY6HWOAx4RKIMaKXqGjh92Jl55ieySH5JshbsmGwpgVYq2FrJUiVt2Eo5JPSxSD
r4+PP1OCz4zgM7KB9lo4J9aBMlecHX6AW4Ulk7eJ36ightGlqqS/eh6Hb6DdbdeGTZ0uYdiFNzmZ
Vi2ELJYISo2DOpLFPvaCXo93M36BilwsJ8WAzYjhHZR//4WtkquJc6+Chd2lQqsSeUCHCqaZbQu
F6KI7YSSE0J34VZmDx5nZShuC1pXSPqiB/pHeM1KBLy3xJ7eAeooBHad/soY6kPaaNmkbhKWX
82rpeHwVtWy7bFyTpm6RYHaRm3B6VMYPIHJ9SR+No3SLPXSlynanMe6J/XjVJjmlSnsYG/2Gq
32bkGOYiXaChB+WX3FLd/+UOnJ+a/ySP3O8YCeQSnUPVluOXHZ/SOqMgnf+PuiKvnmPMcyJB
nstJiKHbHzdG/KF4X6/j9OXFXbKg6CU/Hdkd5KS53gDa6GQ/ni2X92PHHl8VE+NRHpcSLmLf3tNP
SCJz3xSma6IK31jJg6Gp36FG+iFLX/MjyaV3X0/gwaTzn1TdwuRwn1qay+P4H2VX36OCn3SVAC
MQn4nSH2KkX6HPPel9373MZSHan91JfyF+YANG+CEj4DBjnaUzIPrgmKSxoNQdnWQwYLTaq
yO54JwXw+NG4lhU6WLYIHoSDeJvoG4PMilNoGUMXmGt4GAJfSFU4GOG+uD8lGvjckT34pvga
SR5olndbKP7lheafyOGYdmXnWMVlaDUyiKoYW4R9eJCIT3PlIHh4RDMVyGDoOfJfuEeqMbHB2
DQYKkjP2TEZErgOmRlo+HdQKPMY3gaVyNZlxOXeqLpYrSUraJ+olwR4GlaIgbPE6G7YbVMUKF
eIWgJgOEE4SLHEuCpoOBjCOSYJfugBiQeZWfdDuOjpNPaKaMxZEjXUuLCo8PUjOJY40YRx2H
1osvPAyGYYIYMSG8lePjgSDjoXjHHOCiYREi0CRzJ6SfyqP6puec0iOApirZ8aMO5XqXIWKgJN
CUZKI2ZC6RqKHTo5AO7qF3ovVMPaEcol1Jf6DGlcmHJSBsYtsilCRS6UmfjuPbaGacl2NiZ4Ozu
2LwpqyW8eKB5dxUPaIX5RRRiyG1pE/O2qFaY48MM2EAotCJfWCsohPHLGBUoV+iWgQ2qugf
WGPAqeGcZeNIqNmZjeLW59mWyaJn5uCUHKH9pfBRciGb5QSOyeFBpB0MKqDplziJe6CW4IS
HMiBAoX6hmGhaWk9ewKemmr3b8mby2y6ZKWZG26XWbSWbnCLTv2TyXKcRD6RQnS5OXyO
2HbmLuKMfHknJBuKPXuCGtilFn4dhfOf2G/mep+dG3ENb3KaXnl9ZGGXv3OKWX6VJXTxTtKSlN
Z1RB2QI3glOWSNzHmuLtKLg3tkJBWJWX0yGtyHQn8xhYeeenZEekGby3bfbyKZGneEZCCWiHh
OWUqT/3ktTqmRhHonQ/2PI3syOU2M3XxRLsSKpX1/JA+lJ37DGt+Gh4AvhQKdT3xqeciap3yFbri

X/XypY8aVcHzyWQGS8H1QTm6QhH3FQ9KONn5GOTOMBn7VLriJ4n91JAqH3YA2GuKF4oEUh
HWcRoKdeUaZpolvbkGXAoHJY1+UeoGNWKqSA4FmTiiPo4FWQ5+NY4FMORSLQ4FNLq2JLoF
iJA6HMoGeGveFPYHwg+ybYYjVeLKYYyfobaeWLlcAYtKTq4Y4WC+ROoWFTcaO4oTrQ1iMqoR
XOOmKkYPQLqCIGlNbJCSGg4MLGzSEh4LNg1Oal48jeB+YB42jbRyVclwmYIOS9lrNV8GQiomL
TW6OOIhjQxeMB4dFOMKJ9IYzLpWH5oU1JDiF54RbG2uD5YOSgqGZ6JV2d3OXXZNVbHaUy5
E6YbuSU49QVzyP7I2ATQONnovMQsiLcooiOJGJZYiHLoOHw4b+JEWFXoWUG5qDW4Q/gd+Z
VJurdqaWy5jza6KUOpZCYPiRxJPMVpGPXpFwTHiNEl8yQmCK5oz/OE2l3YrXLmSG2li/JEiE44a
6G8KC54TTgQ+Y0aHTddWWS56OatOTvJtNYDmRR5hFVeqO4ZVYS/CMkZKLQfqKaY/JOAulZI
0RLkSGZYpmJEiEelfHG+OChYVTgD+YXqfjdRaV3aQTaiWTUKBEX5mQ3JyjVV2OdpkdS36MJp
W6QaSJ/5JkN9SH/o8bLiqGA4viJEiEHoiwG/+CM4W+e+OopmmLcX+IHWtLZ0ChnG0UXQqeYW7
vUwKbKHDkSTGX9XL5P1mU4XUfNYGR6HdZK9eO+3mklgqMIXwFGciJaX6be6qnIW/ecUijqnEU
Zw2gOXJSXOGdCXOnUuOZ4HUWSRyWw3aIP02TwHhJNX+Q2HoBK92N+3vllhaLNx2fGdqljH+
ge2yl0HXocQ+iY3acZtqe/HdYXLib1ngwUsSYvHkgSQaVsXosP0GSv3tNNXyP5XyEK+ONGH3Ili
GKZX8aGeqHyICoexOkRvDcLyhSnnv+Zo+d63xAXHmax3yfUpGXtH0TSN6UtX2iPyiR0349NXSP
DH7pK+eMUX+jliqJrYB3GfeHHYFleq6jslGecF2gVIFaZjic+YEdXC2Z2YD/UIGWzID1SK2T14EEP
wiRAYEbNWeOSIE9K+uLmYFxljul/YHFGheGcYlwekui0od1b+6ffYa5ZcCcKIYBW7+ZDoVeUfKW
BoTRSGOTF4RePtaQSIpyNU6NIIOsk+6K54NDII6ISIMSGI2FtlL0edaiDo1Wb32ewlwRZVSbcYr
OW1uYW4mjUZyVWliQSCGSb4eXPqiPpYaoNTEm9oXHK/GKS4T2ln6Hp4RCGpyFDIOieUahY5
M3bvCeG5FdZMya0l+HWt+XvY3VUTGUvYw8R8yR1orAPmuPEIIONRSMZofpK+qJvYaVlpSHG
YVbGtOEfYQ7eKOg05j9bkOdjJaXZBiaQpQ2WjqXMZIAUKGUMY/kR1iRSo3lPhaOhYvwNN+L3o
oGK9aJOYgrlqCGnIZhGwCEBYS8d/CgUp64bY6dD5vMY2aZxpjjWZeWtZYkUBOTtZOARuWQzJ
D6PcGOCI59NKqLZYwKK7+lxlmmlqiGLodOGyeDn4Uqdzqf36RcbOmcoaDnYtCZXZ1zWQ6WTZ
oIT5yTTJbyRoWQYpPhPXqNn5DaNH6K/o3fK62IYor4lq6F0ogcG0eDSoWFcaWwAWnpaDSrwW
uwXumnkW1+VZ6jw29WTH6f93FMQ5GcMXNkOp2YhnWUMa2U9HfbKPGRa3oulB2N7HyNGNO
KnX8Scaeufm/oaC2qUHEuXt2mL3J6VZOiaXPUTHeerHVKQ5Ga+XbkOqiXX3iYMcOT23pkKQ+Q
Ynw4IDyM934RGPkjt4AlcZytLnWhaB2pC3ZsXsik9Hc+VYOhN3gjTG2dhnikiQ46Z5Xo/Oq+WWXt
3MdWS43zIKSmPd34flFaMHn93GQ2l64DocXSsEns3Z/Sn+nuQXp+j63vvVWCgLnXHtFGcgnzrQ
3yY632QOquVbX5HMeGSBn8PKUCOqn/gIG2LYYC+GSSIOYGwcTyrHIC9Z7ynC4CiXmmjAICL
VS+fRYCITCmbnYCbQ2CYD4DHOOp6UmoD+MeaRPoFBKvSN7IGQIlUkq4H0GU2Hh4JqcP+qQ
YY1Z3CmOIWoXhOiM4UeVOCEfISeS+aa2lQ1Qy6XT4PmOoCT4lOhMd6QilNoKWWNNNoM7lLqJ
74MhGZyGw4MYcKypgYuuZx+lf4qfXcShf4mTVJedy4iUS6iaLletQwCWp4bhOmSTPIYhMdaP54V
uKXSMloTHIOWJSYQxGeSGFYOycDyo25EhZrKk3Y+IXVug4l3zVDedL4x2S1WZkosSQsCWD4n
KOjSp4iOMcKPVYdfKXiMBYY9lQWlt4UrGiCFgYQ2b7ioTZZ5ZiSkUZRgXMegVJJJU7GcpJBRSu
GZBo5yQmSVgoyvOfaSG4r2MZuOzYlIKW6LglenIRmINoYSGIOFBYSIbyCnzpvlZY2j1ZkuXDOF2
paXUyucKpQfSm6Yi5HAQgiVBo+AOBORn41GMXKOU4sWKWCLC4j2lSeHxobhGn6EnlUDboKn
XaEAZQCjaZ3eW7afcpAUrbw5fDSg2Yl5ThQbuUnJlgOXuRNo9IMVCN7Yy3KVVWKqlOEITSHaY
eUGqKERYVSZ1m3ZmqMXq+y3WwqVjquTm3WTe+py2+wRculSXGrPdKgznPLNdScb3YJLdmY
KHhgJhaT63q9HkCpTH0YF+qLw3+IZ4+14nA1XtuxZ3FbVlqs5HKQTgWoaHPzRd6j9HV1PeufjXce
NfabPXjnLgmXA3rKJkyS1XyuHnGOs36GGBaK04BvZ7C0kXWZXvGwHnZOVmironcSThKnL3glR
e2ix3kaPf6ecXpNNhSaL3uhLjKWAH0PjnyR4H56HpuN0n/WGD2J/4FAZ8mzg3rsXvyvFHs0Vmeq
mXuKTg+mJnwUReuhw3y4PgSddn15NiaZPn5PLIOVHH83JqWRCOAgHsCNDYEHGF6JRoh7Z8
ayjYAgXvKujH/8Vleprn/mTf+IPIACRd6g3lA2Pf+cl4CENi6YZ4DfLmuUToFFJsqQRoGxHuuMUIIk

GJCljoKkZ7mxtoUuXtKtUYSkVii03YQjTdKkb4PLRbmGFIOLPeab04NINiaXqINMLnaTkoM+JuqPi
YM1HyeLjIMwGOeHwoM8Z4uw+4o+XqSsmlk4VfuoJog8TaijvldvRZafZla6Pc+bKIYgNh2XAIWUL
n6S7oUTJweO5ISYH1yK4IQhGTaHD4O/ZzmmWo9JXler9o2+VbSnhlxATWejHlr+RWCexonVPae
ajlJlNgWWalfHLnmSWlbSJxaOT4XIH4aKSoT8GXmGdoQwZs+v05QxXemra5lvVUOm9JA7TQKijl
6GRQqeNozqPWWZ/ltqNdeV2YnzLmGRzYiGJxaNx4cjH6KJxYXEGbGF9oSOZIKvW5kLXXCq8J
aSVNGmdpQnTJ2iDpH8RLSdt4/qPSKZfl31NaeVW4wFLkaRUooeJxKNUIhHH7iJUoZ3GeCFioTe
Zc6u8p3MXP+qhprRVHGMc5fmTEeho5VCRGydTJK5POmZD5BONX+U8l3nLi+Q6ouLJw6M7II
GH8ql8ocQGgiFL4UgyXBuZ2WSulBvvGfQp6JxD2oPls9yU2wxhglzn25LdVR093BeZJ52ZnJnU+I
36XRnQ0d5b3ZrMoV663hplqd8i3qRx4xrt28vtrttS3CApf1u3XHRIVBwW3MZHLdx3HRldDRzYXW3
Y7B0+ncFUyt2pXhPQrh4UXmcMiN58nrqInp7s3xOxclpZ3hstRprMHjfpH9s83lWk/lumXnfg4lwQnpr
czRx8Hr6YttzsHuKUoF1fnwcQjp3TXyxMcx5En1OIJ68H3wxCpndoF+s51pZ4EXoyRrUYC0ksBtF
4B8gnZu3oBKcktwqoAgYhxyiH/1Uep0d3/LQct2ZX+IMYB4Sn+UljB6RH91wqNlzYp1sjRn2IIAodtp
3lgQkZprvYcUgXVtn4YjcXFvhoU8YWLxf4RTUV5zioNrQWZ1IIKKMUJ3loHKliJ5qIDtwTBkY5NKsN
Nmg5FMoJBom49SkHZqko2XgHhsi4vncJtuiYpBYLlwmoidUNVvyIb8QQh03YVIMRh29oP2Ijl5GoJ
fv8JjJ5wgr4lIVplcn2pnfZabj3Jph5QYf5drJGhb91tql81YB1vz4zPUFxyBopzQLZ0PlgiMPV2aoYDI
B4noOzvlRiGqTorj1kU6FbnkNmhp3PjnBonZqHfrhqupdLbyFs3pQaX4NvF5DyT+RxYo3ZQGVzq4
rLMNB17of0ImR4NITlvPBhPK2JrOxfakrnQlltqTTjV1n1aDSfc1p+5zbbllsLJjsXuJuc5UIT2dwzpEy
QBBzKY1IMKl1gYnllnR324Xzu4xghLYSq6dix7D5m+VIA6vljGBnJqclfPRpU6JrbaRrjp23Xk1t4ZkQ
TvRwS5R2P8Jyto/kMIR1lot7loJ3jobhujZf7r58qoJiL7jGmu5kabMji4pmkK13fD1ow6frbQprB6JjXc9
tZZzuTpNv3JeLP4ByVZlxMGV00o0Clo53Toevvmp0j2Vbrk11bGeHnkl2RWm4jmN3EGvgfpB35W
4FbtZ4ynAkXxh5xnJBT1p613RcP7N77HZ7L+x9AHiRIS1+NnrXvM9yB25SrNhzhm+qnPh0MXEFj
S91LXJlfX92MXPJbe93QXUyXld4ZXafTrt5nHgQPzh61nmCL5d8DnrulQR9Y3x3u0xv1HcAq3dxG
3eKm7lyW3gajBNzfHjFfll0pnIzbR513HomXal3JXriTi54fnumPsx52nxsL0x7NH0ulOB8p338udBt/X
+OqhhvZX9Qmnpwxn8YivpyBn8Me5VzTX8HbFB0oH8lXQF2Bn8RTap3fn8hPm14+X8zLwp6cn9
OIMF7/39luFxsZlgYqMRt5ocVmUdvYIYYietwuYVSeqpyGYSVa4lzhYPjXF91BYM2TS52mYKQPh
V4LoHuLtV5woFilLZ7ZIDDtwFrDJCSp3hso47OmA5uMY0QiNRvnYuSebhxEYofar5ykoi2W7l0Kld
UTK910YX8PcJ3eoSqLrN5IYN0INF60YlitaZp4JkPpj9rhpaMlvZtl5QOh9xun5HReOBwJY+eaghxu
o13WyZzY4tcTD91H4IOPXh22odlLpZ4k4VolOt6UINjtEpo4KGApQJqkZ45ldpsOZr2huFtwJf8eAdv
VJUMaVJw+JloWpNyso9TS890foyPPS92S4nTLnh4Foc/IQJ54YSEsvtoCanOo79pwqW8lKdrcaG
yhdJs/54Fdx5umpial9wSZbIWfZyEJM/S1hz7Y/EPOF1yoxPLIZ3qlj8IRV5g4WbsatnVbIDooZpEq
08k4hqxKh7hNZsVqQVdkVt95+2Z9pvrptgWWRxgZcbSutzbJLkPJl1WY6xLjZ3SlqZISV5NIZhsGV
mwroVoWtofrS1kpZqMK9QhAJrxaohdY5ta6T3Z0FvKp/XWOlxBprNSo5y/5XVPF10+ZDiLht2+lwKI
TJ48Ychs4560mUYpEI7Mmc7IR97kmljhh1772uKdzR8UW20aHV8u2/jWbB9O3lWSup9z3RMPEB
+anaHLXB/C3i2H8x/ynscsjp4Xm2Foxh49m7jIBJ5jHBFhSl6GHGxdmN6pnMnZ8V7OnSqWrt74X
Y3Smx8mnfNO9p9WnllLSl+H3rwH6Z+/XyesQB2QHWnof13A3ZKkxV3w3bxhEt4c3eydaN5Jnh7Z
yN533ISWJR6qXo5Sfp7hHsvO4B8ZHwkLop9Sn0MH4V+RX4Hr7J0eX2woMI1X32ckf12P32Ng1
V3CX2jdM931n3DZnF4qX3yWAN5kH4uSYx6in53OzF7iX6/LLJ8jX8IH2l9o39Vrl1y84XJn5Rz8IT5k
Oh06lQvgmJ1yYOVc/12rlIMJZcB3l4KNV3V4l4lcSSB5qoG1Oud6woFSLld734D7H2F9CYCbrRhqx
43qnllvYxij71zyIrggVh0uYmVcx1r4hZZP92rYctVtl3wIYPSKx45oT+Op56D4PwLGx7PYL0H4B8
clHnq9FwkJYTnTBxr5PTjrFyx5GWgGpzyI+Sckl0zY2eZFJ13Yu9Vk93AYnsSER4OIgsOI15cYZxL
FZ6rYtQH5176YMXqodvnp4znAFwxpszjZ9x55g3f3dy8pV7cXZ0BJLRY6J1lZA7VcJ2VI25R9x3m
4tlOht44ojeLD16LoaQH7d7doQnqUdu0KYwmslwAKJrjHRxKJ6tfmtyOZtEcl1zUZfsYuF0d5SpVSI1

tpF6R2t3DI5bOdR4Y4s/LCF5v4g3H8x7FIUVqAhula4VmZ1vVamfi1xwgaUtfXFxlaERb7Rys50GYi
1z4JkPVJp1KZUsRwN2jZFZOZN38o2FLAV5Xom+H956wYXnptdtjLXWmI9uxLDlinBv8qu2fJ5xCq
bYbv1yLKILYZVzXp1TVCJ0sZizRqt2lZQmOVx3ko+bK+95DI5bH+16flabqLeBSGTimkqBJmcAi/y
BBmkjfdWA8GtJb9qA3G14YhOAZG+zVE6Ay3H1RoyA2HQ+OOaA7naOKw6BEXjZHn6BS3tep5p
+7mzbmU1/Am5Dix9/Fm+vfRV/KXEjzZ/PnKmYYI/VnQ8U9l/eXXgRil/qHeQOJV/4HIEKtSAKHrw
HI2Ag3zFppZ84nR8mGN9H3U6ilB9XHX7fGN9knbpPbp99ynexYQt+BHIkU29+SXmqRdB+l3rBOE
5+73vZKqF/WHzpHj9/0H4SpYd7LXwAl2d7i3wZiWI75Xw1e5x8MnxpbfZ8gnysYH18130AUv19N3
1kRXI9o33VOBB+GH5JKNR+nn7BHiV/MX9FpGF5tIOmllx6KoMMiHt6m4J1esx6/IIFbUN7YYGIX+
h7y4FYUoZ8Q4EYRSB8yIDkN9R9VIC1KIF98oCUHiB+mIB0ozd4eltolTp5AooFh2V5hojZedR594e/
bGx6a4a2XzR654XDUfd7clTeRLd8BoQHN5V8ooM2Kj59TYJ2HkN9+YGuogd3aJM3ICR3/5E7hm
p4kl9AePR5Do10a6p5kou8XpJ6HYobUXZ6tYiNRFI7WocQN1x8A4WaKi58uoQ6HmR9boLNoNF
2gZsAkWZ3lZhlhWR3u5WReAl4QpMWat540JCvXel5a15jUPF6DYwsQ/I6v4oJNjJ7dofSKhx8OYX
kHoB89YPOn6J1v6KqkdL2Z58xhD53Bpu+dv93kpiaafV4JpWOXSI4xJKcUft5c4/BQ456Moz5NuN
69ooyKgV7yld2Hpd8klSunnF1Gao8kLV1xqYWgyh2aqH1dgN2+J4kaRI3j5pqXHN4M5bNT85465
NHQyt5tY/UNqh6hYxeKe57ZojpHqt8OoVznUI0irGpj6x1O6zugj1146gvdS92dKOkaGB3Dp8wW9t3
tprbT1d4dpajQtd5S5J/NnZ6Jo5aKdx7E4ozHrx78oYcnfqH2WS2kGuHRWbEgvyGsWjZdbaGH2sB
aKaFi204W9iE9m+BTxKEaXHVQIOD5XQyNa6DbXaZKMqDC3kCHUaCvHulnReFpWxNj6GFPW
2ugk2E1G8UdSSEZXCsaC2D9nllW3ODh3PSTryDHXWOQgqCuXdYNXOCYnkoKJ2CJXr3HSm
B93zynD2DsXOAjuKDbXRAgaqDJ3UHdJ2C2XXwZ76Ci3brWxWCPnf9Tm6B8nkiQcmBrHpbNT6
Bc3uWKHaBV3zPHRCBR34mm1aCE3qBjhSB6HqvgPaBuXridAWBf3s2Zz+BSHudWqyBFHwZT
hqA5HylQYmAun0/NRCAnn3fKFSAoX6HHPmAq39BmmCAq4GwjTWAIE4gC+AeYDFc1mAUYB
/Zq2ALIBNWjOADYAxTbt/84AiQUN/4oAfNOJ/3YAIKDP/9YA7HPiAEoBZmWB/d4kUjDJ/dof2fy9/b
obccnR/VIX0ZeZ/PoUhWY5/L4RmTTI/JIO7QOZ/IoMeNK1/KYKJKC9/TYIFHR9/boGBmFB+bpCGi
zZ+fl67fkh+gYzzcaR+c4tiZTF+a4nnWPI+aYiITMR+bYc8QJN+d4YDNH5+i4TRKCD+t4OzHUN+3
YKQlZr9j5fziit9p5VyfU99tpL2cMN9sJC8ZG19sl6bWFZ9uYyZTEZ9ylqtQDt93ojXNE19/YcHKBt+
NIVGHWN+YIOBlhx81p9BiQx89pwJfC99DJjZb7t9CpX9Y4Z9D5M9V5I9HJCeS7R9Mo4XP9d9U4
ulNBV9flkzKAi9wYbDHX1994RTIPp8OqZ7h/F8YKKdex18ep7FbsB8eptCYql8gJfcVuR8kZSaSyp8
rpFyP3l82l5dM+B9C4tFJ/h9XlgiHZN9noULk9p7t62Uhu574KkkejV7/aSybet7/qB7Ye58BpxjVxk8G
phzSrZ8PZShPyp8b5DmM7V8rl0nJ+l9CIIZHaZ9VIWok42OiGS3huSNTmbael6MGmkDbgWLBWs
sYeCJ8G1nVfqI3G+4ShiH1nIUPjuG33R9Mn2F9Xb0JoeFIHl3HAKEYHw4ks2MdmvyhjiLZm1zecq
KWG72bYqJYnCAyXulbXlgVaeHe3PbSdSGIHWmPgKFuneAMIGE7XIIJmqEOhtUG/WDIH1ykg6
Kp3LPhZOJtXO5eT+IxnSmbRiH6nWjYR6HENayVVuGP3fYSZWfDhKSPdCEtHpfMiuEAXu1JICD
aX0TG+mC336TkUSJJnl1hN+ISHnWeKKHa3o5bJOGohqPYLGF23ssVQGFHnvESU+EbHxrPZ2
DyH0fMgmDMH3fJjqCs36xG9+CP3+ckHWH0oA/hCCHB4AHd/WGO3/Ra/6Ffn+vYDKEyX+hVJm
EH3+pSP+Dgn+7PWWC9H/ZMeiCcYAEJi2CBoBKG+mBoYCfj6KGqYclg0SF9lZcdxaFPIWSazaE
i4TXX4WD44QwVAyDR4OhSJKCuIMfPRqCN4KqMcOBvoJCJjCBW4HyHBqA94Grjr2Fql4rgmyF
Bly+dk2EWotOaoGDtInyXumDGLitU4uCileDSC+CBYzPNeBj4VfMaKBIIrhJjSAw4N9HEmAYYK
ejcaEzZU6gYKENZMXdXCDlpD1abqC+l76XjuCY40YUv2B3ltUR8OBY4mjPI6A94gEMX2AkYZvJj
OAPYTvHHB/4IN2jM2EFpwmglGDh5lUdG2C8JaFaMyCVZPwXWuBxJF3UISBQI8eR0KAzozYPD
iAa4qkMVCAD4h0Jit/x4ZMHJF/dlQwi8eDeqL+f4GC8Z+Lc3SCX5waZ+iBxpjnXKOBn5XQUbGAtP
LcRsiASY/9O+d/740tMSZ/nlpfJiJ/YleLHK1/GITSisGC9qm5fpWCcqWycqKB5KGnZyiBTJ3KW/uA
v5oLUSiAQJZzRmB/2JLzO6J/hY+GMQJ/O4waJhp/CoilHMR+y4VciUGVWmTJfYKTj2brceuRyWk

TZoCQIWtCW0mOfG2FUE2M2m/hRU+LTnJKOIKJ2XTBL3ilcHdKJGqHHXnoGtqF5HzFiKGTa2ur
fPSRw200cXKQHm7CZh6OkXBcWvqNCHIMUA6LhXPZRR2KFXW2OiqLuHejL1yHaXmfJFYGM3
utGtuFEH3tiAGRvHI1fGyQL3MxcQGOonQyZcKNK3VFWq+Lu3ZrT9KKU3erRO6l/Hj8OgaHt3pgL
0SGf3vRJFCFYn1UGt2EVX78h1iQU3iHe9SO1nkCcHyNV3mAZVKL7XoOWISKjXqwT4iJOXtoRL
aH+nwsOeGGz3z+Ly+FsH3eJEaEqn7bGt6DsH/yhquPEX7xezONo37eb+mMNH7OZNSK1X7TW
eqJgn7sTzKIQH8bRHaHFH9RObeF/n+SLxuE8n/kJEWD/IBZGvKDDoDhhf+N9YVoen2Mm4TSby
6LPIQ8ZCuJ6lOwWViloYM4Tr2Ha4LZRB6GS4KCOX+FQYI3LwWEPYH8JFWDTIHgGy6CXYHS
hUKM/YwDecmLsorRboSKYomdY5KJF4h7WNSH2oduTIKGslZ8Q86Fm4WWOU2EmoS+LvKDn
oP1JGWCsYNKG2WBwoKshHCMJ5KreQCK6JDLbcaJoo7sYuWIX40zWD6HKYuTTdmGBYoQQ
3SE+YibOROEAYc1LtmDDoXdJG+CKISdG5SBPYNrg5eLc5kweB6KO5asbN+I+5QrYhGHvJHhV
4WGiY+xTUSFaY2gQwSEY4ueOMuDc4mpLriCiYe8JHCBroXbG7yAzYQPgg+K15+kdzmJppyHb
AGlaptYUaHLZaMVtOF/JPHTLSE3pEjQpmD3l6OOIWC9YwGLpeCFImCJG+BRlb+G92AboSegc
aKUqX7dmiJJaJMa0WH7Z6dYJqGspscVj2Fg5e6TDuEZ5R7Qj+DapFPOEuCiY4xLnuBsIsZJG6A
7lf8G/iAH4UXfxScU2TtdEaZ/mb+aaOXqGkZXymVY2tPVN+TIW2dSs6Q5nAlQLOOy3KANpSMz
HUHLJuK3HejlnGJAnpaGcyHR31MfpWahGt3c9iYS2z6aUeWD26GXuCT4XAxVKaRuXH1SqKP
mnPXQJONI3XHNn6LrnfJLJCJ1HncInOIFHwHGduGbH5hfhaY8nGwc26WznKwaPKUpnO4XpuSi
XTjVHCQdnYkSniOcHd+QHSMg3jpNmqr3pmLlel6XvzlnSHQH2XGemFq39ffZKXm3e2cvWVhH
g5alaTaHjEXkGRVHlzVCePTno1Sj6NWnsPQEuLhHvyNIKJyXziLH6lGn3klNWghX8GGfWFAYBE
fQWWZ33EcnKUXn3EaA+STn3LXduQQ335U9KOSX46SfuMZX6SQBuKoH7tNjel+X9SLHeHW3/
LloCF1lBrGhOEWEffHyVWYPbceCTXoNgZ3eRWoLpXVGPWoKNU1yNaoJGSZ6LkoIXP9iJ2IHt
NhGIO4HNLHCGpYG/lp6FH4HTGliDooH3e+SUa4oMcU6Sflj9Zu6Qg4fzXNSOjlcKUvCMpoY3SU
mK14V+P5yJKITNNe6Hk4QnLGuGA4OTlrmEglMeGpiDAYK4ezeTm5BGcKeRtY6VZk6PxYzsXE
ON1Yt2UnKL9YoYSOWKLYjXP1OIhoefNcOG+YZyLF6FcYVVlsyD84RSGs6CelNhen2S6pZhb+S
RCpQaZyWPHZHcW4qNMY/cUdKLVI33SGSJ4wvPvWH7YpwNYmGaYi7LEeE6YcRIteDdYVzG
vuCBIPyebWSUJxtbxyQc5mZZMKOiZbNWteMoJQ/UTaKxpHMR+eJAo95PpmHZl0vNVCF54rtLD
CEclizlt2DCYZ6GyGB04RueOuRyaJcbmaP8Z79ZB2OC5ujWkCMJJh/ULKKTJV4R32lipKVPkyG
8l+8NSCFeoztLByEC4ooluOCrYdfG0GBUYTXdPCjUmUzax+gR2dOYXmdQ2lxV+eadWuoToKXq
m35RVcU6HBpPBSSRHLqMtWPvHV/KcKNQXgmllEklXriGNOlkn3OdKWhmWtpatyepmz/YT6b
uG6dV7iY+3BQTl6WR3lfrTiTnnQNPAeREHYNMtSONHgiKcuMNHpFIJmJ4Hx5GO+HsH7SdFSg
GHFTapadN3JtYQSaWxOQV4qXq3TKTjuVCXYcRR+SdneKO/mP/XkLMtGNm3qfKdOLRnxAlKm
JB33xGQIG53++c/mezncQakCb+3e0YLWZKXhgV0eWgXkmTgWT6XoBRPaRZ3r0O+GPAHvxM
sqMthZ9KdqKdH4XlLeISX9KGR6GOICTc5GdqHzLad6a4nz1YFqYG30mVvmVen1zTcWS7H3TR
MWQd35KO8KOIH7GMr+L5H9LKeGJs3/gIM6Hk4CVGUWFi4FbcyqcplKEaW2Z7II7X+KXMoH0V
oyUmYG/TWiSFIGfRH6PqIGWO5KNWoGTMqmLJ4GZKemi+4GulPiG2oHdGZSEzolbcrWbwIhN
aPuZFId6X3aWZlaqVimT1IXwTRORVoVNRD2O84TCO2eMrYQ/MpWKgIPGKe+IWINblR6GN4M
IGduEJ4LFcila+o4aaHKYVoyxXvKVrotOVbCTI4oQTKyQrljqQ+yOTofeOy6MDobbMneJ6lXiKe2H
xYT2ITqFp4QcGhiDmYNYcYyaU5PKZ8yXs5HYXkaVDY/rVROSho4rTCOQEYyEQ4CNtYr5OuCL
eYl2MkqJWYf6KeCHPYaIU2FJ4UeGkqDloPWcN+ZwJluZx+XJb2XZ2UgJSEVHmR/JI+S56PiJA
SQxaNLI4EOpSK9lv7Mhyl2on5KdCGxYgAIVqEuYYIGnWCvYRCcC6ZQZ72ZoCWqZv2XQ6UCJj
7U/eRhpYpSy6PE5NzQr2MuJDeOIOKg45PMfWlbovJKcOGX4INIWWEXlBsgpmCaoScawSqB2
WCYiymr2enWX6i/mnTUNSfn2wKSFGcRW5eP/2Y83DUN5+VvnNgLz+SpHYDJw+PIHizHr+MjXt
uF/KJvn5JavWowGtnYhilF20QWWehem6/UMCeKHB8SEOa4XJUP/iXp3RON6WUhnZg1ORfXil

Jy2Of3q5HuKlKHzuGBul0388atinSHEGYfqjr3l6WUmgH3N1UKmc23TASDSZpHYiP/CWgHeiN6
mTcXk6L2OQd3roJ0iNi3ybHwGKSX5QGD6lBYAXaqmB3Z/YcqifXdEWRie+ngPUH+buXjsSBKYi
3neP9qVdHroN6KSdXwBL26Pjn0oJ1+MtX5YHxyJ7n+TGF2HUIDcamuk6XvsYYuhbXw+WNyd9X
yXUEqaun0CR+eXk32AP7yUh34VN5SRIn6wL3KOvX9UJ3SL8IADHz6JM4DFGI2GnYGSaiqj6oF
KYT2gfoEwWISdE4EZT/mZ4IEJR6OWwYEOP4qTvYEqN3eQ0oFML2yN/4F3J4mLNYGrH3OldY
HtGOSF2oI7adajDoavYOufrYYUWDSCTIV8T7CZITwR2WWCIR5P12TC4QbN1yQJ4PEL2eNWY
N3J5uKkYMzH6OHzoL5GTKFLoLQaWmiUowTYH+e+lruV8ybnYnMT1GYdojERxSVY4fTPx+Sao
b7NzSPioYsL1WMwYVIJ6OJ/ISnH8iHO4PwGXWEnINQaOmhs5FbX/aeXY+3Vz2bBI4XTs+X34y
VRqWUzYsqPsiR1YnaNveO+YiQLzSMNIIdMJ5+JdYYPH+KGuYTTGa2EI09aFahKpaXX2Sd1Z
R1Vq+afpJXTk+XW5BYRjeUSY5xPnKRUYykNrmOd4rcLxGLt4kZJ5el/YddH/aGSYWgGdyDuoQZ
Z7ugtZu5XtudYpkSVjeaDJZvTeOW6pPuRduT2ZGHPimQ4Y89NoWOCYz3LvOLTYq3J5ClmliDIA
aF64ZQGgODZIRmYQqxmmYQWQGti2gPUS6pdmmodSYGIY2xbQfahVm63OpGdVXE4Mx+Zc3P
UK6uVrHaKJGiR8XIEHQeOQHv8F6K237CYTWv8mugWSCr9G0tUUKn727JSY2j6HCUQgGf7X
J9OqGcBHSIMzuYNHawK9aUe3jwJJyQ03swHTuNOX1IF1OJ6H+kYU2ufXD0WS2qjXITUUWmk
3NCSZGimXSiQgiernYcOq6a2XezM1OXGXInK/uTbnsxJMqP1nz4HWmMU36wF4CJEoBvYVGtS
nYrWSepY3bhUTilcXemSYOhenieQf2dlmntOqyZzXrWM1+WGnwPLBiSgH1VJPKO+X6aHZCLiX/
dF6ilWIEkYUCsM3tIWRKoWHuUUR6kb3vwSWqgfXx9QemcoH0gOqKY4X3YM2SVOX6YLC6Rq
n9eJRaOL4AoHb6KyYD1F+GHolHIYS+rOYBHW02na4AvUOUjjoAiSTqfo4A9QcGbzoBtOoiYFYC
0M1uUdlEBLDmQ6oFUJTMnclGpHf+KBYH/GD+G1oJbYQCqY4VJWL6mnoS2ULuiyIqWsq2e5Y
PXQZybFoOUOnCXZINoM1KTyINELEKQqoMnJVeMyIMLHjmJWYLSgJSGJYLbYLGpr4pHWHG
I7okxUHKiG4gqSMuePldeQWWacoapOkiWw4YLMzuTLIV1LD6PqYTIJWqMMoRXHmelwoPEGN
2Fj4NIYEypG48jWAWIWI2aUAOhg4wfsGidp4riQRGZ3Ym7OgeWmlisMw+Sm4ehLCqPHlaaJW6
LqYWWHoiLPoSLGRmFEYOkX9ConpPwV4+k15H0T5Og/pAFSASdI45UQL2ZWoy5OcaVrYs2Mu
KSGom0LBKoolg1JW6LMYa7HqKHziU8GUyEpoPyX0yoNpigVx6ka5YmTzSgj5O9R7CctJGVQH
aY7I+EOY+VP42NMryRrouWK/6ON4miJW2Kzle7HreHbYXTGXaEToQzvshqaWCWrrNsD2M7nr
JttGXijr9vSWhpft5w6GrqbxJyl21jX0Z0XW/PT3p2N3ltP754EHSPL8952HbniNp7xnlovPZnVGocrQ
NpRmvXnSVrNW2TjVptD29DfaVu7XD4bgtw0XKzXm5yyHRITs90znYRP0F20nfBL4B4xnlvIMR62
Hs5u0JkqXM2q3Rm2nQYm7xpB3T+jBlrEXX1fl5tH3bubSFvM3fpXa5xV3jhTjZzhnnZPtJ1s3rVLzt3
z3vbILF6BHztuahiYHwgqf5kxXwsmmxnl3w/ivBpVnx5e41riny5bEhtwnz+XP1wC31ATa5yX32APn
B0sX3HLv928n4mIKF5SH6DuCVgaYTrqJ5i8YQvmTBlcYN5idlnx4L0ep1qHYJ4a39sd4IGXFpu4Y
GOTTBxV4EWPhlzyoCpLtB2LYBgIKN4ooAHtr1evY2Np0xhYlwnl/dj/lqTiMdbolTebFo4YgdartrW
lboxW7xt3oXETLdwcYSZPctzAYN7LrV1g4KJIMI4EoF+tV1dQpYsphBf+5PqluBirZGth9NIOI+reOFn
xo20ag5qWlviWzFs+YnfTE1vpYf/PYdyUIYuLp907oSSIO13loLVs/1b+p64pNRew5uylcphhpivhuFk
JpXueBJmyJM3aWJpcZCMWqdsKI3oS+Vu7YtSPURxsljMLoh0aoZ+IQx3LIQJsqa5KcZo5VduqN
KIKJgiJ+BheFjNJwMdzll5ZifaK1on5U8WhZrapHjS3huRY6YPP9xIItaLm1z9ohNISZ20oUYsVIZ+K
9eomJc1KrZk49fqqZZhPniX6lpdm9IHJ4BaAZn5JnfWZBqwZXLsXrtr5HFPL9wol3KLIRzkYn6ITx2
hoYHsBIZMLd9oU5cEbJbkqde6q01hCxbqKg7dcZkb6NIz3lnRZ5dWR9qMJmFSsBtMJTAPlpwNJ
AILj9zPlt6IU52RobVtGJwbGCYpRFxoGMkldty0WW1hsVz82g9d8l1ImrEaO12YW1GWgx3t2/CSy
h5IHl4PFp6i3SyLV577ncjH399dXnEsrtth2l0o5ZvAmsvlltweGzuhZpx1m6ydsZzO3B7aBN0rXJKW
Vh2MHQZSph3w3XoO/B5WHe6LRx65XmGH2x8kHt1sThrAXICojhstXL1k1NuY3Puhllv7XUBddxx
gHYYZ09ziHcyWld0zXhTShh2h3l5O5F4QnqjLOF59nvMH1t7w30Jr8Ro13pwoOZqt3qdkIRskHrT
g4FuQHszdPlv93uaZpFuxXWHWB5zi3x4SaJ1anztOz93SX1oLK15ln3xH0x7Dn6Arlxm84LPn6Bo9

Yl+kQJq74G0goNswIFedB9ul4ESZdtweIDQV4xyaoCPSTN0aoBTOvR2aoAgLIld4YoAHH1F6an/p
rQtlVosUnmFndInlj9hpi4iDgX1rdYd7cz5tZoZ8ZSFvY4WIVvdxclSZSMRzioOxOq91pILTLHN3t4IW
H3p51YFKq71j7JNYnTVmIJFRjs9oTI9RgJRqTY2PcnZsVovYZHxublosVnNwkYiJSGJyxIbyOnN0
9oVnLGN3IYQFH6F5U4KNqnBispuNnAhk95jljCJnM5Ylf6lpRZOPca1rYZEgY9Ztjl69Ve9vx4xnSA
FyElOH0jH0WYfnLFJ2nIXXH8J444OuqTJhqKOcmTzj+KAUjKJmP5yUfqxoXJlscNZqhZZPYyRswJ
M8VWNvDZA5R5pxbl1EOfhzy4pYLDx2KlePH954hoSrp/Rgw6uQma9jGqdXi5VlaKMIfcJnjp9JcA5
pwpt3Yn5sCZevVOBuaJP3Rztw2pBPOb5zToytLCd1wokIH/V4NoWKpsJgAbNcmKZiXa6KirJkr6m
2fPxm3aUYb2ZpGqCEYfNrbZv7VHNt2pelRuxwXpMoOY5y447PLBV1bYqPIAI384ZKqgp2iWCCm
4Z3R2LjSB4BGWCfuZ4vmgHcM55fWqQYUf6RW0fVO17Im+uRvV8EXI9ORh9BXTRKwx9+Xdb
Hjt/DHocqKRzw2jTmkl0w2qPjAx1wmXQffR2s24fb/93pm/3YjV4nnHcVF55pnPIRoF6vnW6OMB72
XevKtV893mbHip+Lnuup15xV3DZmSZyjHHfiw1zvnLqfRd023QQb0N1+nU/YZh3HnZ6U914T3fDR
hh5i3kYOHJ6y3puKqR8D3u8Hht9aH0kphVvPnjFI/1wn3kYigVx/HlxfDJzO3nvboB0fHp4YpV1wHs
PU1t3FHuwRbZ4dXxaOC552n0IKnp7QX28Hg58uX59pNJtBlC3lthu74BUiQFwa3/3e09xyH/Kbb5z
Jn+qYFV0iH+ZUtt1+3+QRVd3fH+QN/B5AX+XKlt6hX+wHhR8F3/Lo5xr4oiJla5tf4eOh+ZvFoZ/elZwi
YWmbOlyAITbX6RzflQgUIB1CINwRPJ2ooLKN7V4PYIsKk952YGmHkF7fIEYomRqi5CVIJRsO47L
hupt5Y0GeXhvbYt5bCpw+on8XwdyjYiSUdV0Mlc1RJI14IXnN4F3kYSiKkV5QYN8Hmp69YJloSlpY
Zh7k3VrIJX3hehs2JN3eJRucJE3a2dwDo8JXmdxtlxUVdza4rpRD51MljzN0x29ocFKjl4vIU2Ho56
glNXn/doYqA8kk5qK5z5hNBr7Zm/d5ttkJbXapFvOZQBxbdw7pFBUM5yto6TQ910j4v0NxR2aolbKi
h4RobXHqx6H4REnslnhafjkTNpVqPzg8lrH6ALdrNsyZx1acpufJjyXRRwPJWEUE9yFJIqQ4N0AI7f
Nt917ouWKhd34lhXHsV5zIUUnaxmya9jkDVon6rbguxqbqZSde9sH6lCaSft2p3FXlvpvJmfT+Vxip
WSQzhzh5GXNrN1ho2eKgl3iomtHtp5hoXGn6Z81mBpkfN9GWLhGF9XmVZdv59q2fbach9+Wp
nXmp+SW0AT8x+pm+eQtB/EHJBNe5/gnTrKNJ/+3eRHQuAjnpynn96LGhKkO56rmoMg397MWvT
dix7r22naSV8LW+LXEN8rXGCT119NHOEQnV9xHWQNap+W3ehKKd+/3mtHPx/t3vlnW131W/Uj/
x4inDvgq15PnlPdYp55nNGaJF6jnSKW8h7OHXgTvh75XdHQiR8mHi8NW19U3o1KIB+HXuqHO9
+930+nF911Hc7jwZ2sne1gdF3jXgzdM94VHjNZ/N5G3l2W0V55XoxTpB6tHr4Qdd7invMNT8aHyl
KF99VH2GHON+TX56m0h0E365jgl1En6GgPB2DX5ZdAx27X5SZ0530H5cWr14t356TiV5o36hQ
Yh6mH7SNQZ7k38MKEh8m39aHOx9rX+vmjFymIZKjPlzsYVuf+t0xISXcyV1uoPrZoh2s4NRWht3s
YLMTad4tYJUQS95wlHoNNV60YGEKEN77YE2HRt9DIDomRRxTY3mi/RyeYxafwFzn4rRclZ0qlI5
Zdd1tog1WYl2yYcJTTZ34oXtQN95AoThNKI6JYPeKEB7U4L1HUd8f4IGl/BwMZV7iudxapM2fgxyn
ZD2cXxztY7zZRt00o0GWPB194s0TL93l4l1QI14VYfJNHx5jIYIKDI6zISYHW18B4MEltJvQZztic1w
hJnwfPlxwJb5clZy4JRTZEZ0B5HFWEJ1OI9TTDP2c4z2QDF3ulqqNEp5AYhiKC16VYYkHY17oY
PilbBucqRHilLvKche/Zw/p0Bb51yJZmyY3xzU5Z9V550jZNIS7111pBkP9x3LI10NBt4h4qDKCB57
oeQHah7SoSkIJZtwKt6h79vEKc9extwWKMBbthxg57/YtNyuZsXVxRz+5dQS1R1UJOkP5R2tpAM
M/R4IlxzKBZ5l4jTHb57AoVKIV2DPGBViH6DE2K4e8OC62UkbzqCwmemYuiClGo5VtiCYWzfStG
CNW+OPtCCEXJGMueB93UIJraB8HfMG+6B/nrLIHaAtmfXh7GAw2mPexKA0GtPbqeA0W0rYnC
A0G8dVnaAzXEOsOaYXNCPo6AyXVpMrWA1XeXJpaA+HnGG+KBK3wik5B+eG7zhut+snAOem
1+6nEybiB/E3J8YgJ/OnPZVhx/YHVLsjd/gXbRPIN/o3hoMol/0XoEJnqAGnuhG9iAcH1ekqV8mHX
Xhh188XZjeb19SHb3bY19i3ewYYI90Hh7Vbh+FXlcSel+V3pLPhp+m3tHMmN+6nxLJmJ/VX1bG8
9/y36Akbl67XzfhUd7Y3zMeP971XzAbOt8MHZjYQF8jn0YVUI8731ISZN9TX29Pd19rX4gMj5+GH6
OJIN+nn8PG9p/K3+dkMd5eoQRhFZ6DINgeBJ6l4K0bBh7BoI4YEx7eYHSVLZ774GESSJ8Y4FEP
Y982IESMhZ9V4DpJIV97oDTHA1+hYDDj8t4OItTg2t43on5dzt5flija1p5/YeCX6l6goZ5VDF7DYWN
SLx7IISxPUp8HIPnMfN8rIMnJlI9UYJ6HDx984HPjsN3J5KUgnN325CFdlR4h458aop5Fly4XvZ5qls

OU6B6QYmESE562YgOPP97coatMc58E4VSJlh8yIQHHGV9doK8jb92RjmgzWh3ApbzdUh3tpQ
7aZd4S5HZXiJ45o+VUvV5io1yR816LotnPKp62lItMaN7iYd1JIF8UIV9HId9DYOLjK11g6C8gF52R
51adEh3AZn/aK53m5b6XVh4PJQVUIJ45pFVR1N5I46tPFI6T4wXMXI7D4I9Jkl76IbWHKR8s4Q/i5
h036ejf2l1p6Otc3B2Zp+4Z+p3A5wGXXK13qZh2Uch4W5UPRux5FZHFPBZ5246PMVd6qYtVJk7k
lgGHLx8aYtZi2iJtWBnf2el7WLfc46IK2VdZ+uHgmfhXHyG12p3UU2GKm0jRh+Fi2/aOvSE+XKcL+
eEcnVsJJuD/HhKGsaDnntlipyHWGeJfrOGwWlicvWGLmtAZ22Fq20nXBeFKW8IUPuEp3E+Rd6E
LHNIOsGDu3WcL8SDVXfdJlImDBXoqGsmCx3ymic+FQm5KfgaE02+ScmeEZnDfZvmEBXI8W7m
DpnOsUK+DSnUzRaKC8HbMOpSCnnh3L6WCV3orJHqCKHvqGsuCCH3NiQWDg3TUfVODL3W
WccyC3HZcZniCkXcwW1CCS3gWUFqCC3kTRWGB0HocOmeBnXsxL4qBdnxTJG2BZX2KGS2B
YH7biDqB8Xt3fJmBuHuncSWBfXvaZeqBRnwgWtmBF3x6T/qA73zqRRiAzn1hOjWAtn3kL3CAqH
50JGqAr38hGuOAvX/gh3KAkYlxe8mAcoHWcFSAUIF7ZS+AK4EsWjqADoDxT3p/+oDPRLh/7IC2
Ofd/5oCqL1Z/6ICqJHI//IDEGyCAEoDphpp/XIkOev5/UogRb5d/RlcSZld/L4YoWap/l4VTTwZ/IYSaR
GF/JIPuOb5/L4NRLz9/P4LAJld/XoJIG1l/e4HYhbJ+U4/2eiJ+WY5Fbsl+WoyWY85+UYsQWQt+UY
mkToZ+XlhVRAJ+bocXOYF+iIXpLyR+p4TFJJB+1IO0G4l++4KrhMp9cpa4eTJ9hZRdbdd9kZIGYv
J9jo/qWEx9IY3qTe19qYwLQ5B9xoo9OTh974h9LwJ+HlbCJJF+WoUKG7F+j4Nhg9J8tJ1meEJ8zp
ptbO984Jd5Yh9845TCV5V875lpTVp9CY+zQyR9MY1QOPJ9Zor7LuJ9ooikJJF98IZDG9N+M4P/gt
d8FKP3d2N8M6BnbCx8SZzZYW58UJI+Vvt8YZZETN98gJMwQsh8sZAZOLh8841FLsd9O4pWJJ
F9I4dWG+995oSGgYyQUWCFdnGO+2L9a4ONqmV7YMiMcmgFVkcLO2qiS/KKB21ZQZ6I53Ab
N0eH3XLrLRKG3XXMIqWF73jEGbIFHHv4gNyOGmdLddmM72kuawSLx2sYYGGKsm0PVe2Jn28
eS7GIkXFIQWyHk3OANySGpnXILP6FxHgflqKE93qKGciEP30kgCyMKW23dUaLIW8Tao2KGNB1
YAGJIXHpVaGILXNwS3WHQXURQT+GYXbANwOFkHiBLOyEynpPlqCEGnwxGdeDfH43f4OKh3
PsdK6JIXTKagilpHWtX5OHvXagVUiG33emSy6GDHjCQQuFSXnnNuKEIXsYLNyD7XxZlp2DWH2
3GeOC0X8wftaJDXowdA+IM3qHaXmHV3riXxmGgXtQVOOftnvRSt2E+XxoQNCETn0CNr6Ds32I
LM6Dln5blqWCoX8zGgOCKoAffi6HwICBc1+G/YBaaMWGN0A0XniFcYAUVFqEtoAJSnGEDIAU
QIGDcYAmNo6C5oBBLMGCYoBrIsCB64C1GkqBeoEMfXmGmYb4crKF64Y3aCKFN4V2XeeEf4T
DU96D1IQnSg+DOYOkQDmCroMqNmOCMYK8LLWBuYJdltiBS4IZGoqA34HhfLKFml1+cfSE+o
wQZ26EVYqmXUWDqlliU1ODCYg2SaCCeocmP+iB/IYhNjKBjIUpLKWBIIQ8luqAvoNIGsGAWoKb
e+aEvZPgcSCEKpHSZpeDj4/KXIGC6Y33UqmCUIw/SRmBylqIP4aBVYkXNfWA84eTLI2AIYYVlv
SAQYSdGu9/64M7ewuEApovcEmDdpeMZciC4ZTvW8SCQJKJUgaBrJA/SJeBKo4WPyeAv4v6Nb
uAaYnmLHaAGYfUlv/1YW6GxZ/jYPFei2DZKBhb4SC3Z0sZRqCTJn5WyWBsJb4UX2BIJQXSCq
AopFZPteAQI6rNYp/9YwGLGJ/sYljlwF/eoazGzd/PoQ6d8WXEWCubZuVMmMXy6GTU2WMWdK
RgWgeUDGPs2rHRsSN7G2LPUMMQ3BbM8WKtnM6KmaJNnYtINKHx3k9GMKGeXyEdzeU/mcb
bSOTQ2j7Yz6Rh2rIWYGP02zvT/KOJm8RRpWMgnFQPSiK93ObM7GJhHX3KmCIHnhkIN2GzHrr
GN+Fln2cdqqlm04bK6RkW6ZYuGP8HAEWTeOVHGRT7eMwXMzRmiLPHTuPQiJzHa2M5+Ic3
iPKlyHJnp3IOeF7nx5GPmEz36bdiGRo3MgbDKQGXIYnOOi3T5WNYM/XYOT2+LfXc2RjCKDnh
0POKlunm3M4qHfHsFKliGSnxlIPCfK33nGRCEIH+BdZGQQHkLa62OyXl1YfuNTHnpWHeLzXqB
TxyKXXsrRe+JA3vqPLWHxHypM3KGnn1uKIWFgH5IIQKEcn9IGTiDdYBadQKPCH7+axmNon7z
YWaMNx7tV/CKxX8ATqmJZX8IRZaIGn9gPhaG63+dM1GF0n/gKIWEwIA1ISiDuYCrGYiCv4EtdG
2N74UPaoqMm4RzYN6LP4PdV3eJ3INITkKlioMCRUaHToK3PD6GK4JwMzOFH4lyKISEF4ICIUu
DFoHvGdCCH4Hpc8iM94ssaegLsonwYEWKY4i9VuyJC4e7TcuHw4bRRomGkoYAO/2Fe4U2Mw
+EeoR0Kk6DfYO/IWaChoMeGg6BloKMcxqMJJEraTOK549cX4mJn42XVkcITowPTTeHDYqfRHS
F4olMO6mE1lf+Mt6D4la2KkCC74V1IXiCBoQ5GkGBJIMYcl2LbZcXaHiKNZS/XtOI8pJwVZuHp5B

bTKiGao5iRAOFRlyHO1iEP4qyMq+DVojgKjGCcocRIYaBmlU8GmyAw4OQcZuK0ZzIZ8uJnpoDXj
qlX5coVQ+HGJSCTC+F4JH5Q6OEvo+SOxSDwl0yMoeC4YrYKiSCCIh/IZGBO4YbGpGAc4P1bg
Kd0WDwZNabQWNnW9eYuWXnUuyWX2h7SimUDGsoQZWRwm3zOPGPI3DNMESNh3O5J8CL
gna3Hw6JinnMF9+HvH0Lbaab3GcKZlaZa2kAW5OXAGr/UrSUwG0VSf6SiG9DQXiQXnGOOOG
OTnPgMEOMVHZXJ82KZ3jSHymlinthGAmG034ObUWaJGzbZDSXzG5aW1CveG/iUn+TTHGBS
daRLXM1QVuPH3UDONGNJ3bhMECLQ3jQJ9iJbXrLH0CHqXzXGC6GB376bOSYq3J7Y9eWZn
OIWvuUlnSdUjiSAnXLSZ6P83cMQTGN+HhjOLmMF3nAMDqKTnsoJ+KlkHydH1WG434tGE2FU3
/NbHaXVngWY3GVJHitWp2S8nlMUeeQ3noFSVyO3XrQQQCM9HuuOJqLJnyMMDCJcX1vJ+yHx
n5iH3KGJ390GH6EpICSbAiWLH2vYvyUDH3ZWISR6n4GUXuP5H5ASQCN8H6NQLuMFH7uOG
2KU39QMB6lqX+3J/iHBoArH6OFa4C3GNaD6IFMa5KVI4NeYoqTE4MAWbeRAYKIURiPB4JdSK
2NIllpQHyLUlIMOESJmoHxMA2H+YHcKAKGXyHTH9CEXyHdGSaDRIHwawuUO4kVYgWSOYg
iWTeQM4c1UKSOQoZsSEqMZIW4QDCKnYUcOBCI8ISEL/SHWYPyKAaFxiNpH/OEM4LsGWm
CulJ/anaTd46uYWeRe400WJSpe4vCUBCNkYp5R8uLullHP8yJ9oguN8ulUYcYL86GwoYGGJ/+FO
IT3IAyDs4PpGaKCKoL4adGSzpQ4YMSQ15I9V/WO2pBJT4CM9I58R0+LH4zIP2uJYYsuN4aHwo
mXL6eGPIgAJ/eEvIZrIB+DQ4TOGdGB34NfaSeSQZmkYCyQTJckV2+OUZSpTweMb5JURueKnZ
AZPxql4o38N02HSlviL4eFy4nJJ/CEVle0IDCC5oWUGfmBjIO2ZHKkr2EzXEChbmO5VDmeO2ZH
TDGbUWjeREyYcWuQPJCvNw5iNMKS53FILOyQSnRCJUCNuHdJHWyLL3pcFxSI4H2JZFGizm
b/XB6fqGkNVBecj2shTBSZum1DRDeW829+PISUPHHWNMSRnXRDLP2PFHbDJWCMmHILHZa
KKnvaF0mH8X56ZCGhJWyJW/GeFW4mU++bEW/KS/WYT3F8RCGVnnNDPHiTAHUKNMSQd3c
YLQyOAHkeJXuLmXspHbuJQ305F3eHH39SY+mfuHHmW7ecu3MXU7SZxnRPS8OXDnWXQ/m
UanbxPF6R3nhhNLuPaXnYLRWNCXtaJZOKuXziHdqleX54F5+GZ4AUy6KebHc2W3GbhHf6U3
GYoXjDS4eV83mcQ8mTW3qHPDyQ3nuDNKuOenyALRmMLH2AJaqJ7H6KHgKHu3+mF9iFtIDG
Y1mdSHx6Wx2adXzWUxaXpX00SzWVBH2WQ4SSeH4IPauQB36ONI+NrX8VLRWLZ3+eJcOJLI
AvHj6G+4DKGDiE8oFpYwScTIHKWsmZiYGnUsSWyIGGSuuUM4FtQ0aRsoFnO92PSof0NHWM
+YGFLRGKuoGZJdilg4GyHnSGUoHRGI2ESYH4YpmbdYceWl+YvYZwUI2WBYXISo2TeYU1Qve
Q/4S2O6KOn4RMNE+MVYPmLQOKHYODJeSH7IMjHp+FvoLDGNaDuYJzYh2av4xUWdqYDosr
UdOVW4oHShGS04j8Qo2QXogFO1COAYciNBeLvIZGLOeJi4VoJeWHYYSJHr6FPYOjGRSDQY
LbYY6aJ5F+WU2Xd4/bUUuUx448SZeSQoy2QiWP0ItEOv+NdYnqM9+LNliPLMqJCYczJeKG54X
WHteEzYRtGUeC24M0YPmZqpaKWMqW+pRmUNiUSJJFSTCRx5BAQc6PV45ROruM/ox8M6+K
wYqmLLGlnIjOJeCGglb6HuyEb4UZGXKChoN+Ws+rpWGrU3SoC2QTTEykbmaKRT2g0WkrPkqd
QmvpN3aZxm7IMlyWaXG/KZmTJ3TMIs6P9XfdG9qMzXruFISJ834FWumpz2cnU4OmTmkfTFKiy
GsmRT+fPW1bPk2bxW+pN3+YZHIVMKOVG3SYKb+R6ncwlwGOzHnFHBOLwHxUFpSI/n7iWvS
oLWxtU4OkwW36TEmhTW+XRTmd1XFhPkyacXNCN4WXKHU9MLST8ndNKeCQ0Xitly6Nx3uK
HEWK032bFsyIJn+pWuKmyXGHU2qjbXKuTC2gB3PmRR6cmHVNpjaZQXbHN3uWB3hWMLuS4
3ntKfmP13uMl1SM4n0oHHCKBX7DFvyHaoBZWscIh3aJU0eiPXdlTAee6HgdRpubg3kePhuYNn
ouN2uVCntPMLyR9HxwKgyO+H2RI3mMEX6zHKOJQn/YFzyGsoD4WqSkbHt2UxqhNnvXS8+d8
nxERMaamnZTPe+XWX1zN0+UNn4kMLKRKn7VKhiOM3+FI56LUIAzHOMlfYDdF6OF7IGEWnWj
e4BqUuigUoBSS5udG4BHRJeZz4BjPciWmlCRNzaTfoDRMKiQeIESKiGNh4FUI76KpoGSHSiH0
YHGF/+FPoH/WimisoVbUp2fkITCS1GcX4Q4RFOZHIpjPZCV7IOgNw2S2INvMJGP2INAKh+M7I
MRI9SKDoLdHVuHO4KbGE2EqoJnWcmiDlouUjOe7YkkSuObvogoQ/GYfodjPT2VUoavNs6SQYY
OMGePRIVsKg2MXITHI9yJhIQcHYGGuInfiGI+ELoK/WVGhhY7wUb+eY412SnSbMYwKQ46X9lr
RPOqUyomrNo6Ru4iYMDyOwYeAKfmL3oZjI9+JC4VBHZ+GR4QOGMaDxoMKWMyhGZOTUU+

d8ZGgSheau4+7QzyXgl4NPKSUV4xzNliRSYrtMBiOU4liKeeLc4fRI+KlpYZCHbiF6ISjGPSDcINJtF
NmMlu0pRtoDI6xlfpp7mGyhuZr4mSid+dt5GeMaQNV+WpvWh9yJm1ASzI0ZHAAPF52nXLCLT14
unV0Hyp6/3hSsnpiq2UQo3Vk3WcxllZnFWIVhZtpXGtvdsIrlqG2QaBZt/W+2WV5wYnHRSqFy03Pi
O/J1PnX1LQJ3jngGHyt5/no2sMhfl24IofFiFW9WkzBkIXCphG9nF3lQdcIpnN5Z0FsLHTjWLFuxnZ
GShpxZ3elO5N0AXkKLM52f3p5Hy15GHv8rypc6XbHolBfpXdHkexiYnfPg1ZIF3h+dNlnz3kxZnlqin
nqWBFtUHqaSZ9wHntGO0By5Hv8LKF1jnzKH54Tn2hrapal39nnyhdfn8lKl1gZ37qgk9jSn7fc/lmL
X7bZcBpEX7fV31sAH7bSS9u937UOvdx5X7aLIF0uH8HH0B3nX8xrE1YnofVneNbpYbXj5VesIXfg
U5huYUecyBkwYRIZRBny4O0VvFq3oL/SMht94JLOrhxCYGnLHZ0BIEwH3V3DYCtqvXW2JA8nLd
Z/Y6Djo5dJozPgGtgUYtRcmFjflncZHNmqIhxVnZp24cGSG1tFIWjOoJwR4RSLG1zZ4M2H6V2kYI
Hqa5VSJiNm4xYhZYXjYdbx5Omf4lfDZFtcaJiVl8+Y9hIno0YVfxo74r6SBNsRYjnOk1vmlbmLGNy3
IUeH852J4M9qGxT86Cwml5XP516jHFakJpLfppd6pdjcNphR5SCYzRkqZGrVXtoFI7dR7drh4wcO
hdu+YlqLFVyYYboH/F1zoRNpzFSy6iymT5WlaTPi3BZfaDwfb5c551ScCFgVpm7YpxjzpYsVQVnU
ZKqR2Nq3o82OeZublvPLEdx9YiRIA51goU9pghRzrCFmD5VLKwLipZYkKeMfQRcCKMrb4ZfiJ7S
Yh1jEpqBVKJmqpZFRxxqT5lbOb1t9Y4ALDxxm4oLICd1Q4YMqmZsD1vam/ltgl7DjaRu+mGxf11w
gmSdcTZyGGeHYzVzwWpvVS91fm1NRyN3TXAiOSt5GnL5KvV61nXEHe18tnjCqLZow2SZmnlqh
2a2jFRsT2jZfjpuHmsDcD9v9m00Ymlx229rVllzznGfRqF1y3PPONB3yHYCKsR5tHgwHe97v3qDpy
5l3W0KmRtn5G5jiyNp7W/EfTRr9HFCb2JuBHLEYbJwH3RKU/VyQ3XSRI50bXddOH92InjsKpl4sH
p8HfF643wnpbTjTnVYI9JlJnXxigRnz3aTfDxqBXdibpBsQng3YQVuiHkTU2tw2HnvRcZzMHRLOD1h
HuxKnR3yHynHfN6IH2rpFxfH2MlpljX1uiPJl531Ye09oSH12bchqrX2cYGBtG33MUuhvk336RW
RyE34qN/t0jn5lKlt2+X6/HgZ5c38eoxVfKIWbIWZhtYTKh9dkQoQCelpmxYNybPlpToLrX7dr3oJuU
mVueYHyRQZxGoF8N8Vzt4ESKIZ2RIDMHj143YCCoddddo2mIElgH4wihtxiylqleYBlaoIhbEBoE4g
nXyJqXlb3UfBtfoXORLJwPYSuN5dy+YOeKIN1p4K3HnB4W4HGoJxb+5Whky1euJNlhd9hd5Ewe
KVkMo85a4hm9Y1NXotpw4ttUXxsmYmWRGBvdYfON2lyToYUKk51G4SGHpt364Lon21atp1ykgx
dgJp9hNFgTJeRd7tjGZTxasJl8JJBXeto1Y/QUQBxrl1SRAluvlrhNzhxs4h8KkR0oIY4Hr93joPlnkJZ
mqUlkPdcbaGHg9ZfQp3xduJiH5qiagtlB5ddXVZoAJQiUI1rBpD3Q7luGl3bNwxxKorGKjp0NYfKHt5
3PoTEnSZYpqyqkABbgKh5gwJeXqRGdithR6A8aXfKp5w8XNlnSZhlUC1qY5RrQ3ZtjZCgNudwto
zdKjJz24kuHvd2/IWDolhyAFvskuFzCi7DhVh0GmGid+Z1PmSlapt2aWd0XX93nWplUFI45G1SQy
x6O3A8Nhd7k3MrKmp84nYRHMZ+UXkunxVu4WQhkZlWNGY8hDtxjmhedvJy82qRac10XGzPXN
V1ym8ZT9B3Q3FmQsF4xnO3Nc16SnYlKKN7ynhXHMI9Y3rOncNsGGwSkG5tq214gzlvQm7mdh
Vw3nBzaRRyfXILXD10H3OuT1V1x3VbQmJ3c3cRNYx5lnjKKlIF6znp+HmT8kHxRnHhpo3Pij01ra3
SagkBTNXVadUFu/nZDaGJwyHc4W6tyk3g6TuJ0ZnlBQgx2QHpONVJ4GntjKGN573yCHM171X21
m0Jnf3uwjjdpbnu3gUxrX3vHdG9tT3wJZ7FvP3xZWxpxMHy2TnBzKn0XQbl1K31+NSB3LH3vKFF
5JX54HOF7K38LmhplqoNpjRxnuYLGgEJpyolsc4dr2oHFZu1t6oFuWntv/YEnTfZyF4DIQWR0OICs
NPN2WIB+KFN4clBrHRp6joBZmPNkC4sijBFmNlnSf1RoX4ilcrdq4dzZj5suoZvWe1u64V9TYlxJIS
WQRlZYYO7NMx1nYLrKFZ30YI8HU96B4GJl8lin5LNiwNk3JDKfmNnG47NceRpX40MZYtrpotcW
Vxt8YnCTRtwRYg1QM1ynoa3NKV09YVFKFV3RYPyHXx5koKYlqhhZppQie5jsJeafV1l/ZTtcP9oU
ZKHZMhqqpA1WL9tCo34TKNvdovKQHtx6omrNHp0XYeTKFB2yYWOHaJ5MYOEIYxgWKG3iOVi
qp5dfGlk/5sLcChnYJf9ZBJpyZUCWCxsPJldTDRuvY9KQDBxS4yGNFNz14nEKEp2XocKHcJ43o
RSIlnfc6jzh/lhyqUGe5tkJaEZb3NmkZ1YY3hpCJmqV7FripYVS9duH5KYP/Fww48sNDJzZovBKE
R2BYhaHd14mYUElpR4J1vzibZ4v16/fPI5YGGTcFI6HWR0Y+h62mdgV7B7mGpZS3d8YW1VPz9
9NHBTmx5+C3NYJrh+43ZaG7J/1XmWIWV1ImO+iKp2AWXZfBF26Gf8b5R34WoyY0V42Wx5Vyt
50G7TSw16ynE1Pux7yHOgMuR8y3YQJpt903h8G7V+8HsVIERydWszh69zj2ypezt0r24obuJ122/

EYrJ3B3FuVrR4MnMqSq55WXTzPqN6f3bJMrJ7qnikJoF833p9G7d+JHx4kzRwIXKAhr1xanNYem
pyuHQ4bjJ0DXU7YiB1YXZNVjx2tHdxSIF4BXiePI95VXnUMoZ6qnsTJmt8BnxdG7l9cH2+kiluD3na
hcxvgXoMeZVw9npGbXxybnqsYYhz5nsjVcJ1XnusSfJ21Hw8Pht4SXzVMI95wn14JmB7QX4xG85
8yH75kR9sTYE4hMxt3YDKKeKJvb4BibKhxBYAnYNVym3/+VTF0M3/qSYR1yH/fPdF3W3/fMjx48H/
pJmh6ioALHAp8JoAykBNqwligg9hsZ4eld8huEoZ1a+lvwoWTYDRxdITEVLBzKYQOSSJ02oNIPY
92ilLKMh14N4l6JnF56YHEHEF7mYFPjwNpZZAAgt5rHo42duVs3lxyayFuoYrqX4pwaol4VCdyN4
ghSLx0AYbaPUt1yYWIMfx3koR4JnV5XINiHhB7lIJLjftoPJc4gdhqA5TBdeVrzpJSaj5toZAqXshveo
4bU4xxWYwpSEhzOYpJPP91G4h5Mdd2/latJnR44ITpHJh6uoMnjOxnP55VgNdpDZtAdPRq4Jgxa
WdsvpVmXhFupJK1UvlwkpAiR9tyho2kPLd0gls2MbR2eYjGJnF4dYZQHLL6ZIPni+NmaqVJf+1oP
KGkdClqE53/aLJr+pqGXXZt7JcqUn5v6JPuR39x7pDNPHtz/l2+Mzd2CYqsJm94G4eNHNv6HISLj
MF+X1v6gKx+ll6yDLt+02F0aO5/IGRUXVh/ZWdEUgZ/o2pJRrt/5W1WO3aAK3BrMEWAenOIJMK
A1HanGq+BRXoAi9B7gGNkf9h782V2dAd8bGeRaF1872nNXOF9bGwfUaZ9426MRnB+VXEGOzt
+xXOMMB1/QHYJK9/ynilGrSAZntgiuB48mptfwx5l2vjc116Qm1kZ9F69G8SXHR7pHDSUU98Tn
KpRit87HSQOwd9hHalL/p+JniFJJ5+3HqEGrh/oHymie92xXE7fj13kHljcrJ4YHMOVZ0N5NXQ0XAF6
CnVmUPJ63XauReR7o3gCOtR8YnljL9t9K3rLJI9+CnxBGrt+8n3PiQl0y3gnfW91u3hycf52r3jHZqt3
pXITW4J4nHn0UI15knqrRZd6e3trOqB7XXw2L798SH0LJlt9R331GtJ+TH7viCVzFn8jfl50JH7ccSN
1Nn6dZet2R36PWuF3Wn6WUAx4bn64RTd5dX7kOmF6c38dL6V7eH9fJJt8jn+1GxF9pYAUhzpxl
YYye7Vyu4VJcF5z5IRnZT51DoO5Wk92O4Mjt5h3a4KqROF4jIJA0il5pIHIL456wYGUJkP77IFW
G0t9EoEehkZwSo1Bes9xgourb4lyvlodZIFz+YjPWa51O4eeTxh2glalRIJ3uYWMoe146YSeL3R6
HYO3JLN7XYLeG3x8YlIhVRvNJQteddweJHwbpBxvo+8Y6JzB43WWO50V4wNTOb1q4pnRBJ29
4jVOaZ4PodUL1V5iYXTJLd64IRPG6Z8LILUhFVuSZsCeOBvk5gsbaNw35VdYsxyMpLYWDdzjJB
zTe507Y40Q6h2S4wMOWN3p4nyLzd5B4fTJLh6dlWiG8h704OFg1Vtg6G0d/1u0J5ObNtwH5rrYh
hxepe+V5xy3pS0TXN0SpHVQ051ul8QOST3J4xcLx54momiJLh6GobNG+V7ilQcg0CEolwdeAKE
Q17rbO2D7WHBYf+DvmShV0eDi2eVTMyDVGqeQIGDJ22xN9aDBXDOLXWC63P4ls6C2ncvGa
GC4nqhgmeB8mMnd0aByGVabE+BqWeTYX6Bqmnbt+Bqmw5THuBp26yQhOBp3E4N6iBq3PL
LVuBuXZpIsiB03kSGbSCAXvqgZF/j2nTdpF/IGt3a7x/oW0hYQh/zG7iVoF/+XC1TDGAJXKfQdqAT
XSZN4CAAdXaiLUOAp3i1IsSA6XrVGcSBOX0XgMF9hnBDdd19rXFkayN923KKYIp+JXPfVhx+c3
USS+B+w3Z1QZ5/EXfhN1h/XnlYLS9/tHrcIr+AGnx2GdGAin4pf/Z7rXbGdSZ79ndZaoJ8RXfyYAJ8
q3ijVax9FnImS4h9h3o/QV199HseNy5+YnwFLR1+13z6lsV/WX4NGfJ/4n8xfzV6EX1UdGF6dn1lab
564X13X1Z7YH2WVRx75X3JSxV8cX4UQQI8+H5INvt9f37BLQ1+Cn8plt9+oH+rGjp/NoA3fml4plQ
Fc6J5IYN5aQ15o4LuXrt6OYJ1VJl61oITSq57eoHLQL58GYGONs18toFdLQB9VoE4lvh9/YEqGnx
+oEjfZF3ZYrActV39YmHaE54iohSXhB5M4dDVah54oZNSjx6moVzQGx7ToSoNpt8AIPqLO98tIM
1lwp9bYKPGGrV+llHzfLt2U5FVcfd28498Z213l42nXUZ4S4wGU1t5B4qASbN5zIkZQAI6kofCNI97W
IZ2LNh8lYUsIxV88lPhGuR9tIKme9d1a5fWcRh2FJVnZpd2wZL8XIR3fpDDUrV4RI6oSTB5FIlywP6
t56YrHNIz6xIjpLMJ7olcHlx18g4UWGwx9WYNAeu90qp44cEp1WJs3ZeJ2CZg2W+F2zZVeUil3nJ
KnSMJ4d5AXP1t5Wo2aNfZ6RosoLLB7M4iylyN8KIYjGy19DYPEedGLCFxJb3aKH18YZUeJQGH
wWz6lhmTXUWSHzGfSR8OHEmrkPhaGbG4BNGOF1nEpKtCFSXRjIP2ExXezGKyEXHs7eRSIg
WL0btWHyWU0ZMKHGWd7WtGGjGnTUQ6GAGxBR4CFd27LPeaE+XFGNESEhnQDKsOEHXa0
IQWDv3l6GMuDD3xseF6GSGILbjuFuGsFZEWFmwzHWmyEx26eUL+EY3CIR0SEBHKLPbqDq3
SaNCeDWna3KreDEnjilQyC13siGOaCrX2Dd62EX29lbaCD73CkY8CDhXHqWgCDNnNEUGiC7n
SxRwCCsHYzPYqCeXe5NAuCTHIKKq2CJXrqIRKCCnyoGP2B/H5/dvuCp3WHbQCCVHZFYzOC
BncJWYmBz3fiUAiBonjNRrSBf3nMPVSBZ3rLM+uBV3vRKqWBTHzoISKBSn4jGSeBUX9udlGBJ

nuwbFKA63v4YoaAtHxBWPGAIHyQT4iAf3zyRICAdX1qPQ2Ac33kM8SAeX5kKqCAgn7zIUaAj3+
gGXiAoYBYdaJ/y4IBa6x/p4GzYet/h4FnWGh/fIEntXR/fYD8Rfd/iYDpPM1/nIDbM6B/tYDWKpt/zoD
eIWWh/6oD/GcKABoEpdOp+lohiavp+ildtYUF+e4Z8V9B+goWuTpR+k4T3RZJ+sYRaPIId+14PFM3h/
AoM6KpN/LYK3IYJ/WYJGGgB/g4HfdDB9jY6eajZ9jo0RYHI9klukVxt9oYowTfl9vojwRRI96YfNPDF
+HoaxM0Z+XIWdKoV+moSLIZV+2oN5GjV/FIJ7c2d8rJTHaXB8tZKrX7p8wJCVVm582o6rTWV9AI
zdRKV9NYsvO999eYmJMxd9ylfoKnd+GYZDIaN+bISSGmF+t4MCcpl78ZrRaLI7/5gnXxl8DZWAV
d18L5MATOh8XpCeRER8m45gO5p87lwtMu99S4n/Kmt9rlfNIa9+EIWGGoz+aoN0cHGRklx6Zwe
QII8+XcqOt2IPVKeNZ2UAS66MGmgGQuWK1GslOgeJqm5OMRuImXGFKFKHkXTOH06Gkng0F
82Fs3vMb9ePMWLMZoSN7GULXV+Mq2dWVE+Lg2nBS2mKYWxDQrKJR27hOeWlQnGIMQqH
UXQ9KFKGaXcCH2OFjHniF/iEynzn0eNHGjVZgqL+2qVXPqK3WxiU/6J1G5TSyqI1XBZQoOH4X
J1OcaG/nScMPqGK3bQKFOFYXkTH3aEpHtwGB2D/X3nbrOLUW6fZYeKS2/rXImJSHFDU6OIVX
LASuKHb3RPQkyGmHXyOaGF1XeWMOiFI3IBKFOEeXr+H4aD2HzcGD6DSX7ObhmJtnRpZP2Iy
XU+XBGH3HYdUz6G/3chSpCGL3g2Qg6FcnldOXeEynp/MNSENHulKFWDo3zdH5+DGH47GHC
CnH+nbYCIvXo0ZGWHd3qdW3yGmnsLUrqF0nuNSiKFGXwgQbmEcnzGOT6D3n1pMLeDwn4P
KFqC2X7FH82CWn+aGMmB54B2bO2HCoAlY9iGQ4AAWveFfH/iUkSEyH/dSb+ElN/rQW2Dj4AO
OQqDDoAxMJ6CmoBZKF+CJ4CNH/iBs4DaGRqBSIEtbFaF4IYIY0GFLovkWmSEfISsUcCD14Qg
SU+DQoOqQReCv4NLOM+CToLtMICB6oKUKF+BhoJEIBqBIYIDGV6AwYHNa7aE44wFYpeEPY
q4WbWDI4I0USGC/YhkSMecC4drQK6B+4aMOlaBmIWtMFmBRYTQKFmA84P2IDOAoIMbGZiA
UIJVawiEC5HTYeuDbJACWQ2CzI44UliCO4yfSEWBuosgQEmBS4m7OECA9YhYMDOAs4b0KF
KAcYWNIEiAMIQZGch/8YLJalaDVpeAYUuCu5UoWH+CHpLWUAiBlpCvR9eBHI6kP/SAtYy4OA
WAa4rPMBOAN4jIKeYAA4b3IFI/0oT1GfB/oYMrZxaYFly5XqWV+l+RVI+T62JzThuSFGVmrFuQR
mhxPgSOg2uXNfWM3G7KLdeLTXINJd6Jx3VgHbCIRnjKFwCG8HxYZq2V2mK8XkqT42UWVhG
R+Wd5TduQRmnyRcmOnWyBPeGNAW8rNeOLenHhLdaKB3SnJe6Innd5HdOHPXpeFzaGAn1c
ZkST42h6Xe6SC2peVcWQP2xLTZ2OqG5QRZqNH3BpPcGLpXKYNdGKPXTTLdSI5HcbJf2Hlnlu
HfKGU3vTF2WFMX5HZdWSLm39XYiQcG9zVWmOunDxTVKNNnKKRWCLw3QzPZaKY3XuNb
mJGHerLc+H3nluJgmGrns+HgyFhn0nF46Ee38aZWQCp3N6XRuO/3R/VQaNYHWNTp+L7na0R
R2Kj3fqPWSJRnkvNZule3pvLceG83uwJhaF2Xz/Hi+ExX5rF8iDyn/cZOaPVHjwXKCNv3mQVlyM
MnozTJKK1XreRMKJinuZPSGIVHxkNXCHMX0rLbeGIH3yJieFEn7EHmiEBX+qGCmDEICSZG+O
H36EXC2Mn36fVB6LJ36/TDCJ3H7qRHClo38nPOSHfn92NUmGa3/DLamFZoARJjaEYoBnHpuD
XYDLGICCbYeyY++NDYQjW62LoIoU6KKOYM9S8CI/YLpRBKH0oKoPjyGuoJ8NRqFtYJPLZW
EvlliJj6DxIH6HsSCyoHWGMqB44G8Y2KMI4mkWxeKwoisUwiJZoe8SzWINlbqQ5yHEoYuPEGG
BIWGNnyFCYTdLXaEHoQyJj+DNIOFHuSCSILQGQIBb4IxYsiLXY8WWn+KA42iUnOlYwzSq+H
gorhQymGZ4mkO+mFYIh/NKCEclDWLveDkYYoJj2CtIT2Hv2B2YOyGTyBDoKVYiyKvJRpWfKJY
pJyUfalDpCASj6G6o6pQsmF1ozpO5+E1YtDNG6D7ombLT6DGoftJjuCSYY8HxOBe4R0GWiAvY
LpXeqet1zyVm+b7F/hTxyZNWLWR7iW0GXPQHGUeGjgOUuSMWwPMguQBG9NKruN73Kel5C
L5HX5HDKJ3HlgFkiLDXzbXbecImKrVj6Z7mUiTu+XWWefR5GVEmooQFSS22zGOTmQt299Mgm
OpnJEKsuMqHUbl7GKtnf4HGOIznrcFomHGn3JXXiasmgjVgWYJ2osTryVrmw8R2mTgW5WQDa
RaHCDOSiPY3LGMgaNbHUWktilhXdzl86Jq3nTHI2H4Xw4FsGGRX6gXTKZDW1IVb2WmW8FT
naUNHCsRy6SGXJhQAiQE3QkOQmOJXX5MfmMSXfPKuCKfnmpl+alwXuJHLKHEH11FvKFJH9f
XOOXj3KZVW+VNHPQTiqS53UNRu2Q3XZXP9SO6XeuOOWND3kSMeiLSXpwKuSJIXvLI/+H7H
0uHN+GTX6gFzOE2IAOXI+WRHfBVRWUAXiXTcyRy3lvRpiP1HpDP46N8nsILOMKHwUMc2Kc
Hz+KuKlx33IJBuHJ37PHSGFjH/BF5qEGYCsXDaVJHz8VL2S9X1WTXiQ0X2yRkyO7H4NP0+NG

n54OlalX37xMbOJsn9nKt+IE3/cJDOGeoBQHVyE4oDEF/eDcoE2W82UK4I9VFSSC4IPTRKP9Y
HmRfGOHYHKPwOMWIG9OE2KqIHBMZCJBhHDKtSHb4HCJESF34G+HY2EToGyGEaC5IGtW
1OTV4diU9ORQoa9TI2PNYYcRXmNZoWJPP6LqIUHOACJ/4SWMV2IZYQhKr2G2IOIJEqFUYMj
HbKDzYKQGlicBoIRWsuSprox6U0yQlotfTAuOjopHRQWMxYk7PjuLDYg/N7SJaYdXMSqH1oZnK
qWGUYVvJEyE1YRvHdCDXYNXGMCCCoJnWkCSGZF0UtGQCI/bS52OAI5ERKOMPYY7PeiKiY
tFN3Sl6onjMQCHXYh6KpGF4lcJJE6EbIWRHemDAIQBGO6BulKuVJ+lgF1WTgCiUGAxR5CfJW
MYQSKcDmYfOseZDGk/NIKWlmx8Lh+TV2/MJ6eQpXMvIVGOAnaTGSsLZ3n3FZuJGX1aVKGjZ
mLATfygXmUsR4idV2elQRiaW2pGOsGXdwZ6NISUrW/GLi+R+XKjJ8mPWnWPIYKMz3h4GwKK
VHtaFealIX40VJmhmf3Teyenmn7R3ObtWwNQQiY025FOreWCXCNNIOTXXLqLjuQwHVVJ+aO
NHfKla6LwHo5GziJYnyeFieHSH72VHef6Wz3TcSdF26YR0aaQnBIQN+XcXlgOpeUunQGNHGS
nX6LjyPnXfvJ/yNLHnklDOK0nvVG2eljn3DFI+Gi3+jVEOec3HbTZGbu3MeRxSY/XRuQLKWPXXjO
nOTI3diNFuREXjsLjiOn3psKA2MQnmvlfjJ+31dG56HyH7UFqaF1IA9VBedNnazTVqalXeaRtaX7ni
KQHmVQXmPOkSSq3qfND2QM3u6Li2NzHzMKBiLeH3Xlh+JNn7bG+qHBH/XFxWFEIDFU+WcK
nuYTSSZmHwIRp6XAHyCQEaUZH0WOhuR3n22NCKPcn5gLiONFX8GKCKKyH+klkKlioA5HC2
GWIC9F3iEZYE6U5+bSoB/TNqYwYBvRIGWMYBrQACTo4COOeGRJ4C+M/mOxYD6LgyMcIExK
CGKKYFgllqH8YGDHGWFW4GPF8uD1IGfU0WajoVLTHOYC4TLReSVgoRXP5+S+oQLOY+QhY
PLM7qOKIOYLeSL2INdKBKJI4MWImaHZoLCHI+FQYJRGBGDW4HzUtGZ9loGTAGXcokYRXaU
6og1Pz2SZ4d3OT2P9obGM3uNnYYhLbuLUoVxKAGJFoSzIm2G7IPoHLGE0YL+GEyC9YI7Uk+Z
eo6hS5OW9I07RRqUaovhPuyR64qvOPmPfYmLM0eNKIh1LZiK4YdTJ/KIqYYilnOGhoTpHM6Ec4
OSGH2CoYJ3qZdiBfblmzJkVlpMjOVmpF22fqdo1WEBclRrGWRHYoNteGeHVIZv8Gq4RoxyfW3
YOKV1C3D5KoB3i3QJHYh6MXdDp61eA2AcMypgwmKpi35jemU3fXFmB2e0b39onmo3Ya5rRmz
AU9xuAG8+RglwyXG0OE1zkHQtKlt2R3ajHaF5G3k7pfJag2jomAtdm2qqijlgqWxwfFpjfG48bpVm
WXAKYO9pRHHaU0RsO3OIRZVvPHVvOAFyOXdAKjt1I3kcHbZ4JHsSpFIxb3F8lqJa2HJziQFeM
3Nxe1BhQ3SKbbdkWXWoYDpnd3bLurZqonfpRStt1nkHN71xBXovKh90IHtzHcl3S3zHouJUvXnil
VRYYNogh95b+npjelhfQHRhHbOhiiXszX5NI2HunUjNpMnwWRMxslXyHN4Rv830IKg9zPX2xHet2kn
5joZJSaYITIB5WQYGahsZaC4EmeWhdf4DabCBg9YCWxvNkclBbUbhn9IAhRHdrgX/sN1hvCX/J
KhVygn/WHi52AX/joFFQUlo4kwFUVokKhc1YTYfeeJNb64bba29fioXhXmVjLoTyUU1m2YQKRCx
qjIMtNzNuPIJmKhtx4IHHWmt1hoFBnxbOc5JFkelSnpBehNdWuY55d8FaeozBasFePIstXdpiBlxU
OJI04fcQ+NpqoZYNw5tgYTqKh9xUIO3Hp91HYJ5nexM15ofkNFRGpd9g9dVTJThdudZJ5J+agtdB
pAIXUdg7I3YUHRk3YubQ5po2llXnups1ldBKh9wz4V5Hsp0xYOLnMhLa6HUj8dPwJ6HgupUBps9
dh1X+JgsaWRb75UkXMBf75lnUA5j/o8/Q1ZoG4xsNsIsO4mrKh5wX4cXHu90eoR8m7hKLKIWjttOI
KVugiJS7KGEdXRW8p24aNda/pn4XE9fF5ZET7IjQJKsQx5neo8vNq1ruYvFKh1wAliHHw10PIVM
oCBnulc2koJpqlqDhP9rlF3Vd4ptXmEYajdvOmRcXQ9xLmefT+NzPGraQrd1YW4MNad3iXFCKGV
5rHRtHGI78XflnmhkAV/RkQNmVGJXg7hon2TidnRquGdraVBs32n6XFNvGWyQT09xZG8jQkhzv3
G2NV92HXRNKEh4dXbgHIF66HmanNpgsGghj6djWGnpgpBI9Gu3dXpoT22WalFqt295W6ttMHF
hTstvtXNNQeVyRHU+NR901Hc2KC93XnkyHJV5/HtNm2ddwHBHjmVgsXFTgX1jIXJmdJFmKnOa
Z79oyHTVWw5rc3YYTIBuK3ddQYxw7HimNOhzrnn5KBI2ZnthHKd5LXzemhNbK3hljTdeVXiJgHhh
cHkGc7FkOHmOZwVnB3ofWnhp4nq6Td1syHtXQTpvuHv5NLlyp3ypKBB1i317HMI4eH5amNpY6
4AcjBNcRn/Lf21fkX+Acs1ihH9gZkdlnf9JWeFog389TWtrkn82QO1uq384NJVxwn9LKBt00H+EHQ
534X/CI61W54fqiwVabYbpfn9d4oXucgJg/IUfZZ9kHYRaWVpnSYOiTQZqf4L1QKltvYJWNHJW+Y
HJKCd0LYFpHUx3X4EIlloZVH4+iifxYx43vfZRcXYxBcThfmlrEZPdi2olTWNVmkIfxTKJpgYaeQGZ
s44VdNFhwQ4QvKC9znYMvHYF28YIrlW5TmJcsiPJXVJTffJ5bAJJlcGReUpBFZEZhr44wWEIdG

owqTDhokoo2QCFsF4hWNDhvnlaEKDJzHYTZHa52IIMplFhSPp6Th/RWC5uFe7pZyJh+b6FdL5
W2Y6NgopL6V8VkJpBMS9dnvl20P+JrYYsxBpVB4i6KDRyroZgHdR2RYQIk05RDqXGhxBU7alH
evpYu55+bvxcNJsDYxlfupeXV1djVZQ7S4ZnBJD7P61qx43UNAJuioq6KDVyUle6HfN2BITHlrthld
nid9vEVqifSJwl13jcH5yAGEgZABzdGRiV7F09merS1p2kmrZPwF4Rm45Msh6AHGGJmV7u3TMG
199k3hHIT5p/l+BiJNr5GIDfAhtxGSKb45veWcYYzlxNGmxVxBy+WxXSt10z28CPqV2tHGyMpB4o
HRoJIF6j3cZG3V8IXn2k+Nm1GdWh2ZpCWkoewlrNmsAbrdtK2zrYodvJW7hVoFxFJ3DjSm5zM3Lw
PIR1SHUJMI53Y3cnJj95g3lGG4h7s3uFkp5kAW8lhkxmfnAtehp07nFZbe1rGHKkYeFtRXP7VftveH
VgSgdxtnbMPgtz/XhCMjN2SHnBJjB4IXtPG5l67Xz1kXBhh3anhUBkOXcieTBm3XelbSdpN3hOYT
1rk3kEVXht83nJSaRwXnqUPcZy1HtoMg51S3xJJix3wX1GG7t6PH5SkFZfXn4mhDRiPX3+eDdID
n3cbFBnKH3hYlpqFH32VOpsnn4cSTpvMX5KPYJxzX6EMfJ0an7NJj13BX80HAF5n3+jj0Fdb4Wig
zpgdoTRd1ljboQEa5FmFINhX+povoLPVGxrbIJQSNxuJYHfPUVw5YF/MdpzpYEtJk52YoD+HEF5
F4DTjitbu40Lgj9e4YuNdnth94oSatBkvjljX0pnhleRU+xqVIZwSH5tLoVhPQlwEYRIMcFy84N3Jlp10
YKsHHh4o4HijR9aRZRlgUFdf5lcdY5gqY/3agJjg44PXp1mZlw6U2RpTop9SBpsR4jUPMlvTlc+Ma
VyUIWyJmF1UYQ+HKZ4QoLNjBdY/5tmgE5cRpiYdLff5XQaUNibJNGXf1IYJDQUuVoYo50R71rd
4wuPI5unln7MYtxwlfOJmV044WvHMx38lObix5X46JVf3VbOJ7vc/hefpuMaKJhe5hXXXZkpgU5Un
pnmJI3R29qxo9QPFxUB4yAMXZxR4m2Jml0hob1HO13q4RLjUJzg1eLgSd0n1q6dTb1ul3vaVh2w
GEIXa13ymRmUjh422e1RsV5/mslO1R7NG5iMAN8cnHEJH59unUoGmZ/GnjCjAZwG189gBVxkG
G/dEdZAGRGaJN0TWbVXQx1nGI0Ubh27mwnRmN4SW7lOw55rnGuL9p7HHR/JHJ8mXdQGnt+
JnpOiuBtEmadfxdu0mh8c3Fwi2pgZ+JyFGxYXHtzn25eUUZ1LXB2Rgt2vHKeOtB4TnTXL7d56ncV
JGd7l3IXGo19T3u7idJqZW3RfipsZm8ScqduXXBaZzlwFnG9W/Fx0XMuUNdzkHSyRbh1TnZCOPh
3EHffL5d42nmGJF56sns7Gp18kX0KiM5oAHUCfUJqOHWkcdxsY3ZLZo1uSXcVW2RwMHfvUGd
yGnjcRWV0BHnUOmF18XrYL31343voJGB54n0RGr57435Kh9FI73wrfFBouXwxclPqXw8ZclssH
xpWsBuvXypT+dwznz+RQdy331hOiZ08H3RL2l3Bn5OJHd5Jn7kGwZ7QH+FhtdkFoNWe21mnIK
4cC5pFIldZRlrQYGoWi5tcYFJT3NvpoECRLNx2YDMOfJ0DYCpL1h2RICSJlx4glCWG0h6soChhd
pid4p1eoVIGYkqb11nrYflZGNp9YbQWZVsQoXTTvxuloTyRFxw6oQIOb1zP4NuL0V1IYLBjJ1374l
sG4B6OoGchOJhE5FreZNjyl93bnZmbo2LY5hoyoveWOtrLYpKTnZtmIjUQ/xwCod0OYFyglYpLy1
09oTjJKd3clOpG6951IJ2g+df4ZhDeKhioZWYbZ1IVJMnYtpnv5DeWEtqNI6vTfhstlyhQ6FvQYqrOU
lx1ojlLxd0aobnJK53AYUFG9d5f4M0gvNe3Z7sd9NhpZvHbOZkYJilYjpm2ZW3V8RpXZLnTY9r8ZA
6Q1VuIY2rORpxRosxLwRz9li4JLR2pIY4G/h5OIPWg+95i1eseJJ6S1rGbV17BI3qYIN7n2EdV318N
mRhTOV8y2e5QIV9bWsbN8t+HG6KLV9+13IElrJ/p3WFGX6Ai3k9guh2UF7+d7N3XGF2bKV4YW
P1Ybp5PGaJVwJ6FGkyTIV662v1Qg57w27JN5t8oHGSLUZ9i3SYlq5+jneKGZN/oHqngfRzbWX1d
uN0u2fSa/l2AWm5YS93FGvBVpN4Jm3bTC95N3ALQc56QnJQN3B7SnSpLS98YXcllqt9IHluGaV
+0Hv1gQlw6Wyudhpybm3+a1Jz6m9YYKZ1KXDUViV2aXJjS9h3qnQIQY544nW9N0d6FneDLRx7
WXIRlqh8t3svGbV+GX0lgCZupHN2dVJwW3QvaqZyBXTxYBVzbXXeVa501nbfS3t2Qnf3QUt3o3k
dNx55AHpRLQx6aXuQlrB763zkGdd9bX5Kf0tsqnpCdHxuh3psadtwVnqdX2Rx4Xr0VRtzb3tiSwh0/
3vpQPh2hXyBNu14BX0oLQB5jn3ZIs17LX6fGiJ8xX9ufm5q44Eac7Js4oClaSnu1IA3XsRwf3/0VJZ
yLX/KSqBz33+9QK91hn/ENsJ3J3/dLPZ4zoABluh6h4A4GmZ8M4B2fYtpVYfsct1rb4bPaGBtflW6X
hlvQITeVAZxCYQeSjFy2lN+QF90nYL1NpR2XYKBLOl4IYIVlv159IG2Gp97toFefKhoBl6YcflqMlzY
Z39sTosjXVNu14m0U19v/4hjSa5x44c2QANzxIYgNI11o4UfLNZ3hoQelwp5dIMdGtB7TYlne8Bm5Z
UscRhpG5LTZqprQ5CEXJZtJY5+UsBvEYyYSTJxB4rZP6tzAok0Nih0/4ehLMN2/YKIXR5BoRmG
vl69ILWetxl9JuacE1oL5isZfZqXZXFw/dsSpMgUjpuRJCgSMpwS45KP2FyXlwSNf10c4nuLLR2iof
Elx14qYWFGxt6q4Nretp/pVfgcFV/y1sPZft/8l5EW8mAFGF5UceAN2S/R/+AXmgaPjaAl2uBNG6A5

G74KsiBPHJ9IOmBp3YXGlqCJ3nmefZ8k17Sb5J9A2FpZVI9cmQFW0R90GakUV1+MGIZR65+k
mwnPfx+/m8GNEp/dXH1Krt/+XTyIPKAk3f/GKuBPHs2eRx52mVsbtp6imd2ZMJ7NWmFWsp7xmuf
UP58Wm3MR2V88nAPPch9jHJINct+LHTOKq9+2XdCIPp/nXnGGMiAa3xqeFB3gmvKbil4Y21SZ
C55Pm7fWIB59nB3UJx6s3lgRx17dnPfPZN8O3WqNAX9BHeEKqV92XlrlQF+xxTrGOJ/tX2Bd4h1Y
nltbXd2cHMTY5N3d3QxWc54U3VCUDN5NnZIRsh6H3eePVp7CnjfM+x7+HosKp588XuIlRJ9/X0C
GQ1/CX6LdstzgniVbLt0tnkTYtx14nmRWS9233oUT65343qrRmJ47ntaPRN5+XwUM8V7B3zaKpt8
G32tITl9QH6aGWB+XX+Pdghx1H8SbAZzK373Yjd0d37cWJ91kn7NTzd2tH7URgZ33n72PNN5B3
8mM6J6MX9IKp17X3+wIVx8mYARGax9x4B5dT1wWoWta0dxzYTSYYdzNIQTWAR0Z4NzTrZ1o
YLrRaJ25YKBPI14KYIoM3x5boHfKpN6toGfIXh8CIFvGex9SIFGdHVvFovvantwnlqRYLxyF4k5V1
BzWYgLTh50pYb5RS11/YYPHDt3WYUnM054u4RVKoh6HYOGIYx7iIk3GiJ83YH2c6Jt/JI1abBvk
ZBDX/pxGo5VVqVyaYyXTY5zw4r3RL11LIi5O+52n4gOMyJ4G4awKnx5I4VPIZ17GYPhGk98g4K
OctBtC5hXaPVurZXQX1dwQZNNVhVxm5D4TRRzAo7ERF90eYy3O6x2AlrAMv13IljWKnJ5JYbol
ap6vYTmGnV8OIMOcciF1lgXaDaFc1tIXsCFEF6AVWuErGG7TEGETWUIQ0mD+WhrOkaDv2vc
MT6Dn29cKFuDi3LxH0ODiHajF6yDnnqHcReC7V6sZ5CCz2FQXjaCsGP5VPqChGapS+iCXWluQ
weCP2xMOhqCM285MSeCO3I3KFqCT3VGH1iCd3hvF9mCsXu9cFyAXmTyZvOAe2cSXbaAlGk4
VJOAIWtpS5iAnW2tQsuAr3AHOOfKAz3JxMRKA/HTtKFiBNnd3H2uBhXoaGAGB4HzYb61+LGr7ZII
+dmyiXTR+uW5PVCh+3XAGS0R/CnHPQox/Q3OsOch/inWQMP1/3neBKFiAPnmCH3uAsHujGC
OBKX3Xbv8MXECZbx8o3ltXKx9DXNcU7Z9UXSVSuh9oHXgQkZ9/Hc9OZI+ZXidMOd+3HoFKFI
/W3t/H5V/6n0dGFeAe37HblR6cncKZRV7BXe/XAp7j3hzUyl78nkmSnN8X3nrQe182XrFOV19YHuj
MMp98nyKKF9+iX2AH8R/LH6UGLJ/y3+ubat43H0tZHd5kH1PW3h6On1yUqp6uH2aSgl7QX3XQZ
172H4rOSd8eX6HMLB9I37tKGR90H9gH/B+hX/sGQR/MoB8bPx3c4NYY9B4RILaWtt5CYJgUiB5
nYH+SZZ6PYGzQUV664GCOOx7pIFbMJJ8ZoFAKGZ9KoEulBN984ErGUp+slEvbE52PIlhYxx3IY
hRWiZ3+YdFUX54nIZcSQ55TIWNQN16DYTZOKV63IQwMG17t4OQKGJ8koLzlC59c4JWGYV+
Q4Hla5R1LY9YYmd2IY26WXI3ClwgUOV3t4qtSI14dYIV/QHp5RYgcOGF6KlbuMEI7GoXGKF18D
YScIER9BINmGbZ95oJMatl0RZUsYcF1RpL9Wod2OZDSUGN2847QSCB3vYzsQCZ4mosqOCh
5j4l1MCt6lofIKFh7nIYWIFZ8p4RRGd99moK7aOeMIVhOYDSLOft3V66KTF6pTzeJvMhuRuala2
VHPr6HkWi2NoKG22wyLjiGR2/AJhSFv3NjHb2FSXcsFuSE8XsfaEqJYl6LX62luGEwVz2lCmPeTt
yHR2aiRp+Gj2l7PoyF5mxtNmOFW29rLi6E7HJ6Jh6EiXWbHd6EOndFxyEAXw8Z7SG+mSCXy6
GhmarVtSGCmjeToeFcGsqRI6E4m2lPI6EZm/9NkiEAnJ9LiWDtXUOJiiDdXevHfuDSXptF02DLn0
+Zx+E6Wo/XquEnmv2VmWESW22Ti2Dzm+QRhiDYXF8PiuDB3N5NiiCx3V5LhqCn3eBjJCCgnm
cHhSCdnvbf3eCd34mZoaDEW/yXiOC6HE1Ve+Cs3KBTcuCVnPmRcqCB3VaPfGBzXbeNgWBrX
hflGyBpHnjJjqBpHt6HjaBsX06F7KBx37+ZeuBeHwfXY6BZ3Z3VWOBsndTTVCBCXg4RWWA13
ktPaaAuHo0NdSAsns5LfmAwXxCJkma1X1bHm+A8n6TGBWBE3/LZVx/+XtnXQeABnu2VOWAC
HwITOJ/4XxqRQl/yHzfPWF/xH1mNah/1X3vLeh/+H5+JlaAHn8ZHqGASn/MGG6AdoB/ZMx+noE7
XHI+yoDyVF1+54CwTGI+1YCPRKR+0YCDPRV+4oCMNXZ/CICZLdJ/QICrJl9/eoDGHst/t4DuGLI/
8YEaZDd9dobwW9x9tYYfU7195lVXS9p94lS1RCx97lQpPLp+DYO2NTI+R4NFLbZ+IILYJmJ+44Jr
Hu1/NoH/GPh/goGfY5d8elyWWz98w4tBUyR8/onzS1J9BojNQ7p9HofAPGJ9TobMNP99mYXbLZ
p9+4TqJmN+XoP2Hwh+xoL1GS1/JIIRYvZ7n5lbWq579JA7UqN8OY5jSt98S4y1Q1l8b4siPBI8qo
mrNM59Bog6LYJ9eoblJmR974VTHx9+aYPJGVI+1oJwX/2SYliPWECQzlvLUK2PP18PSRCNxmJ
ZQZGMXmW4OjWLDWkvMsCJ4my1Kzul2HBOl9uH23P5HEmG73fGFiuGKHuvX4yPzF58V9uOd
GE7UFaNHmQDSMmL0mbXQVvyKI2m/OhKJc2y9MrCibW/KKz+Hg3Lol/SGp3YVHHaF3nldFm6F
NXy0XxiNiGQmV3mMX2ZzUAWLN2jISliKEmSqQSul/m2eOfGIAXAmMqCHHnK6K0KGuNvdJAm

FIngNHJ6E7XrTFqiEYX2gXqGLImmaVw6Kkmt9T6eJjG1nSDylg29fQPCHjHFmOcmGrXN9MoyF6
3WVK0KFQXeyJByEpXneHMCEG3woFtqDp35yXi2J028DVqKI8nB6T0WIDXH3R+mHH3OAQK+
GRHUXOZqFgHa6MnKE2nhWKz+ETHnzJC+DynueHOuDVn1rFxyC9X8zXbGITnRgVimHh3VyTt
CGvHaGR4KF6XeWQFqFKHizOV2EfnneMk6D7XsEKzaDcXwrJEiC/n1cHSyCln6IF4aCPX/IXTuG
6nnUVbmGQHpkTmiFkXr2RyaE13uJQA6ELXwrOSWDmnzdMiyDHH2NKy+CsX4/JF2CS375HW
eB7n/AF+SBNYCBXMCfR9H9RVT+FHx9TTfKEiH9aRr6D4H9wP7eDSn+XOOSCyX/QMgSCXYAJ
KyGCAYBFJGyBq4CDHZeBW4DFGDWBFIEIXD2EnISyVLSEIIQ2TWODnIO/Rj6DAoNaP0yCeY
MIOJKCBoLIMc6BqYKJKwqBXoJIJHSBGIIeHb2A2oG5GHiAo4F5W6+DtooGVCEdRYkOTNqCzl
gZRcOCPlc3PuWBvoZoOEObV4WtMZqBCITxKvOAzYQxJHmAmlNrHd2Aa4KUGLCAQ4HaWyG
C9486U6aCjY29TGaCGoxERVyBlrhPo6BH4mTOAGAw4hcMW+AgIckKt+AU4XoJH2AK4SnHfe
ADoNRGN9/84IsVz6YuljCUHaWeFwXSdSUQI9vQxKSPWLDPGaQUGYrNdSOgWmrLyeM2G08
KGeLUHDilciJ2nSUGvOlc3hfFYeHPnw2VvyWSF5kUDaUP2FDSZiSP2QmQuKQYWcOPESOnG
oGNcGM9G0ULyWLBHAXKHqJ/3Nfle+lpXaVGyyHYHncFdOGSX0kVq6UHWPHt/OSQWY8SWC
QbGi1QrWOtGswPCKNFG27Na2LkHBXLYKKJXMAKlml03W3lhCHIHhyG16Gbn5FhWFcX35Vli
SRWj3T6GQiWsJSRSO020fQnaNNW83O/OLr3FbNZCKRXOMLxil+HW8KJSHxXfvl2GpHopG4
mFm3x2Fk6Etn63VguQjm4bT1KO+W/HSMaNZ3F4QjOL43MnO76KdXTgNW2JJHajLwqH73hbK
JuG1HoQIkqFy3vNG72E1n2fFpeEAn9iVbSPFXMvTviNoXR8SGqMLXXIQeGKw3cCO3uJbnhHN
T2IMnmXLvGHD3reKJ6GAXwilmyFAn1pHAAEFH68FwaDRX/9VVuNynhTTqGMcXklSBmLF3n2Q
ZiJxXq5Oz+lhnuJNRKHx3xjLtuGSn07KJ+FSH4QloqEUn7IHEiDa3+7F2qCoYCDVPSMqn19TjyL
aH3JR7aKJH4XQUci435kOvaHtn6+NNyGn38ILruFmH+LKJmEoH/ulp+DtYBNHH+C2ICKF7+CFo
D3VICLtYKMTcCKhYJjRzeJUyI7QM+IHIIUOpag+oH6NJSF7oHtLoyE8oHdKleEBoHllquDJ4GrH
KmCWIF9GAaBooFYVAKK6YeQTUGJw4byRulmYZUQGCHbYW1OjmGVIUkNEyFUyShLI2EX4
QZKHSDfoOllrKCq4LvHMyB6YI/GEGBQYGqU4SKRYx3TM6JI4tZRIOH/Yo8QAOG2IkjOeuFxogZ
NBCEzlcLjaD44YgKGODC4UYIrmCQoQKHOMBjoLIGHOA8IHvTI2fMlKWSHach1xdQrmZ21+tP
OeXKGMsNyGUlmaMMWmSLGocK5CP7m2/JZ6N1nF1H8WL1nUvGa2J6nj4FOylRHy6TkWc2F
5oSfuaaGFBQpyX8WQmPNGVX2cqNxSS8Go8MWeQq21gK56OiHCTJcCMhnPUH/mKn3cVgfk
l1HpbFUGHSn2QTjSatWOJSEoydmX/QoGWLWh/PLqTwmsdNwWRdm3GMWOPUXB+K6mNS
HNBjd2LXHyoICeJjXjXGi6H4XufFYuGch5QTgOY62h7SA6WzWqSQkqUomy1PluSUW72NuGQ
HXE/MVGODnORK6qMH3XfJfOKT3gqlE6lnXpzGmKHDHzDFcuFsn75Tc+XRW1UR9eVTW8QQ
hKTRnDWPfMrdXK0NruO73SYMTqM9naBK6eLHHhcJgWJYnorlHWHxHv6Gp2GRX3TFhqE/H
+QTZmV4XlbR5uUB3N+QdKSHXTmPB+P/nZVNouN+XfKMRqME3IEK5uKRnqzJhOllHwXIKGG/
H12GvCFgn7SFo+EO4AUTWWUqnbvR2SS6nfdQZqRGXjRO+uPEXnPNmCNIXrUMP2LTxvfK5
CJjHzjJh6H4X3flMeGTn7QGzme13+1FviDIICGTSTpHvGRxqR93w3QU2QOHyvO6WOQn08Ni
aMYn3SMNOKnH5uK3ql5X8GJh+HQH+VIOOfTlAXG3WEQ4CEF1GDBoDmTMGSzoCDRrGRLo
CIQOCpfICTO0ONklCyNdOLuYDYMJOJ/IEFK1KITyErJhOGroFHIPOfKoFSG6SDw4FDF5yCj4
E4TFGSH4U5RkGQhoTOQHKO2oRqOuKM9YQXNYGLJYPMMFSJclOIKymHxoM6JgOGLoLeIP
2EsYJ0G8qDVYHtF9qCLIF9S9+RkonWRdqP+4jtQBaoUlgOOpcMcYdINTyKp4aLMB+I+IXWKwe
HVIUWJfaFwoRIIqaETINwG+qC+oJ9GA6B2YG2nwVdo1lxXRgTFXyg/1i7Fm2dpJIY11caUhn9G
EAXChqqGShTxBtemgzQgBwZmu3NQxzWW87J+V2TnKuHAR5ZHJZJnRZJZGFs3j81cQ14lgqBfY
WEVdXBipmP3aFxlMbeW2xoN2nLTn1rVmywQZRuiW+ONMpxv3JxJ9Z08XVQHDr4OnhSm1J
VEGPajlRYpWYFgWxcKmg1dG5fXmpnZ4lioWycWsNI+27VTfppY3EMQTNs13NDNJBwTHWEJ8
pzuXfQHF13NH06mbVRhmw/jO9Vf22mgD9ZYm8Tc3Nc33CZZr1gZnljWiJj/XO0TX9noXVCQNtrU

XbTNF1u/3hxJ75yo3orHIF2TXv9mEROYnRti6pSqnUjfyhW23Xecopamna0Zf9eYHeSWYxiMHh4T
Q5mDnleQI1p9XpINDVt2XtGJ79xsnsxHLJ1i32klv9LpHxgioFQMXxnfh9UpXxzcapYnXyeZUhcmmz
RWPxgoH0OTKVksH1QQEpoync2NBts4n39J9Zw8n6OHQN0+38qlc5JKoRHiXNN9YOfdTRSpYL
2cORW0oJwZKVbAoH0WH5fOYGCTEljeYEdQBBnwoDKNAdsCoCNJ+twTYCJHUt0goCLIKIG84
wSiG1L74q1fFBQz4lbcCJVKYglZAdZhob6WAJd6YXdS+9iVoTSP9hmzIPfM/NrQ4MEJ/1vuIJkHYI
0G4HGk5ZFAJOIh21KHZGVe2VPH4+Kb1xTnl2uY2RYHoveV4FcqlocS5NhP4hwP6FI44bfM+Fqi
YVvkKAtvM4QfHb1zxILako1DP5sQhnxleZhYeo5NmJWlbqZSNZMhYs9W15CpVwxbg44+Sz5gQY
vvP29IEIm+M9Bp44ehKBZuvoW1Hehze4PMkZdBsKJAhaZHCZ7wedIMRpufbg1Q/5hpYIFVvpVA
VqlaiJIoSvhfaY80P0ZkXoxiM8JpV4mnKCBuXlccHgXzPoSdlgdjNVKkiTdlgVZJfINnxVnzb99p2l2RY
19sB2ExVwxuUmTSSrpwvmhrPm1zSGv/MkN13G+XJfV4eHMkGwN7L3bhlEde/1sYh7lhw13+e0Zk
dmDobtdm5WPRYohpZ2bBVmJsBGm3SjhUUWyuPhBxg2+mMg90VHKkJe13K3WfGy96FXjDkrN
bL2NIh5eWGV1eiZhcGeobelkLmnqYchm/mwwVcpp5259ScVs4HDQPb9v53MrMeFy83WOJeZ2
Anf4G1V5G3qEkT9XyWtFhSFbUGy7eR9ewW45bQxhxm/VYRRk2HF4VTxn/XMkSVprMXTUPXV
uc3aMMblxt3hRJeB0+notG3d4P3wij/NUyXMTg/5YmHPheCVcT3S2bDtlfXWqYGLi43apVLZmQXe
ySPdprXi/PTNtJXnWMZpwnnr+JeV0EnxJG6d3g32njsZSJnquguhWNHrXdylaKXsFa2ddpHtWX71
hKHuxVDBkunwYSJZowXyIPpdsA30GMYhvrX2VJgFzUn5OG/h27H8SjahPwYJBgehUCYg/dkp
YN4FCaqt5IDpXyRfmoCcU7ljXIBdSEBnKoAtPMNrAoAQMXlu2YAHJhtyrIAtHEJ2bIBbjJZNolm7g
PFSFoiLdXFWcYdgafNaSYZfXo1eKYVrU0RiFoSIR+1mEIO6PJqFYMCMWpuGIJgJjByGIHtHIB1
/oF/i5ZLyJD/gAFQXY8jdJNU1o1OaTZYzouuXfJc0locUstg4YicR5ZIBlcZPF9pNYXjMVptZYSkJkBx
IIOPLV1o4J+ippKlpgbfx1O0pWec8ITZpMnallXeZDIXWNbml6zUltfyoySR0dkEYqNPdFobliiMUx
sxobHJk5xIYUNHOF1VYNdiatlRj7/fIFNeJvrcx5SJ5jaZ/dWUpXvXOpai5MWUf5e2pBQRwVjQ42tP
ApnwosoMUBsQli0JllwwlZeHQZ1FYQcjRlo4FL0gQZqzVaHdRRssVohaTxuZ122XYtwLWFSUgZy
CmT2Rn10CWizOvd2KWw/L5p4U2/tJCF6jHOXGhR82Xdzi5Rk3lrx7hnOF3Qc/1phmC1aFJrlWO
hXMttsGaXUXBv22maRg5yHmylOqt0eG+4L3N23HLSJB95TXXrGj17zHkvijFhPmKrfolj+mTfcwFm
pmcZZ4BpAmlIXCFrZ2u7Uott224eRatwXXCPOmly73MNL1J1inWUJB54MXgeGmB63nrKiO9d/m
o9fXNHe2vGchIkFG1YZr9msW8IW4VpVHDEUHFSA3KORVFuwHRiOi1xi3ZCLzR0XHguJB13NX
osGn96DXxEh8ZbHXGtfG9ed3KWcTlhu3OFZgRklHSXWu5ncXW2T/tqVnblRPttSngdOfdwS3lhLx
5zTnq0JCd2VnwlGq55VX2phrlyknj2e3NcJ3lBcFJfpXmSZT9isHoGWkxlwHqJT35o2XsdRKNr/nu+
OcNvMHxuLxNyYn0uJE1IX4PGwF4t378hbRWRoA3eohaEH/fb4NdwX+LZJBg/H9cWb1kO389T
w9ng38zRFVq1387OZZuNX9WLwpXk3+CJGV07n/UG0t4MYAthLNUPYdkeaFYMYZjbrhcDIVnY+
NfboSWWS9i1oPYTqJmRoMxRADpxYKhOWptT4InLv9w14G/JH50WoF6G4t3voE8g79Sfo5ceLx
Wjly2beVag4sXYy5eA4msWJphiohXTi9IHYcaQ7dow4X2OTtselTpLvNwKoPpJJBz14MEG8B3Xol
ngtBQ8pUyd+JVF5LvbSJZI5CyYoZcvl6sWBBgX4y8TcVKElrnQ2xn2oktORFruleLLudvklXyJKBzZI
RsG+13DIL1ge5PlpvVdyBT0Zj6bH5X9JYkYfpbo5N8V5xfXpDtTWxjKo57Qy5nFlwqOO1rFYn0Lt1
vD4fHJKxzBYWpHBJ2yYOkhBtusIM2eMFwN1a9bY5xtlpKYn1zDV3ZV5h0bmFxTOZ13WUXQjV3
aWjEN4p5D2x7LQl6w3A9ImN8jXQFGTV+ZXgAgtRq2FrNd6lsxl2sbKNUqmCQYblwVmN7VvhyCG
Z1TGpzxGmCQdx1kWyeN1J3bm/ILPF5WXL8Imh7W3Y0GVt9ZHmWgalnYmlydqlprWRWa85r6
maaYQht3WjxVmpv12tVS/xx2m3LQY1z43BUNyB19XLxLNx4FHHWWIm16S3hCGXx8gXsNgJtkT
Gk0db9m6mrWawhpdmx/YGVrpm5DVedt2nAXS5dwF3H8QURyWXPYnVR0oXX3LMI283gllnB5
WnoqGZh7unxlf5xhh3A/dN5kanFGakZnOHJVX8Npn3OFVWRsCnTFSzFufXYYPxw8nd6Nslzb
XjrLLx173pqln94gXwCGcZ7Bn2rfqtfGXc0c/piMnemaXRINXgeXw9nzHi1VNBqaHlgSsBtC3ohQK5
vsHryNp9yWnvWLLh1B3zGlqR3v33RGhp6Y37mfcBc534mcyZgMX35aLhjZH3RXm5mJ33MVExo

8H3dSltrv34HQQhukX5GNnlxZX6cLLV0O37+Isd3F399GmV51oACfNVa9oUHclBeZ4Q0Z/lhwINo
XcpkqlLJU8ZnloJCSfRqjYHXQCFtilGGNIJwhoFNLK9zg4EgluJ2goELGqZ5XoD9e/FZS4u7cXVc1
YpHZyXgR4jbXRljSoeoUzRmVYaQSYVpbIWXP9RsJYS4NidvtoPxLKVy24Mxlvd1/4J/Gtx4+oHWe
wxX2JJPcKJbc5BAZmxe9447XHRiD4xzUq1IM4rHSR1oZ4k8P4xrllfONf9u/YZ2LJxyR4Uilwh1joP
SGwl4pYKSejJWmpiub+RaQpYMZcpd1JNxW+lhAJENUjtkOo7JSMZnh4ypP1Bq64qqNd5uYljDLJ
Nxy4bglxZ1L4T7Gy94X4Mze0N0j1NzcJ51ulbmZiZ23VpiW+B3y13sUcl4vWGGR+p5t2UxPhF6xmj
sNEJ77Gy2Kpl9I3COIMB+eHRzGGZ/2HiKeitw5Vqsb7Rybl2AZWhz7GBcW0F1K2NKUUI2bGZM
R4I3sGlnPc94/myXNB16WW/aKo97xHMoIMx9UXZ/Glp+4nn9eTJtnWGQbuBvemPLZLlxSWYRW
rJyymh2UNh0T2rsRzN12W14PZR3YXAcM/x47XLYKoZ6i3WcINZ8S3hpGKI+CHtTeEtqtmg2bhxs
3GnmZBdu8GufWi1wqW18UG1yZ29qRuF0KXFuPVx153OGM913pnWyKn55dHfoIN97Y3ovGMV
9SXyKd21oF27ebVtqe2/7Y3Jsy3EiWaNuuHJyT/5wqXPWRo1yn3VRPSJ0j3beM792fXh8Knp4eH
omIPN6kHvlGPJ8mX2zdppllyHV8bJFoXHYOYrZq23anWQBs9XdmT3hvFHg8RidxOXkrPN5zVno
uM511cHtFKn13k3xmIR15zX2bGUh78H7XdcprXwfa9Nmb3wWYgxpHHwVWG5rYHw/TwBtqXyC
Rcxv+HzipKFyP31bM390gn3rKn92yn6DIUR5I38uGZV7Xn/edPZh0YK2axBkuILLYVtniYFqV9Vp7
4EAToJsW4CyRW1uz4CGPGBxPoB1M15zqYB8Kn52FYCKIWN4joCkGdd64oDEdCNgO4kiakBj
OYfcYJNmIYahVyhon4WqTfVrJYTSRQFtt4QdPBdwS4OFMzZy4IMDKnh1coKCIxt4DIICGg56eo
GMc1Be3Y90aXdh6Y2YX9dk4lVlVoZncYpBTW9qDojbRjxsuoecO9JvcYZ6MxFyLoVsKnF044Rcl
Y53m4NBGjx6lOl4cohdtJWaaMNgypMpXzhjy5DEVf1mbl6oTP9pHYyweREhr4YriO5lut4k1MvJxloe
eKmt0a4YDIZ53PYRXGmN52YLLcph6e1O5aMx7DIdAXy57nlrNVbl8GI5YTG98nmHzQ1d9MGWi
Oj593WlflMSh+p20wKDt/gHETHyOAdXUQF4qBdXk7cat29lqTaAJ36V2GXod41WB+VS95nGN3T
AF6aWaEQwZ7P2mqOgl8JmziMQ99InAwKDx+LHOMHzp/VHb+F7qAgHqScM9z22EdZ0d1HmO
DXe12V2XvVLF3W2hmS594Z2ruQr15fG2M0dl6mnBAMPh7xHMKKDx8/XXhH09+VHjJF+N/qHv
McAdxJWdrZppyqmlRXVx0lms8VDp1WW0zS0F2mG87Qnd34XFYOax5MXOEMOJ6inXCKD178
XgNH2B9cnpXGAh+63znb0Fusm2xZe1wcG8UXMhyHnB7U79zhXHust509HNzQi12bHUOOXt36
XazMMx5bnhnKEB6/XopH3x8pHwJGD1+O330bolsgnPyZTpucHTUXB5wTnW4Uyxx33aiSmZzeH
efQdN1G3izOUB2wnnWMLJ4bXsHKEp6H3xFH697432cGJp9kH74bdBqh3o+ZJBsonqMW4VurHr
cUqhwZHs4SfpyJXuqQYJz73w2OQx1u3zUMJt3in2EKFN5XH4+H9x7On8OGO98+3/fbRjoxoCIY
+BrB4A3WuRtNH/qUh1vDH+5SYhw7X+hQS1y2X+mONR0yH/AMIF2u3/tKFh4rYailAJ6p4BkGTZ
8flCqbFdnRlarYyZpnYXEWi9r5ITkUYBt0oQrSQdvzIOOQMxx04MQOJNz44KnMGF1+4JPKFh4D
oH5ICB6JoGkGXJ8E4FXa5hl74y1Ym9oXos+WYRquInOUOxsuliLSI5ux4dmQHBw5oZiOFZzFoV
0MEJ1UISVKFd3g4OylDh5toLFGar7uYHtatxkxZKTYclnSJCmwPRpto6KUHBrxoy4SCht54sHQ
CNwHII7OCNyZogIMCI0v4ajKFV3DoU7IEx5WYPBGC57b4JsagWAdFP/YRuAgVelWF+AilsZT76
AhV6qR0OAjmJMPvOAqmYBNpmA7GnHLjqBUG2hJgWBw3GQHaaCUHWIFsOC6nnjaT59G1p8
YG99g118V8995GCCT0h+JWOMRuh+cWapPrJ+zWndNnF/RG0jLix/1nB+JhCAeHPrHciBM3d2F
v6B9XsealF6K2CyX89632MwV0x7iWWzTt58AmhBRpZ8h2rgPnd9HG2UNk59xnBaLiB+hHM2Jhp
/UXYgHeaAOHklFzGBHnw8Z9N3oGavXzh4kWi1Vst5c2rATnV6HGzXRkR60G7+Pj17IHE3Nip8a3
N7LhN9VnXMJiN+THguHgB/WnqxF12AYn0+ZyV1VmyeXp92em4rVkh3j2++Tgh4Y3FZRe55QnM
EPf56MXTCNgN7MnaDLgV8RXhOJi59YHoqHiR+j3wrF5p/sX4wZoBzTHKEXf90m3OdVbF123S3
TYZ213XPRYN33nb4Pa949Hg2NdJ6Gnl6LfR7TnrlJj98iHwkHI59z32eF/5/An8VZd5xcHh6XWly63
kGVSp0VHmTTRF1dHokRSR2n3rJPWI32XuFNaZ5IXxLLeR6dX0dJk57yn37HpN9J37vGFI+a3/h
ZTlvyH5zXM9xZ35mVJty9H5cTJV0Mn5oRL91e36KPR521H7FNXZ4PH8MLdB5rn9hJII7H3++Hr9
8IIAoGKZ96oCRZJduWoRLXCpwE4OxU/hxulMcTAhzC4KmREt0bIHPMd13oIDNT53Y4HNLbh4

9YGhJI96g4F2HuJ8E4FLGOZ9foEoY+5tFooOW4du5YjqU15wn4fMS4JyA4bRQ91zd4XvPHV1AI
UrNQL2oIRzLZ94UIPEJmN5+oMRHv97pIJTGRx9l4GpY0Zr+4+oWvJt4l33Ut1vroxMSxJxIYrIQ4Fy
polhPDB0QogaNNx1+obiLYt3xYWYJmZ5h4R9Hxh7R4M3GUI82IIWYYeGf1RAWYmGCVfDUbeFj
ltQSe2E/F7sQkOEF2KZOryEHWZaMx+D6moqK3WD4W4PI/SD6HIMHEeEB3Y2Fg+EOXqBYOK
DV1pmWPqDNF1oUT+DCGB0SY2CuGOUQfqCeGbGOouCUGoMMwWCT21iK3SCcXDMJAqCo
3RJHHSC7nfrFISDRHuhYEiAj2BLWHaAtGLVUNOAy2VoSTaAr2gSQbmApGrMOI6AsG2ZMu6A3
nB0K3KBKXNiJB2Bg3ZhHJuB9XI+Fo+CbHylX7B+K2X+V/N+h2gXUGR+02o5SNx+4mx0QXR/A
m69Oi5/OHEWMtR/j3NyK26ABHXZJC6AhXhRHL2BG3rvFsOBsX2PXxp8BmudV258kW1HT/J9C
m75SH59QnC/QSp9iXKUOfp95nR3Mrd+Y3ZWK2h+/Hg5JEB/nnovHOiAUUnxNFweA/35nXoN6Jn
EtVuB61HJtT297cH0wSax7zXT6QM18OXZTObh8u3e7Mo99WHkjK15+D3qOJFh+ynwIHSp/kn2
hF3KATX8xXfl4Z3bPVmB5PneJTvl6AnhlR6Z6gXkSQHt7D3ntOXx7sXrZMmt8bXvJK1V9Pny+JG
5+EH2/HWZ+6n7VF9J/sn/iXW921Xx3Vdt30XygTnt4uHzRRzI5U30cQCF5/X15OTp6vH3qMkN7k3
5hK0l8f37fJH99aX9iHZd+V3/yGCR/L4B7XOR1fYICVUt2koGpTet3kIFYRrt4P4EmP7p4/oEIOO15
1IEAMhN6xoD8Kzd7zID9Jlp80ID9HcB914D7GGH+wYD9XFN0Uod+VLx1eYakTWN2iYXSRkN3S
IUgP1d4GoSEOKN5BIP+MeV6DYN+KyV7LYL/JJN8SYJ8HeJ9Z4HrGKF+ZYFtW8RzUYzYVDt0ilt
0TO91p4oYRd52dljjPwR3VofHOGZ4UobFMb55clXJKxV6poTOJJt72IPNHf59CoK5GNF+GIHKW
Q6Me1SHUgOLXVgfSyCKQVu9RCqJMV9hPUyIPGMVNoqHaWbeL62GyGq3KMCGUm6llfaF7X
KoGvuFn3bWFWuFbHsXWI+JhFpZUZGItl15Sr6H5WCfQ92HD2PSPRWGUWcTNmeFsmplN6C
FPW3LKMqE7XFCIheErnTIGzOEiHhvFbmEdXwdWBKG5V/wUSmGWGKhSmmFxmVZQ5mFI2g
bPOGEImrrNkaEJ23ML5SD23C6KNKDr3O5IjSDk3bDG2ODkXnnFfyDnX0JV5SEpWVYULmETG
eiSgmD62nyQ0uDcGxMPKeDCm6yNiGCv3EIL4OCmXOYKNIcknYQlk2CmniWG42Cuns+FjeC4
X3bVx+Cm2qwUE2CcmySSaiCPm54QvmB6nBmPGiBqHJdNfaBf3RfL2+BenZYKNqBk3hQlmeB
uXpXG8CB8nyAFoGCLn6bVqWA1m/2T9iA0XF0STmAw3LyQpmAknRoPBmAc3XoNb+Aa3d0L1
GAgXj6KNmAsXqAloeA6XwPHAmBMn22FvKBeH9KVjJ/OHVLt21/WnZLSNZ/cHdKQkN/YHhFO
9N/YHINNYx/dnphLzV/pXt1KNd/6XyKIqSANH2lHEuAin7NF1iA2n/jVbx9yHqmTvl+DHscSGI+Q3u
VQeJ+THwWO4R+Z3ymNVJ+I31DLxN+3H3jKNB/NH6FIrp/kX8oHIJ/+H/NF66AVIBnVUF8jX/mTn
h86n/jR+V9OH/jQW59VH/sOyR9goAENQp9xYAsLuZ+IIBVKMF+jIB9lsh+/YChHK9/eIC8F/Z/5YD
WVMF7glUZTfZ77oSdR2Z8TYQjQP98d4OyOsh8tYNQNMV9CoL/Lrp9d4KuKLJ994JaltR+flH+HN
R/CoGRGDJ/h4E1VEJ6olouTYF7GYktRvx7g4gwQKJ7uodBOnt8BoZINlt8aoWbLpZ86lTRKKR9e
oQFIt1+D4MxHPJ+r4JJGGR/OYGEULuSi1S6SqWQvIhtRLGO/VwiPpGNZF/ROIGL72ORMoKKp
WdjLGWJj2tHJjSlpm9AIB6H0XNIGc2HFHd0FNiGfXuiUGOPv1o+SIGOQl1/RGaMyWDFPliLYmQ
OOFqKHmdjMm2JBGrGLGWIFG45JkmHSnG+IEqGlnVMGg6F/XjyFS6FhXyQUAmNQF+LSgeM
AWJoRCiKxGVHPiaJkWgmODWifWsPMImHjG4FLGOGv3EHJlyGFHQWIHCFf3crGkeFCHpQFX
mEq31kT6eLIWStSauKEmcsQ9eJA2mtPeOH92wqOASHBW6vMjyGM3E9LFyFh3PHJmqE/nZRI
JGEinjjGniEMnuNFbqD734gT1WJLGnDSVilUGvfQ4aHcW3+PZ+GiXAUN9CFuHlwMhyFBHRSFL
GEdHZIJnWEB3hyILODrHqGGRkDa3y2FgqDO37KTv6HeW7FSP+Gx3CDQy2GEXI+PVCFTHPg
N4+EnHWKMe6EBXc5LDuDjHjeJnyDLnp+INuC4HwfGwOCqn3PFoGCgX9hTqWF+XPSSKuFbH
UYQt6E2XZZPQqEM3eHN1aDoXi7McWDJnn3LCeCwHsvJoKCb3xkIP6CLX2WG0yCAX7KFuuB
4H/kTkGEq3jkSEmEPXmnQoCDx3ppPLaDOnshNxCCwHviMZKCXHyRLAqCCH10JoCBxH46IRm
Bj375G4iBb3+wF0WBV4BVTdWDj33bR9ODOn4tQgeC235+PEuCXn7DNreB9X8RMU6Bon9oK+
CBXX+8JnOBJ4AMISqBAYBRG7eA8YCEF5CA5oCzTWCCooLJR1yCXyKpQZGCDYKGO+KBn
YJWNI+BQYluMQuA/IIQK7aAxYHuJmSAnYHEITaAhYGOG9+AhYFCF9CAhoEDTO2B4YecRvC

Bplb8QS2BXYZbO4uA+IWzNhaAqIUVMNOAb4SCK5KARIPqJliAKYNLIUGAHYKjG/+AK4HkGAS
AN4FGSD+YrIT/QxKWd1itPgiUP1xiONSR+2AnM6KP6GP6LnaOD2feKSSMbmVSI7SLAG/ZHIWJ
rnPnGK2lengQFE6HfXwqSAmWEFoxQtyUJF13PdWSMGDHOLCQGGQwM46OMGegLnCMf2sb
KTGK/26jl9aJq3l4HoyldXXOGPiHY3l0FK2Ggnz+R++Tn187Qr2R9mliPbSQQGURQJKOXmgWM
3mMpGshLmmLGW40KTuJtnFPI/Sle3RyHryHYHeRGtqGbnq3FP+Fp327R6+RmGQfQnuQH2at
PXK0mGIDOFyM2mvuM1OLP26bLleJzXFLKT2lh3PwJAuHa3aQHuwGbnkuGXKFmXvbFUWE6
X5iR3aPv2jxQj2Od2skPTONHW1fOCeLg2+oMyqKBXHyLkClrHQ7KTuHf3ZtJB+GeniRHw+Fk3q1
GbOE0nzpFZuEM372Rz+Olm2qQgGNBW+GPPSL1HFIN+yKXXM/MvmJAHUaLh6Hw3bzKS+Gp
Xi9JC+FqXp5Hz+EyXwvGguED33mFheDdn93RwyMuHJvQcqlwHPZPLuKs3VEN7eJW3atMs2IG
ngXLf+G93mCKSOF6HrkJDyE8nw7H2mEGX2HGlqDzn7HFoaC0n/mRsSLh3c3QX+KrHgmPG6J
unkYN3KlenoRMpKHUHSLLdSGQHwHKQyFPnz/JD+ET33tH4mDfn7LGpuC1H+UFuSCR4BERm
KKkXvjQRSJy3xrPACI63z1Nw+HuX1/Mj+Gnn4LLZOFnH6YKOSEpH8eJDWDvX+ZH5yC9YACG
s2CV4BPFzKB1ICTrfSJyICPQKWJDoCpO5KIOoDENq2HE4DfMeyGAoD6LVKFDIEVKLuEHIE
JCeDPYEsH6qCfoEgGveB7ID2F3SBcoDVRyqJJlUqQEGlclTDOzaHo4RgNluGhIQJMaefFYozL
RyEkYNdKJmDqoL+JByC0YKRH7WCG4IYGxqBk4GEF6qBloENIExZOE2Th6Fbx1GLew5eYVW
MbnthMlmTYg9kKV2cVdVnTmGISahqlmWhPYht/mmQMYlxc21/JWJ0+XFaGpd4IHVckmZUIIZTh
gRXNVmLebxaUVzMbW5dmWAZYT5hAGNsVTNkjWbFSS5oNGoXPTJr8m1kMV1vt3C3JWpZg3
QFGtx3WHd3kK9PbV7RhJdS+mFOeJhWkWPVblBaSGZ1YIFeGGkZVKJiB2vBSMFmBm5qPOZ
qFHEVMTduJXPLJXFyNHANgxl2QXlujvtLXWb9gy1PV2jCd3VTv2qTa5ZXaWyMX8lBjm6LVBVfy
nQCSFpkE3KVPKBoaHSfMRVsvHa5JXdxC3juG051TXs/jYIHs27uge5MCHAQdmtQYnE6arxUyn
KKXx1ZP3PhU5Jdw3VAR/xiUnahPGRm6ngMMP5rf3mKJYhwDXsyG450hHzxjE1EfXajgNJJH3cg
dXFNyHekaexSenhNXnZXNnj/UxNb/nm7R6Vgznp+PDRlpXtRMPlqenw5JbBvSH1TG+tz9n58iydB
l35Jf9BGfH4ddJVLax33aTVQYX31XeNVYH39UqRaZ34QR1hfdH40PAtkhn5tMPVpln7AJdVun39
KHD9zf3/hihc+9oXSft5EEYT7c8JJNYQqaIVoa4N+XVZTpYLdUjly5oJKRw9eLYHPO+ZjeoFwMP
Now4ErJfVuCIeGHIZzGoEfiSQ8mo0effhB2Yufcu5HJYomZ9RMjYjaXMdR+4eZUctXb4ZnRsd7o
VRO8Jid4RbMPNn/IN9JhFtf4LVHMJyxol1iD46cJQ6fSM/05Ifci5FRJAJZzJK2Y4ZXERQdYw1UWh
WF4piRoRby4iwO6NhjYciMPNnSoWrJiltBoRkHPRyf4Mph2Y4f5sSfGc+CZhpcYxDoZW+ZqtJYJM
kW9dPJJCZURVU8l4gRk1a04vSO4hgx4msMPNmtlegJjxsoYXEHR5yRIP6i9VerE4Xf9xg5IH8dAJ
jKIXsaDNloFnrXltoN13tURRq9WHyRaBt22XyOjZw6GnuLvd0Am3vl553MXHlGbf6bXYHihNZ8lZe
fmJcplmPcs5fYlzKZzliQGAaW8ZIPWNxUHxoX2bQRTJrnmoxOe5u+G2ULtdyXHD/l6t1znRoGfJ5
Qnf4iHhViV5vfQhYsWDtcbZb32N3ZlhfIWYiWxVif2jTT/VI/2uLRM9pkW5MOa1tNXEXLrtw33PrI7Z0
kXblGil4OnnHhwNRmGY8e9BVK2gOcLIYwWnsZYhcXmv2WnBgEG4JT3dj23AIRHVntnJHOXRrn
3R0LqNvi3avl8BzeHkCGlp3U3twhbpOGG3WerVR/G8Kb8xV5XBIZMRZ1nGxWdVd1nMjTwJh6H
ShRCNmCHYmOUJqM3e2LpRuXnlal9VyhXsiGpd2kHz/hJRK/nU3eapPKXXPbt9TWXZvZAJXknc
5Wtpb2HgNT01gLXjuQ9RkjnnbORlo+HrYLPrtYHvqJAFxwH0kGvV1+n5ug4BILXyPeLZMI3yDbg
5RCHx/Y1VVhXykwLFAhZVTiZen30WQ49jPH1pOPZn4n3SLpZsgX5SJCIXF38AG0p1fH+5goF
FpoPLd9NKQ4MZbUhO6IJvYrBTn4HwWC1YYIGBTcNdK4EjQ01iAYDdONZm34CzLpdrtyCfJEtw
glC7G5F1EIDfgaBDblRfdv9ILYl1bIVM+lgsYgxR24cUV6hWyoYLTv9bxoUVQwtg0YQ7OLZl6YN/L
phq9oLWJGhv+IJWG810toHfgM1BcJGTdxGT0+ta9VLOY3NYXZQQowZVy5VWop2TQNagYjmQ
s5fv4d3OJlIDIYnLplqTYToJIFvgoPNHAB0a4K+gAo/s5godZFErZWya0FJtZM9YPhO4ZDpVshUHI
6mTLdZaox6Qpxe1lp1OIJkUliTLpppvobEJJVvH4UWHCp0LIN+g2ZkOE6HeCJmG1JbbQJoCFY8
Yf1qIVoxVxsU14uTGhupWlyQbJxl2Y5NwFzy2pELIN2gW5YIfV5TnJqGN18IHargdxfr1ZgdtNiBV

mJa+5kZly+YRdm7GAOVmRphWNpS91sNmbRQVBvBmpCNsdx9W2+LG9073FCIgZ3/HTIGRh7
BHh0gHNbgl4BdaJeR2CDavRhFWMSYEpj+WXFVcFm72iDS2Fp+WtOQPltGG4oNpNwSnETLF5
zhXQHlhV2z3cBGUx6CnobfytXxGVwdl1a7WdNahJeHGk4X5JhU2tUVTBkmG18SvRn7W+yQK1r
U3H0NmZuyHRFLFByQnajliN1xXkUGXh5L3uffgRUbGyuc49X4m32aTxbXm9JXuNe43DLVKZicn
JbSoxmDXP6QGZptnWINj5tbXdeLEhxl3koljt023sQGbn4cX0LfQFRbXPAcqFVJXRzaGZY5HUwX
jBcrnYZVBdggHcRSiFkXHgbQB9oRnkzNh1sOnpeLE5wKnuamp0Fnz3GhJ31X5gfAhOuHrLccVS
qXrgZ6hWpHr+XZJarXtJU5levnulScNi2HwWP+Jm/XybNgBrK303LFRvUX3llpZza364Gmd3T3+T
exxMTog8cPJQbYEwZvJUI4CsXPIY14BbUx5dH4AdSWhhbn/3P6Zlzh/qNeRqMn/3LFlui4AVlRty1
YBYGq9230CjekdKNlhscCpOcodlZjpSvYYuXF5XJ4VIUqJbm4R4SQtgGYBPB2pkqoMINchpR4Kk
LFxt04lyltlyToHbGut2f4GOeX5IWI70b3FMr41AZZJRfYUtw9FVo4oYUjBaPYizSLde5ldqPzNjplY/
Na5odIUvLF5tMYQrlvNx2YM8Gx52L4JbeMZGupVJbs5LJ5MFZQVPpZDGW1pUVI6wUdFZElyzS
HfD24rWPwVixYkcNZlnwoeALF9sqYXvIwhxelRwG0h17YMKeuBp207pcFBRyIKyZels9VaHW6dut
Fp0UY1whl5rR6Fyb2JuPbd0fmZ8M9d2sWqWKid49G68lGF7VHLsGBh9s3dJeZllg1ZkbzZneFmK
ZPxpeVy7WuFromAJUO5t12NIRypWHWbUPWdyempUM610723IKiB3c3GAIHd6EnUjGE58pXjqe
Gthjl2UbjNj62AfZCVmUmK2WjBo0mVyUGFrYGG8RsFt/WsWPSFwpG4IM4dzWXEOKhp2GnQfll
p483c3GH17uHprd1ZeAGSYbUlgua2aJY2VjfmIHwZNmS2q1T+VpJGzxRmRsCW8/PONu9nGgM2
dx63QUKhV053aUIJt39nklGKZ66HvLdlpay2t+bG5d0mzbYqxcg4W5FWPpj92/hT21nFnGORgpqP3
NPPKdtbHUGM0pwn3cDKhZz03j0lLd3FHR+GN96L30XdW9X8HI/a5ZbM3MPYepf3PpWFdh1XT
yTullINXYORalonXdAPGlsCHiHMy9vdXnhKiFy33tJlOp2TnzKGT55jH5TdJFVWHj5atBY0nkxYT1c
VXlyV8Vf53nkTnVjgnprRVNnJXsNPDJqyXvHMXlubnyZKixyCX14IRl1on5xGZN5AH9vc7pTCH+ca
g9WrX84YJNaXX7dVzZeln66TgJh8X6wRP5lx37BO/xpon7wMwFtfH85KjRxSH+OIUB1C3/5Gdt4i
YBpcvBRcYYKaU5UylURX+BYllQkVqBcglNzTYpgeLcRKdke4JkO8RoiYIKMuhsmoHKKjhwmlG
RIWB0hoFIGhh4JoFCcitPSYxUaJhTGyrNXztW+4lQVhZbClgQTR1fJlbrRFZjToXoO5Bni4UEMtBr
0lQ5Kjtv/YNyIXp0EoKwGkt30oH9cXJNxpJnZ/ZRpZBTXrFVml5JVaJZw4x0TMFd/4q/RBNiTIkvO2
Vms4fBMrxrJoZuKj5veoUglZBzsoPRGnV3jYKdcnxvik9DaJ1ww1L5Xu5x/Va+VXJzWVqnTCB0wl6
dQv52OmKjOeJ30ma7MNB5iGrIJ+x7U28fHuZ9QXNsF2B/KXfjcWNRXlZmZ69s/vmBXilupVyoVMx
wZl/3S5tyL2NaQpx0AWbTOaR15GpkMLZ32W4MJ/F54HG/HwJ8DHV/F5N+KHlfcGJnol0vZtVpn1
+3XXVro2JOVDhtt2UWSyVv02fwQkZx+WreOW50l23oMJ92VHEMJ/Z4lXQ5Hxp6+ndvF799Rnq8
b31kUmPEZhRmmXBXNdo8GfNU7ZrTGoQSR9tsGxlQftwHW7POT5yiHFQMlt09nPoJ/p3cHaJHy
56CHk5F+Z8f3v6bqhQmpOZvtj12u9XDpmcm07UzRpEW71SlhruHDDQa5uZ3KnOQ5xEXShMH
hzuHavKAJ2ZnjHH055LXryGB57y30nbdxeh3C/ZJ1hVHGoW41kKHKeUqBnBHPLSeBp6XUOQV
ds1HZrOnhvuHfhMGVyl3lsKBN1d3r/H4V4ZnyjGH57ln5LbRdcDXcsY+xfDHeCWvFiFXfkUhtlKXiA
SXVoR3k2QQlranoKOKhuhHr6MFRxl3wDKCN0pH0TH7h3uX4xGNR6kX9RbFJZ232GYzldAX1D
WIngMX0OUZVjdX0dSQpmw31lQLtqGH2UOHhtZn3/MEFwrX6EKC9z6H8OH+F3IX+iGR16FoA2
a5BX9YOvYn9bMYLeWaZee4lbUQVh4oGiSjhlVoFIQGho1YEQOENSVoD4MCpv1ID4KDVzPoD
4IAN2nYD4GVt5r4D8atJWR4m1Yc5ZllhbWQRc8YcQUH5gd4YOSC5kDIUsQBtnsIRwOBjRYYPT
MBVvEoNMKDlyqYLCIB52K4luGY55WIGnaiJUzY+DYTFYLI2kWHxbnlvRUA1fPYpCR9Vi8ljWP9
pmtoeTN+lqklZyMAJuboVoKDxyLIRaIDZ1zIM8Gbl5EII5akF1Nk+cYTp14FNoWGN2mVdAT7B3iF
spRyN4iF8fPsB5n2MnNlt63GdALfp8PGtvJcd9rm+yHXN/PnQUFpyAx3icaU5xNFZcYGpySVmZV7
Vza1zgTyB0umA6RrF2FWOnPm93fmcPnit5AWrClet6nW5yJdZ8SHlXHZl+EXYIftl/ynn6aHVtq1z
YX65vGI+IVxZwk2JETp5yL2UeRkxz2GgJPid1jmsHNgB3Um4eLd15J3FOJeN7CHSKHbp9BnfZF
w9+63s5Z65qimMZXwNsRGVJVohuCGeETipv6GnhRfFx02xOPeZzy27PNdp1zHFiLdF32HQHJe

557Xa6HdZ8G3mFFz1+KHxZZu5ntGIFXlprWr2VftrsWyxTbRt0G6ORZVv+nB7PaRyLnJ9NbN0a
HSMLeZ2qXarJfx47njWHf17RnseF3x9dH1pZjxIJ29YXbNnVXCRVV5pj3HQT5r6HMmRSpuSnS
PPVZwnYPNYVzJXegLb1mHIAJhN4CHrsHjt6g3ytF+N8yn5rZY5i1HV0XRVINHYfVNFnoHbQTLd
qLnecRMpsxnh/PRFvZnl7NVxyBnqMLa10pnuwJih3PnzcHnR52H4ZGEB8Nn9SZOBgwHuIXHVjS
XucVEFI4Hu5TD1om3v9RGlYHxbPMtuLXzVNTFW/H1mLaBzyX4LJjh2iX61HqN5RH9qGI57uYAb
ZDZe84FxW89hl4D6U6RkS4CNS7InJ4BQRABqD4AtPH9tA4AnNQNV/4A5LY5y/IBdJkN15YCAHs
I4wYCiGNB7UYDGY4xdWlc3Wy9gFIY9UxBi4IVMSz1I2ISKQ55o34PkPDIr9YNeNNdvG4LvLX1yR
YKPJkx1VYIsHuI4UYG9GQd6+YFbYulb74zGWp5ew4tKUpRhp4nWStZkt4iMQ0xn24dhO/1rD4ZZ
NLNuWYVqLW9xqoSLJIN03IOmHwR39IKxGTV6slHYYYih67k/0Wfx7HFPFUgB7WFehShp7y1uPQ
IR8V1+MORF9AGOZMwJ912e4K05+2Wvul8h/6XA8HB+BFXS0FeqCPHILYVV3IFZWWUd3slmiU
WI4Ulz5SZ95IGBIQfZ5/2PhOnB69GdxMt98DWsUK0p9Rm7PI+B+jXKcHE5/7naKFjGBQHqLYJ1z
xlyAWKd0rV9JUOF1n2leSTF2tmUQQaJ33WgQOjd5GWsjMsB6b25LK0Z73nGJI/V9V3TWHHd+6
Xg9Fm+AY3utX+1w0GJzWA9x/2TFUGNzOGciSMp0j2meQVJ19WwpOgB3bG7FMqR4+nFsK0J6
nHQiJAd8RnbnHJt+BHnMFqV/onyyX0NuKWhLV3tvkmopT+VxBmwRSGJymG4VQQF0OHAnOch
15nJKMoV3qHRyKz55enajJBx7T3jjHMh9M3tHFut+8H2IXqlrx24JVudtYm95T1lvCHDuR+tw0HJy
QKJypXQHOYV0h3WtMI92eHdbKzd4dnkTJDh6cXrXHQ58cny2F1h+Qn6JXhFpmnPvVlXrZXS9Tt
xtO3WrR4BvNnasQE5xPHfAOulzTXjoMj11a3ocKzF3kXtdJFF5rnymHUx7yX4DF7t9rH9SXXInp3
mfVc5pm3n2Tlrm3pVRxNtwHrVP/dv8HtpOQtyK3wTMhl0cnzNKyl2wH2UJGZ4/35gHYB7Nn82GA
59LYABXORI939DVTpoB38XTc1qJH70Rptsah72P5duuX8OOMZxGH8+MfBzhn98Kx11/H/FJHZ
4X4ANHat6tYBTGFN8w4CWXE9keoTKVKtmoYQiTUho1oOARitrM4MCPz5toYKbOIzWlIJOMcpy
s4IPKxF1T4HZJIN304GdHdB6RoFUGI58aoEVW75jLYohVCpla4j4TNdntYfVRc1qKobUPvNssOxt
OFBvTIUjMapx/4RpKwZ0vIO2JI53XoL9He556YlyGL58IIGBWiyAs1BJUuGAZ1QTS8SAJVfrRK6A
ElvIPbGAH1/sNs6AUWQEL9SAvGgrKMyBWGxrle2CAnDDGuaCxnVOFUqDiXnvWXV9IFZPUkV
9LVmgS0R9RVz/REI9h2CBPWZ94WQRNp5+WWeYL75/AWtjKNN/0W8rlg6ArnMGgx+Bo3cHFZ
uCjnsSWNh581wjUb56T17/StR6tGHoQ+97O2T2PSN72WgRNnJ8kms7L6t9cm52KNh+dHHElit/g
XUhG1CAo3ieFeCBsnwZWDI3JWHNUTh3x2Q5SmV4cGazQ5V5NGITPN96C2wANKd6+W65L5h
8DHF3KNx9PnQ+IkR+d3cVG3t/wnoRFhyA9H0EV6V0pWdaULd1fmlaSfd2X2tnQzp3W22WPJl4a
G/RNhh5iHIXL4F6ynRZKN58J3aell99h3jzG69+9XtwFmmAQX3cVyJyaWzNUDZzcm5mSX10gnA
GQtJ1snG8PEd28HN+Nd94QHVNL2N5rXcaKN57MHjrloB8r3rHG/p+NXzAFtx/kH6kVqFwYXJKT7
5xI3NiSQ9y1nSDQnV0NHW8O/11oXcDNax3HHhZL0h4sHmyKN16VnsRlp578nx3HD59jX3wF0N
++H9TVhxuknfHT0Jv73hXSJ1xVHjxQhRy2XmuO7B0bXp7NXV2D3tYLyp3yHw8KNp5jn0mlrh7SH
4THHd8+38HF5t+dn/qVZltBH0jTr5uen08SB5v+n1eQadxnH2iO1IzTn33NTd1EX5cLwZ27H7IKNJ
41H84Isx6rX+kHKd8e4ALF+R+C4BqVRhrqoJuTkBtMolQR6ZuxYG5QUFwglGBowdyTYFbNPx0
LYFILuR2J4E6KMp4L4Eultx6JIEZHM98DYD1GCF9sIDXVJ5qgoeSTdBsGYazR0FtvIXdQOtjYU
nOsNxcYSFNMpzaoP5Lsd1gINzKMN3pILulup5soJgHPB7sYG+GFR9ZYEzUimGaFCcS82FeFR8
RZqEnFhnP1KED1xqORqDqWB5MvWDc2SYLLODeGjHJl6Dr20NICqD9XFpGcKEU3X1FLiEuHq
KUY+DD1ZIS0mCdVm6RSyB7l03Pv6BrmDOOOCBj2RvMtKBmGgdLKmB02vbJm6COG+sIFOC
q3ONGgSDNXeSFRGDvnuSUQ6ADivLStt/v17TRNJ/gmHIPrV/h2UNOKp/qWhAMrJ/62t+LKCaw
G7JJnyA6XIIIHeBhnWLGjyCOXkMFV6C43x+Ult9ZWErSmh9XGPORG99Y2Z5PmV9o2k7OG99+
WwGMO9+aW7ZLJR/A3GqJod/v3SAIJeAhHdhGm6BXXpkFaCCJH1QUA17DGZxSfh7OWiwRA5
7d2r2PhV77G1NODJ8dG+sMml9EXIRLIV903RqJpB+tXa/ILh/nHkhGqiAk3umFfKBcX4QT5949Gu
gSYt5UW19Q6Z5v29cPbt6ZnE/N+t7HXMOMjd75XUZLG18zHcCJpV9yXjolN9+yXrVGvp/03zZFm

uAvn69TzF3EXDXSSV3m3I5Q0I4NXOePWt5CXUJN6t563Z8Mgt63Hf3LFd743lwJpp8+3rqIQN+E
HxmG0N/LH3sFteAln9UTsB1aXYOSLI2F3buQuN21nfUPRN3z3jHN2R41XnEMdh56XrKLDx7D3v
SJpp8QHHzclR99a33iG4F+m37oFzJ/n3/VTk1z/3slSEF0x3uYQmt1oHwNPKx2s3yLNw931n0TMZp
5B32ILBh6SH44JpN7k37LITV8139UG7N+HX/SF35/MoBCTdlyylAyR8tzo4A0Qfh0joA3PEI1t4A9N
r928YBOMV54PIBqK/R5loCGJop6+oCdIUZ8VYCqG919sICjF79+1oCfTWpxwYUhR2NygYSkQZh
zooQqO/Z034O3Nnt2LoNQMSx3j4L1K9d5AIKbJoN6eYI9IVV76IHVHAB9VoFWF/R+iYDtSkOMK1
DSRN2KjITSP5eJEFjXOiGH/1zoNLCHIWEEL0iGfGUtKb2GFmlmJBmF422zHoqFwnIQGLmFvHa
ZFDaFxXsaSceJC1YoRG+Hw1nEPzqGI11nOd2FymEXNISFK2TNLzGEvmiKKb6EhmxWJDKEen
AyHryEgHQXGQOEoXgZFJaEynwHSWeGLVtfRB6FL16bPveETWHcOaWDx2UkNFuDZmhxLxy
DLWvFKb6DIG8iJEiDOXKLHueDY3X3GUODqHI4FOqD73zbSQSDpmB6Q76C8GNZPp+CU2Y8
OV6CBmkINCmB1GwRLwGBwW7/KbqB23HIJFqCGnTIHwyCaHexGXqC0Hq2FTKDMn2WSJyBc
WV+Q1+A8WgBPkqAiWqGORiAcG0KM/WAbG+PLuKAf3ITKbGAvHSEJGmBG3bpHzOBhnIVGb
qCCHveFYiCfn4/SEd/d2puQwl/KmyUPfV+9G63OM1/C3DJM7d/NHLCld/cHTvKZ5/yXb0JHSAO3
jvH2GAtnrrGhKBR3z0FgaBx37UR/B9t29IQrd9IXEVPaI9i3LCOlp9znRgM4B+IHX/LpB+gnegKY1+
9Hk6JH5/dnrPH4qAAXxeGmGAoH3tFneBKX9UR5F8NHRbQlp8NnWPPU98TnbCODp8tHfsMz99
KHkZLmB9qXpIKXR+M3t1JIF+x3yfH6p/Yn29GqKAEH7QFtWApH/CRy1673kyQet7C3oAPNt7PXr
KN9V7u3uCMut8SHw+LiB843z7KU59gn22JHl+J35qH8B+1H8PGtd/IH+hFyWANIAfRsF5234EQX
p6Cn5nPGt6Tn7EN3J64H8JMpl7gn9QLeF8MX+ZKSh85H/eJG99m4AaH9F+WYBHGWJ/KoBbF2
d/14BtRIV48oK/QRR5L4KmPAx5goKHNYb6JIJXMIR62IlqLa17m4IAKQh8XoHSJGd9JYGbH9998
oFWGyZ+0YD6F55/ioCuQiOR/FEBPbWP8FUPOWKN+1kiNNKMRI1NMDqK02F9K5uJp2W4JtGI
wmn/IeOIFm5WHPuHhXK1F76HE3c5E76GvXunQcePHIYGPV+NYFm4ORiLtV1xNKGKOmFGM
B6I/mUZK5KIBWjuJt2HRmzMlgaGunC1HTWGSXSfGA+F+3ieFCSFxHx5QaGMVvr4PTeK5I5QO
O+JiWGuNHuIWWUhMAOHwGiSK4iGjGwCJuaF8m92liSFhHLvHWeFMnZiGFaFBXniFH2E6X00
QVaJ+F/NPOmlzmLROKKHsWXbND6GuGj2L9qF4mwMK3aFNG8cJuqEu3IbljyEcHUPHZKEPng
AGJSEL3sFFMuEK33ZQQSH4mSTPJyG8GdAOFuGCWnxNASFQmysL6+Elm9hK1+ECHIMJum
DrHSYIIKDencRHb+DYHmHGNmDaXwTFSaDdn5rQOOF92IGPGiFQWuaOBuEk23tM8iD/HA6L3
2De3J/KzuDFHS7JtyCzHbhlmOCpHj1HfOCiHr/GTeCp30PFamCvX7pQLOEVm4APDODym/hN+
ODR3HCM5OC23OdL1CCgnVyKxqCPnc/JtCCXkAlnOB6XqzHiGB4XxWGYuB/33uFh2CHH9V
QGiC/nK4O+aCkHQiN5WCK3WMM02B4Hb3LxSBp3hdKu2Bf3m8JriBW3sUlniBQ3xgHkSBRX2X
GdCBb364Fn+BIH+xQAeB53dRO3uBj3hbNyaBP3liMumBCnpgLr+A53tXKQuA1HxGJpCAv30ulm
+AsX4IHluAvn7LGgeA9X9yFtGBI3/+P5WA+nvpOwmAtHyMNreAdX0rMoeAUH26LmuAPn5AKm
mAPX6+JmeANH8ylmSAMX+XHmuASn/mGjSAjYAWFxaAxA/PyGAL4B0OqB/+ICbNII/yIDAMjR
/sYDdLiV/r4DzKjF/vYEBJkV/wIEEIIP/xoD4HnI/6IDbGIqANoCiF0+AdoB1AAD//wAA//8AAP//AABtZ
nQxAAAAAAMEIQAAAQAAABAAAAAQI
DBAUGBwgJCgsMDQ4PEBESExQVFhcYGRobHB0eHyAhliMkJSYnKCkqKywtLi8wMTIzNDU2Nz
g5Ojs8PT4/QEFCQ0RFRkdISUpLTE1OT1BRUINUUVZXWFlaw1xdXI9gYWJjZGVmZ2hpamtsbW
5vcHFyc3R1dnd4eXp7fH1+f4CBgoOEhYaHilmKi4yNjo+QkZKTlJWWl5iZmpucnZ6foKGio6Slpqeo
qaqrrK2ur7CxsR00tba3uLm6u7y9vr/AwcLDxMXGx8jJysvMzc7P0NHS09TV1tfY2drb3N3e3+Dh4u
Pk5ebn6Onq6+zt7u/w8fLz9PX29/j5+vv8/f7/AAEBAGlDAwQEBQYGBwclCAkJCgsLDAwNDQ4PD
xAQERESExMUFBuWFhcXGBkZGhobHBwdHh4fICAhliljJCQJiYnKCkqKissLS0uLzAxMjlzNDU2
Nzg5Ojs8PT4/QEJDREVGSEIKTE1PUFJTVVdYWlxeYgJkZmhqbW9xdHZ5fH6Bg4aJi46QkpWX
mZudn6Gjpaeoqqytr7Cys7W2t7m6u7y9v8DBwsPExcBHyMnKy8zNzc7P0NHS0tPU1dbW19jZ2dr

b29zd3d7f3+Dh4eLj4+Tl5ebm5+jo6enq6+vs7O3u7u/v8PDx8vLz8/T09fb29/f4+Pn5+vv7/Pz9/f7+/w
ABAQICAwMEBAUGBgcHCAgJCQoLCwwMDQ0ODw8QEBEREhMTFBQVfYhYXFxgZGRoaGxw
cHR4eHyAgISlilyQkJSYmJygpKSorLC0tLi8wMTlyMzQ1Njc4OT07PD0+P0BCQ0RFRkhJSkxNT1
BSU1VXWFpcXmBiZGZoam1vcXR2eXx+gYOGiYuOkJKVl5mbnZ+ho6WnqKqsra+wsrO1tre5uru8
vb/AwcLDxMXGx8jJysvMzc3Oz9DR0tLT1NXW1tfY2dna29vc3d3e39/g4eHi4+Pk5eXm5ufo6Onp6
uvr7Ozt7u7v7/Dw8fLy8/P09PX29vf3+Pj5+fr7+/z8/f3+/v/2ukgV9rxOK/i7VEb5uGBI+bRqgPS3a5Pr
uW6k5Lpxs926dcDXuH7M0rWH1c+zjd3LsZHjya+V6caume/DrZ31wK2i+sCtovrAraL6wK2i+sCtovr
AraL6wK2i+sCtovrAraL6wK2i+sCtovrAraL6wK2i+sCtovrAraL6wK2i+sCtovr2ukgV9rxOK/i7VEb5u
GBI+bRqgPS3a5PruW6k5Lpxs926dcDXuH7M0rWH1c+zjd3LsZHjya+V6caume/DrZ31wK2i+sCtov
rAraL6wK2i+sCtovrAraL6wK2i+sCtovrAraL6wK2i+sCtovrAraL6wK2i+sCtovrAraL6wK2i+sCtovr2uk
gV9rxOK/i7VEb5uGBI+bRqgPS3a5PruW6k5Lpxs926dcDXuH7M0rWH1c+zjd3LsZHjya+V6caume/
DrZ31wK2i+sCtovrAraL6wK2i+sCtovrAraL6wK2i+sCtovrAraL6wK2i+sCtovrAraL6wK2i+sCtovrAra
L6wK2i+sCtovr2ukgV9rxOK/i7VEb5uGBI+bRqgPS3a5PruW6k5Lpxs926dcDXuH7M0rWH1c+zjd3
LsZHjya+V6caume/DrZ31wK2i+sCtovrAraL6wK2i+sCtovrAraL6wK2i+sCtovrAraL6wK2i+sCtovrAr
aL6wK2i+sCtovrAraL6wK2i+sCtovr2ukgV9rxOK/i7VEb5uGBI+bRqgPS3a5PruW6k5Lpxs926dcDX
uH7M0rWH1c+zjd3LsZHjya+V6caume/DrZ31wK2i+sCtovrAraL6wK2i+sCtovrAraL6wK2i+sCtovrAr
aL6wK2i+sCtovrAraL6wK2i+sCtovrAraL6wK2i+sCtovr2ukgV9rxOK/i7VEb5uGBI+bRqgPS3a5Pru
W6k5Lpxs926dcDXuH7M0rWH1c+zjd3LsZHjya+V6caume/DrZ31wK2i+sCtovrAraL6wK2i+sCtovr
AraL6wK2i+sCtovrAraL6wK2i+sCtovrAraL6wK2i+sCtovrAraL6wK2i+sCtovr2ukgV9rxOK/i7VEb5u
GBI+bRqgPS3a5PruW6k5Lpxs926dcDXuH7M0rWH1c+zjd3LsZHjya+V6caume/DrZ31wK2i+sCtov
rAraL6wK2i+sCtovrAraL6wK2i+sCtovrAraL6wK2i+sCtovrAraL6wK2i+sCtovrAraL6wK2i+sCtovr2uk
gV9rxOK/i7VEb5uGBI+bRqgPS3a5PruW6k5Lpxs926dcDXuH7M0rWH1c+zjd3LsZHjya+V6caume/
DrZ31wK2i+sCtovrAraL6wK2i+sCtovrAraL6wK2i+sCtovrAraL6wK2i+sCtovrAraL6wK2i+sCtovrAra
L6wK2i+sCtovr2ukgV9rxOK/i7VEb5uGBI+bRqgPS3a5PruW6k5Lpxs926dcDXuH7M0rWH1c+zjd3
LsZHjya+V6caume/DrZ31wK2i+sCtovrAraL6wK2i+sCtovrAraL6wK2i+sCtovrAraL6wK2i+sCtovrAr
aL6wK2i+sCtovrAraL6wK2i+sCtovr2ukgV9rxOK/i7VEb5uGBI+bRqgPS3a5PruW6k5Lpxs926dcDX
uH7M0rWH1c+zjd3LsZHjya+V6caume/DrZ31wK2i+sCtovrAraL6wK2i+sCtovrAraL6wK2i+sCtovrAr
aL6wK2i+sCtovrAraL6wK2i+sCtovrAraL6wK2i+sCtovr2ukgV9rxOK/i7VEb5uGBI+bRqgPS3a5Pru
W6k5Lpxs926dcDXuH7M0rWH1c+zjd3LsZHjya+V6caume/DrZ31wK2i+sCtovrAraL6wK2i+sCtovr
AraL6wK2i+sCtovrAraL6wK2i+sCtovrAraL6wK2i+sCtovrAraL6wK2i+sCtovr2ukgV9rxOK/i7VEb5u
GBI+bRqgPS3a5PruW6k5Lpxs926dcDXuH7M0rWH1c+zjd3LsZHjya+V6caume/DrZ31wK2i+sCtov
rAraL6wK2i+sCtovrAraL6wK2i+sCtovrAraL6wK2i+sCtovrAraL6wK2i+sCtovrAraL6wK2i+sCtovr2uk
gV9rxOK/i7VEb5uGBI+bRqgPS3a5PruW6k5Lpxs926dcDXuH7M0rWH1c+zjd3LsZHjya+V6caume/
DrZ31wK2i+sCtovrAraL6wK2i+sCtovrAraL6wK2i+sCtovrAraL6wK2i+sCtovrAraL6wK2i+sCtovrAra
L6wK2i+sCtovr2ukgV9rxOK/i7VEb5uGBI+bRqgPS3a5PruW6k5Lpxs926dcDXuH7M0rWH1c+zjd3
LsZHjya+V6caume/DrZ31wK2i+sCtovrAraL6wK2i+sCtovrAraL6wK2i+sCtovrAraL6wK2i+sCtovrAr
aL6wK2i+sCtovrAraL6wK2i+sCtovr2ukgV9rxOK/i7VEb5uGBI+bRqgPS3a5PruW6k5Lpxs926dcDX
uH7M0rWH1c+zjd3LsZHjya+V6caume/DrZ31wK2i+sCtovrAraL6wK2i+sCtovrAraL6wK2i+sCtovrAr
aL6wK2i+sCtovrAraL6wK2i+sCtovrAraL6wK2i+sCtovr2ukgV9rxOK/i7VEb5uGBI+bRqgPS3a5Pru
W6k5Lpxs926dcDXuH7M0rWH1c+zjd3LsZHjya+V6caume/DrZ31wK2i+sCtovrAraL6wK2i+sCtovr
AraL6wK2i+sCtovrAraL6wK2i+sCtovrAraL6wK2i+sCtovrAraL6wK2i+sCtovr2ukgV9rxOK/i7VEb5u
GBI+bRqgPS3a5PruW6k5Lpxs926dcDXuH7M0rWH1c+zjd3LsZHjya+V6caume/DrZ31wK2i+sCtov

rAraL6wK2i+sCtovrAraL6wK2i+sCtovrAraL6wK2i+sCtovrAraL6wK2i+sCtovrAraL6wK2i+sCtovr2u0
gU9bxOKve8VEb5uF9k+LZof/S5apHsu2yi5b1usd6/cb7ZvnnJ1LyD0tG6iNnOuY3ey7iR48e1IOvCs
5fzuq+b+Lqvm/i6r5v4uq+b+Lqvm/i6r5v4uq+b+Lqvm/i6r5v4uq+b+Lqvm/i6r5v4uq+b+Lqvm/i6r5v4u
q+b+Lqvm/j2u0cU9bxOKve8VEb5uV9k+LdnfvO7aJDtmvg5sFrruDEbrbxbxPF18d8zNHDg9bMvof
ex7qM5cK2kuy+s5rxs7GV9rOxlfazsZX2s7GV9rOxlfazsZX2s7GV9rOxlfazsZX2s7GV9rOxlfazsZX2
s7GV9rOxlfazsZX2s7GV9rOxlf1u0cU9b1OKve8VEb4uV5k97hlffK9Zo7uwWee6MZpq+LMa7bZz
nLA0cd8zsvCg9jGvYjfwrqO5b+3lOu9tJvvrrKR866ykfOuspHzrrKR866ykfOuspHzrrKR866ykfOusp
HzrrKR866ykfOuspHzrrKR866ykfOuspHzrrKR866ykfP1u0cU9b1NKva9U0X4ul5k9rpje/G/ZlrxWS
b5cxmp9zVZ7LTznHDy8Z80MbBhNnCvYrfv7uQ5L24lum4tpftqrSO8aq0jvGqtl7xqrSO8aq0jvGqtl7
xqrSO8aq0jvGqtl7xqrSO8aq0jvGqtl7xqrSO8aq0jvGqtl7xqrSO8aq0jvH1vEYU9L5NKva9U0X3ul1k
9Lxheu/DYYnoymKX39VkodbWZbTNzXLExsZ+0MLBhti+vozevLyS47u5l+ezt5Lrp7WM7qe1jO6nt
Yzup7WM7qe1jO6ntYzup7WM7qe1jO6ntYzup7WM7qe1jO6ntYzup7WM7qe1jO6ntYzup7WM7qe1jO6nt
1jO70vEYT9L5NKfW+U0X3u11k879fd+zHXoXi0V+R2t1Zo9DVZrbHzHPFwsaA0L7CiNe8v47cur2
T4Li7luStuY3opLeJ6qS3ieqkt4nqpLeJ6qS3ieqkt4nqpLeJ6qS3ieqkt4nqpLeJ6qS3ieqkt4nqpLeJ6q
S3ieqkt4nqpLeJ6qS3ier0vUUT879MKfS+UkT2vFxk8MJcdOfMW4Dc2liO099YpcrUZ7fDzHXFvse
Bz7zDitW6wl/aub6U3rK8kOGpu4rkobmH56G5h+ehuYfnobmH56G5h+ehuYfnobmH56G5h+ehuYf
nobmH56G5h+ehuYfnobmH56G5h+ehuYfnobmH56G5h+fzvUUT8r9LKfS/UkT2vFtj7cZZcOHTV3n
V4FSRzd5ap8TUabi/zHjFu8iDzbnEi9O4wpDXtsCS262+i96lvYfhnruf4567heOeu4XjnruF4567heO
eu4XjnruF4567heOeu4XjnruF4567heOeu4XjnruF4567heOeu4XjnruF4567hePzvkQS8sBKKPPAU
UTzwFhg58xVadndT3fP5VWTx91cqL/Ubli7zXrDuMmFy7fGjNC2xJHUr8KM16jAh9qhvtcm76E3
pu+hN6bvoTem76E3pu+hN6bvoTem76E3pu+hN6bvoTem76E3pu+hN6bvoTem76E3pu+hN6bvo
Tem76E3pu+hN7yv0MS8MFJJ/HCUPEvxVNa3tVPX9LkT3vl6FaVwN1fqLrVb7a3z3zAtcuGx7TIjM
yuxonQqcWG06PDg9WdwLXI8GC2ZfBgtxwYLZI8GC2ZfBgtxwYLZI8GC2ZfBgtxwYLZI8GC2ZfBgtxwYLZI8G
C2ZfBgtxwYLZI8GC2ZfBgtxwYLZI8GC2ZfBgtnwwEER78NIJu/ETklmzU1Q1OFJYcrrUX7B51
mWut5jp7bWcrOz0n67sM6DwqrLgcekyX/KoMh+zJvHfs+Xxn7QIMV/OpTFf9KUxx/SIMV/OpTFf9K
Uxx/SIMV/OpTFf9KUxx/SIMV/OpTFf9KUxx/SIMV/OpTFf9KUxx/SIMV/OpTFf9LUwj8Q7MVfJezGT
EDY3EFFzOpLZclxVIC66F6VtOBopLHada6s1Xq2pdJ4u5/Pd7+aznbClsx3xJLLd8aOy3jli8p5yYv
KecmLynnJi8p5yYvKecmLynnJi8p5yYvKecmLynnJi8p5yYvKecmLynnJi8p5yYvKecmLynnJi8p5yY
vKecnrxTsP6clCI+DSQTPO50JLw/JOabr0WYC06mORr+Ntnqjdckeg2m+umdZvs5TUb7aP02+5j
NJwu4nRcbYG0HK+g9B0v4PQdL+D0HS/g9B0v4PQdL+D0HS/g9B0v4PQdL+D0HS/g9B0v4PQdL
+D0HS/g9B0v4PQdL+D0HS/g9B0v4PQdL/nyTYN5M48INHjOjLF8UZPu/ITabT2Xnyu7WiLpedol5z
iZp+U32aljdxdnqonbaKyF2WmugtlqsIDYa7F9122ye9Zvs3vWb7N71m+ze9Zvs3vWb7N71m+ze9Zv
s3vWb7N71m+ze9Zvs3vWb7N71m+ze9Zvs3vWb7N71m+ze9Zvs3vWb7Phzy0K1t0tG8fwPDe8+k
pRtP5YZq75Y3ej8mGEemxfjo/nX5al5F+bg+Jhn3/hY6F84GSjed9lpHffZ6V13mindN1qp3Tdaqd03
WqndN1qp3Tdaqd03WqndN1qp3Tdaqd03WqndN1qp3Tdaqd03WqndN1qp3Tdaqd03WqndN1qp3Tdaqd03WqndN1qp3
TdaqfY2R4Kyuwzlb35QTm0/1BPrv9dYaL9W2+X91I6jfFYhIXtWYt+6lqReOhclHXnXpdy5mCZceVi
mm/kZJtt5GacbONonWzjaJ1s42idbONonWzjaJ1s42idbONonWzjaJ1s42idbONonWzjaJ1s42idbO
NonWzjaJ1s42idbONonWzjaJ3+sj4R/LVOJf21Vj7+smJa/bBrvcqyboPzs3KT7LN3ouaxfk/gr4O626
2MxNerIMzTqprT0aad2M+endZNlp/gzJGn48qSsuPClrPjwpaz48KWws+PClrPjwpaz48KWws+PClrPj
wpaz48KWws+PClrPjwpaz48KWws+PClrPjwpaz48KWws+P+sj4R/LVOJf21Vj7+smJa/bBrvcqyboPzs3
KT7LN3ouaxfk/gr4O6262MxNerIMzTqprT0aad2M+endZNlp/gzJGn48qSsuPClrPjwpaz48KWws+PC
lrPjwpaz48KWws+PClrPjwpaz48KWws+PClrPjwpaz48KWws+PClrPjwpaz48KWws+P+sj4R/LVOJf21Vj7

+smJa/bBrcvqyboPzs3KT7LN3ouaxfK/gr4O6262MxNerIMzTqprT0aad2M+endzNlp/gzJGn48qSsu
PClrPjwpaz48KWs+PClrPjwpaz48KWs+PClrPjwpaz48KWs+PClrPjwpaz48KWs+PClrPjwpaz48K
Ws+P+sj4R/LVOJf21Vj7+smJa/bBrcvqyboPzs3KT7LN3ouaxfK/gr4O6262MxNerIMzTqprT0aad2M
+endzNlp/gzJGn48qSsuPClrPjwpaz48KWs+PClrPjwpaz48KWs+PClrPjwpaz48KWs+PClrPjwpaz4
8KWs+PClrPjwpaz48KWs+P+sj4R/LVOJf21Vj7+smJa/bBrcvqyboPzs3KT7LN3ouaxfK/gr4O6262
MxNerIMzTqprT0aad2M+endzNlp/gzJGn48qSsuPClrPjwpaz48KWs+PClrPjwpaz48KWs+PClrPjw
paz48KWs+PClrPjwpaz48KWs+PClrPjwpaz48KWs+P+sj4R/LVOJf21Vj7+smJa/bBrcvqyboPzs3K
T7LN3ouaxfK/gr4O6262MxNerIMzTqprT0aad2M+endzNlp/gzJGn48qSsuPClrPjwpaz48KWs+PClr
Pjwpaz48KWs+PClrPjwpaz48KWs+PClrPjwpaz48KWs+PClrPjwpaz48KWs+P+sj4R/LVOJf21Vj7+
smJa/bBrcvqyboPzs3KT7LN3ouaxfK/gr4O6262MxNerIMzTqprT0aad2M+endzNlp/gzJGn48qSsuP
ClrPjwpaz48KWs+PClrPjwpaz48KWs+PClrPjwpaz48KWs+PClrPjwpaz48KWs+PClrPjwpaz48KW
s+P+sj4R/LVOJf21Vj7+smJa/bBrcvqyboPzs3KT7LN3ouaxfK/gr4O6262MxNerIMzTqprT0aad2M+e
ndzNlp/gzJGn48qSsuPClrPjwpaz48KWs+PClrPjwpaz48KWs+PClrPjwpaz48KWs+PClrPjwpaz48K
Ws+PClrPjwpaz48KWs+P+sj4R/LVOJf21Vj7+smJa/bBrcvqyboPzs3KT7LN3ouaxfK/gr4O6262Mx
NerIMzTqprT0aad2M+endzNlp/gzJGn48qSsuPClrPjwpaz48KWs+PClrPjwpaz48KWs+PClrPjwpaz
48KWs+PClrPjwpaz48KWs+PClrPjwpaz48KWs+P+sj4R/LVOJf21Vj7+smJa/bBrcvqyboPzs3KT7L
N3ouaxfK/gr4O6262MxNerIMzTqprT0aad2M+endzNlp/gzJGn48qSsuPClrPjwpaz48KWs+PClrPjw
paz48KWs+PClrPjwpaz48KWs+PClrPjwpaz48KWs+PClrPjwpaz48KWs+P+sj4R/LVOJf21Vj7+sm
Ja/bBrcvqyboPzs3KT7LN3ouaxfK/gr4O6262MxNerIMzTqprT0aad2M+endzNlp/gzJGn48qSsuPClr
Pjwpaz48KWs+PClrPjwpaz48KWs+PClrPjwpaz48KWs+PClrPjwpaz48KWs+PClrPjwpaz48KWs+
P+sj4R/LVOJf21Vj7+smJa/bBrcvqyboPzs3KT7LN3ouaxfK/gr4O6262MxNerIMzTqprT0aad2M+en
dzNlp/gzJGn48qSsuPClrPjwpaz48KWs+PClrPjwpaz48KWs+PClrPjwpaz48KWs+PClrPjwpaz48K
Ws+PClrPjwpaz48KWs+P+sj4R/LVOJf21Vj7+smJa/bBrcvqyboPzs3KT7LN3ouaxfK/gr4O6262Mx
NerIMzTqprT0aad2M+endzNlp/gzJGn48qSsuPClrPjwpaz48KWs+PClrPjwpaz48KWs+PClrPjwpaz
48KWs+PClrPjwpaz48KWs+PClrPjwpaz48KWs+P9sj4R+7ZOJf22Vj7+smFa/LFqcfmzbIL0tXCS7b
V0oea0ea3hs3+43LGIwtiukMvUrZfS0aqb18+IndzMm57hy5Om5cSVseW9I7HlvZex5b2XseW9I7Hl
vZex5b2XseW9I7HlvZex5b2XseW9I7HlvZex5b2XseW9I7HlvZex5b2XseX9sj4Q+7ZOJfy2Vj79s2F
a/LJocPm1a4D1tm6Q7rdyn+e3d6vit3u23bWEwNmzjMjVsZPP0q6Z1c+rndvMop7hypWm576YsO
e2mLDotpiw6LaYsOi2mLDotpiw6LaYsOi2mLDotpiw6LaYsOi2mLDotpiw6LaYsOi2mLDotpiw6LaY
sOj9sz4Q+7ZPJfy2Vj79s2Fa+7Nnb/i2aX/1uGyP77pwnem7dKnjvHm037t/vdu5iMXYuJDM1LaX0t
Gzm9fNq53gxJmi6LSYruqvma7qr5mu6q+Zruqvma7qr5mu6q+Zruqvma7qr5mu6q+Zruqvma7qr5
mu6q+Zruqvma7qr5mu6q+Zrur8sz8Q+rdPJfy3Vj79s2Fa+7Rmbve3aH70umuN8L1um+q/cqblwXe
w4cN8ud3EhcDbxI7G1sKUzdK8ltbNtZLftaKY6aecrO2lm63spZut7KWbreylm63spZut7KWbreylm6
3spZut7KWbreylm63spZut7KWbreylm63spZut7KWbrez8sz8Q+rdPJPY3VT79tGBa+rVlbf5Znzzv
WmL78BsmOvDcKPnyHws5M17s97RhbjZzozC0siRzMzDk9XCvo/eqq2Q6JimpO6aoqzmqKs7Z
qirO2aoqzmqKs7ZqirO2aoqzmqKs7ZqirO2aoqzmqKs7ZqirO2aoqzmqKs7ZqirO38tD8Q+rdPJP
u3VT39tGBa+rZkbPW7ZXvxv2eJ7cRqlerJbp/lz3Sn39h9q9nVh7TRz43By8mRzMXEINS7vpPdorm
I55C2nuyTr63rk6+t65OvreuTr63rk6+t65OvreuTr63rk6+t65OvreuTr63rk6+t65OvreuTr63rk6+t65Ov
rev8tD8Q+bhPJPu4VT38tGBa+bhia/S9Y3nwwmWG68hokuXPbZrf2nWf2dp7rNHUhLfLzozBxcmR
y8DEIdS4v5XcoLqH5ZO4luiTuafok7mn6JO5p+iTuafok7mn6JO5p+iTuafok7mn6JO5p+iTuafok7m
n6JO5p+iTuafok7mn6JO5p+j7tD8Q+bhPJPu4VT38tV9Z+LihafO/YXftxmOD5s1njeDYbZPa326i0tl
4r8vSgbvFzInFwMiPzbvEk9Szv5HcnruH45a6kuWXu6Dkl7ug5Je7oOSXu6Dkl7ug5Je7oOSXu6Dk

l7ug5Je7oOSXu6Dkl7ug5Je7oOSXu6Dkl7ug5Je7oOT7tT8Q+bhOJPq4VT38tV9Z97tfZ/HCX3Tpy
mF+4dRIhtrfZpTT322ly9d3ssXQgL2/y4jHu8eOzrjDktWswlzbLnL2F4JS8jeKVvJrilbya4pW8muKVvJ
rilbya4pW8muKVvJrilbya4pW8muKVvJrilbya4pW8muKVvJrilbya4pW8muL6tUAP+LlOI/q5VD37tl1
Y9b5cZe7GXHDI0F95291gg9TiZJjN3WyoxdV2tb/PgL+7y4jluMeOz7TEkdSnwYjZmr6E3pO9iuCS
vZTfkr2U35K9IN+SvZTfkr2U35K9IN+SvZTfkr2U35K9IN+SvZTfkr2U35K9IN+SvZTfkr2U35K9IN/6tk
AP+LlOI/m5VDz6uFtW88FaYenLWmve2F1w1eJdh87lY5vG3Gyqv9V3trvPgMC3y4jltceOzq7Ei9Oj
woXXmMCD25K/h9yQv4/ckL+P3JC/j9yQv4/ckL+P3JC/j9yQv4/ckL+P3JC/j9yQv4/ckL+P3JC/j9yQ
v4/ckL+P3JC/j9z5t0AP97pNI/i6Uzz4u1hT7sVXXOLSV2PW4FdzzuddisfjY53A222sutR4t7bPgcC0
y4nHrsiHzKbGhdCexlLUlsKB15DBhdiOwYrZjsGK2Y7BitmOwYrZjsGK2Y7BitmOwYrZjsGK2Y7Bit
mOwYrZjsGK2Y7BitmOwYrZjsGK2Y7Bitn4uEEO9rtNIve7Uzv0v1VO6MtTVNndUF3P5lZ3x+xdjsDi
ZJ+522+stdR6trPQg76szlLEpcqAyZ/lf82Zxn7QIMWA0o7Eg9OMxlfUjMSH1lzEh9SMxlfUjMSH1lz
Eh9SMxlfUjMSH1lzEh9SMxlfUjMSH1lzEh9SMxlfUjMSH1lzEh9T2uUIO9bxMlfa8UjvuxFFH3tZMS
NHIUGLI7Vd7wOtekLniZqC023GrsdZ8tKrSfLujz3rBnMx5xZfLeciSyXrKjsh8zlrHf82lx4LOiMeCzojH
gs6lx4LOiMeCzojHgs6lx4LOiMeCzojHgs6lx4LOiMeCzojHgs6lx4LOiMeCzojHgs71u0MN9L1KIPS
+UDrlzko80+JHSsntUWbA81l+uepikLPjap6v3Xspp9h1saDUdLeZ0nO7k9Bzvo/PdMGLzXXDh8x3
xlTMesaCy33Ggst9xoLLfcaCy33Ggst9xoLLfcaCy33Ggst9xoLLfcaCy33Ggst9xoLLfcaCy33Ggst9x
oLLfcbzvUEM8cBIH+zGSTDW3j4yyuxlT8D1U2q49V1+sutmjq7kbpul322knNtsq5XZbLCP1m20itR
utofTb7iD0nG6gNJzu37Rdbx80Xe9fNF3vXzRd7180Xe9fNF3vXzRd7180Xe9fNF3vXzRd7180Xe9f
NF3vXzRd7180Xe9fNF3vXzRd73wwD4L7sNEHd3VOSDM6kA4wfVLU7j7V2qy9mF8rO5piqLoZp
WY42SekOBPlrdZqiF3GirgdpprX7Zaq582WywedhusXfYcLJ213Kztdtdys3bXcrN213Kztdtdys3bXcr
N213Kztdtdys3bXcrN213Kztdtdys3bXcrN213Kztdtdys3bXcrPsxDgJ58o8F8/mNSPC9EI9ufxQVbH/X
Giq+GN4n/FghJXsXo6M6F6VheVfmoDjYZ584WOgeeBkonbgZqN032ekct9ppXHea6Zv3m2nb95t
p2/ebadv3m2nb95tp2/ebadv3m2nb95tp2/ebadv3m2nb95tp2/ebadv3m2nb95tp2/ebafmyjAH0+AnE
cTyOii5/Ec/sf9VU6n/XGOe/Flxk/ZYflnxWISB7lmKeutaj3XqW5Jy6F2UcOdflm7nYZds5mOYa+Zkm
WrlZplp5WiaaeVommnlaJpp5WiaaeVommnlaJpp5WiaaeVommnlaJpp5WiaaeVommnlaJpp5Wiaae
VommnlaJra1xkExu8vFbv8PSqy/009qP9TTPz/UlyR/1Foh/xRcn73Unl39FR/cfJWg23wWlZq71qla
O5cimbuXYtl7V+MY+1hjWLsY41h7GSOYexkjmHsZI5h7GSOYexkjmHsZI5h7GSOYexkjmHsZI5h
7GSOYexkjmHsZI5h7GSOYexkjmHsZI7/qjUN/6xEIP+tTzf/qlxQ/6pnZv6tb3b6rnOF8615lO2rf6Hnql
Ws4qSMtt6dkb7blpbF2JCaytaLn87ViKXQ1ler0dSHs9LTiL3Syo290ciOvdHljr3Ryl690ciOvdHljr3R
yl690ciOvdHljr3Ryl690ciOvdHljr3Ryl690ciOvdH/qjUN/6xEIP+tTzf/qlxQ/6pnZv6tb3b6rnOF8615lO
2rf6HnqlWs4qSMtt6dkb7blpbF2JCaytaLn87ViKXQ1ler0dSHs9LTiL3Syo290ciOvdHljr3Ryl690ciOv
dHljr3Ryl690ciOvdHljr3Ryl690ciOvdHljr3Ryl690ciOvdH/qjUN/6xEIP+tTzf/qlxQ/6pnZv6tb3b6rnOF
8615lO2rf6HnqlWs4qSMtt6dkb7blpbF2JCaytaLn87ViKXQ1ler0dSHs9LTiL3Syo290ciOvdHljr3Ryl
690ciOvdHljr3Ryl690ciOvdHljr3Ryl690ciOvdHljr3Ryl690ciOvdH/qjUN/6xEIP+tTzf/qlxQ/6pnZv6tb3
b6rnOF8615lO2rf6HnqlWs4qSMtt6dkb7blpbF2JCaytaLn87ViKXQ1ler0dSHs9LTiL3Syo290ciOvd
Hljr3Ryl690ciOvdHljr3Ryl690ciOvdHljr3Ryl690ciOvdHljr3Ryl690ciOvdH/qjUN/6xEIP+tTzf/qlxQ/6p
nZv6tb3b6rnOF8615lO2rf6HnqlWs4qSMtt6dkb7blpbF2JCaytaLn87ViKXQ1ler0dSHs9LTiL3Syo29
0ciOvdHljr3Ryl690ciOvdHljr3Ryl690ciOvdHljr3Ryl690ciOvdHljr3Ryl690ciOvdH/qjUN/6xEI
P+tTzf/qlxQ/6pnZv6tb3b6rnOF8615lO2rf6HnqlWs4qSMtt6dkb7blpbF2JCaytaLn87ViKXQ1ler0dS
Hs9LTiL3Syo290ciOvdHljr3Ryl690ciOvdHljr3Ryl690ciOvdHljr3Ryl690ciOvdHljr3Ryl690ciOvdH/qj

UN/6xEIP+tTzf/qlxQ/6pnZv6tb3b6rnOF8615IO2rf6HnqIW4s4qSMtt6dkb7blpbF2JCaytaLn87ViKXQ
1Ier0dSHs9LTiL3Syo290ciOvdHlj3Ryl690ciOvdHlj3Ryl690ciOvdHlj3Ryl690ciOvdHlj3Ryl690ci
OvdH/qjUN/6xEIP+tTzf/qlxQ/6pnZv6tb3b6rnOF8615IO2rf6HnqIW4s4qSMtt6dkb7blpbF2JCaytaLn8
7ViKXQ1Ier0dSHs9LTiL3Syo290ciOvdHlj3Ryl690ciOvdHlj3Ryl690ciOvdHlj3Ryl690ciOvdHlj3
Ryl690ciOvdH/qjUN/6xEIP+tTzf/qlxQ/6pnZv6tb3b6rnOF8615IO2rf6HnqIW4s4qSMtt6dkb7blpbF2JC
aytaLn87ViKXQ1Ier0dSHs9LTiL3Syo290ciOvdHlj3Ryl690ciOvdHlj3Ryl690ciOvdHlj3Ryl690ciO
vdHlj3Ryl690ciOvdH/qjUN/61FIP+uTzf/qlxQ/6tnZf6vbXT7r3KE9K93ku2tfZ/oq4Or46iLtd+ikb3bm5
XF2JSZy9WPns/Ui6TS04ir09OItnTOi7zTxo+808SPvNPEj7zTxI+808SPvNPEj7zTxI+808SPvNPEj
7zTxI+808SPvNPEj7zTxI+808SPvNP/qjUN/61FH/+uUDb/q1xQ/65nZP2wa3P7sW+C9bF0kO6wep
3prn+p5KuGs9+oj7zcopTE2JuYytWUndDTjaLT0omq1dKJt9bljrvVwJC61r6Quta+kLrWvpC61r6Qu
ta+kLrWvpC61r6Quta+kLrWvpC61r6Quta+kLrWvpC61r6Qutb/qzUN/65FH/+vUDb/rFxQ/69nY/yya
nH6s22A9rNyju+zd5vqsny5a+CsOGsi7ncqJPC2aKXydWam9DTkaHU0YUq18+LudjCkLnYupG5
2LiRudi4kbnYuJG52LiRudi4kbnYuJG52LiRudi4kbnYuJG52LiRudi4kbnYuJG52LiRudj/qzUN/65FH
/+vUDb/rF1Q/rBmYvuzaHD5tWt+97ZwjPG2dZnrtnqk5rR/reKxhbrero+/2qmWx9ahms7SIZ/V0lyq2c
eMt9u7kbjas5K327GSt9uxkrfbsZK327GSt9uxkrfbsZK327GSt9uxkrfbsZK327GSt9uxkrfbsZK327G
St9v/qzUM/69GH/+wUTb/rV1Q/rFIYfu0Z2/4t2p89rhuiV6K6cpbtunih6Lp9quS5g7PgtYu727CSxNanls
7QmpvVxJGk27yOtN2ykrbdrZO23auTtt2rk7bdq5O23auTtt2rk7bdq5O23auTtt2rk7bdq5O23auTtt2r
k7bdq5O23auTtt3/rDUM/69GH/+wUTb/rV1Q/bJkYPq2ZW33uWh79Ltsh/K9cZPuv3ad6sB7pubCg
a7iv4e33LeKw9Stj83HopbVuZed26+Sq9+pLlXfppS13qWUtd6llXepZS13qWUtd6llXepZS13qW
Utd6llXepZS13qWUtd6llXepZS13qWUtd7/rDUM/69GH/+xUTb/r1Q/bNiX/m3ZGz2u2d5875qhfD
Bb5DuxHSZ68h7oenMgKniyn+22cGCwsy1isy+qpHVr56X3KOXo+Gel7Phnpa04J6WtOCelrTgnpa
04J6WtOCelrTgnpa04J6WtOCelrTgnpa04J6WtOCelrTgnpa04J6WtOD/rDUM/7BGH/+xUTb/r1Q/L
RhXfi5Y2r1vWV38cFpgu7FbYzrynOV59F7m+PUfKjc03u20cl+wcS9hcy3sozUp6aR25mdm+GSnr
PilZq04ZaatOCWmrTglpq04JaatOCWmrTglpq04JaatOCWmrTglpq04JaatOCWmrTglpq04JaatOD
/rTYM/7BGHv+xUjb/r11Q/LVgXPe6YWnzv2R178Rnf+vKbljm0XSP4tl5l9zbeKjT1Xu1yc99wb3Ggc
uwu4fToK6L25ClleGJp63ijaG04Y6ftOCOn7Tgjp+04l6ftOCOn7Tgjp+04l6ftOCOn7Tgjp+04l6ftOC
On7Tgjp+04l6ftOD/rTYM/7BHHv+yUjX/sF1P+7dfW/a8YGfxwmJy7Mhme+bPbILh2nWH3OF0l9Te
eafM1n20ws+AwLfJgsqyxYLSmriG2omwkN+Csqnhh6q04lintN+lp7TfiKe034intN+lp7TfiKe034intN
+lp7TfiKe034intN+lp7TfiKe034intN//rTYM/7FHHv+zUjX/sl5O+rhdWfS+X2TuxWFu6M1lduHXbHvb
4XKE1uR1ls7eeqbF1n+znCDv7PKhsmmxYPRIMKB2IS/jN2AvqTeg7i23YOztt2Ds7bdg7O23YOz
tt2Ds7bdg7O23YOztt2Ds7bdg7O23YOztt2Ds7bdg7O23YOztt3/rjYM/7FHHv6zUzX+s1xM+LpbV/L
BXWHqyWBp4tRkb9vfaXfV5G6lz+Z0mMfffKW/14CyuNCFvq/LhseixoLPIMOB1ofBidmEwZvahcG
u2YLBtNmCwbTZgsG02YLBtNmCwbTZgsG02YLBtNmCwbTZgsG02YLBtNmCwbTZgsG02YLBtN
n/rjYL/rJIHv60UzX9tVpK9rxaVO7FW1z1z15j291iaNTkZnzO6GuNx+RxnL/deKm51n+0s9CFvqnMg
8aeylDNk8WB0ovDiNWJw5bVisSl1ljDqdWlw6nViMOp1YjDqdWlw6nViMOp1YjDqdWlw6nViMOp
1YjDqdWlw6nViMOp1YjDqdX/rzYL/bNIHf21VDT7t1hH88BYUOnKWFbd2VtY1ONebc3pZIDG6m
mSv+JvoLjbd6yz1X+2rNCBvqLMfsWYyX3LkMd+z4rFhdGlxZDSicad0InGodCJxqHQicah0InGodC
JxqHQicah0InGodCJxqHQicah0InGodCJxqHQicah0InGodD/sDcL/LRJHfy2VDT4ulZD7sRVSeHT
VEzV4VZbzeldccbuY4W+6GiVuOBvo7LaeK2t1X22o9F6vZrNecOSy3nHjMI7y4flgM2Ex4nNhMiUz
YTII82EyJfNhMiXzYTII82EyJfNhMiXzYTII82EyJfNhMiXzYTII82EyJfNhMiXzYTII83+sTcK+7ZKHPu
3UzPzviM+5sxQQNfeTkfO6VVgxvBcdr7vY4i35mmYseBxpKzad62j1Xa1mtJ0u5PPdb+MzXbDh8x
4xYPLfMeAyoLlf8qLyH/Ljsh/y47lf8uOyH/Ljsh/y47lf8uOyH/Ljsh/y47lf8uOyH/Ljsh/y47lf8uOyH/Ljsj8

[illegible]

eBvs3JhsDMwlrAzLmLwM21i8DNtYvAzBWLwM21i8DNtYvAzBWLwM21i8DNtYvAzBWLwM21i8DNtYvAzBWLwM3/pC0J/6c9Gv+oSS//pVVH/6xfVf6xZmL8s2lv+rRte/m0cojzs3eT7rF9numtgqflqlmw4Z+QudyUlsLWiZ/K0IWqzsyEuM/CiL7PuYy+z7ONvs+vjb7Pr42+z6+Nvs+vjb7Pr42+z6+Nvs+vjb7Pr42+z6+Nvs+vjb7Pr42+z6+Nvs//pC0J/6c9Gf+oSS//pVVH/65fVP2yZGD7tWdt+bZrefe3clX1t3WQ8LZ6muu0gKTnr4Ws4aWKuNeaksLNkJrKxYqlz8CJsdG6i73Sso690a2OvdGrjr3Rq4690auOvdGrjr3Rq4690auOvdGrjr3Rq4690auOvdH/pS0J/6g9Gf+pSS//pIVH/69fU/20Y1/6tmZr+Lhpd/a6boL0u3OM8bt5luy7f5/ltYGr26uGt8+gjsLEI5bKu5Cf0LWNq9OxjbvUq4+806eQvNOIkLzTpZC806WQvNOIkLzTpZC806WQvNOIkLzTpZC806WQvNP/pS0J/6g9Gf+pSi//p1VG/7FfUvy1YV75uGVp9rtodPO9bX/xv3KI7sF4kenCfZzgvHyq1bGCtsmmisG9nZLKs5Wb0auRpdSnkLTWo5G61aGRu9WgkbvUoJG71KCRu9SgkbvUoJG71KCRu9SgkbvUoJG71KCRu9SgkbvUoJG71KCRu9T/pS0J/6g+Gf+pSi7/qFVF/7JfUPu2YVz3umRn9L1ocvHBBHvuxHKE6sh4i+TleJvbwneqz7d+tsKshsC2o47KrJuW0aKVn9WclK3XmpS515qTutaak7vVmpO71ZqTu9Wak7vVmpO71ZqTu9Wak7vVmpO71ZqTu9Wak7vVmpO71ZqTu9X/pi0J/6k+Gf+qSi7/qIVE/rNdT/m4YFr1vGNI8cBnb+3Fa3fqynF+5s91ieDRcpvUynOpyL57tbyzgsCwqonJpaKR0ZqbmdaSmayYj5q52JGXudeTlrrWk5a61pOWutaTlrrWk5a61pOWutaTlrrWk5a61pOWutaTlrrWk5a61pOWutb/pi0I/6k+Gf+qSi7/q1VD/bRdTvi5X1jzvmJi7sRma+rKa3LI0nJ34NdwiNrdaZrO0XCowsV3tLa6fr+qsYTIInqmL0JGjk9aloKHZhK42YmdudeLmrrWi5q61ouautaLmrrWi5q61ouautaLmrrWi5q61ouautaLmrrWi5q61ouautb/pi4I/6o+Gf+rSi7/rVVC/LVcTPa7XlBwwWf68hIZuXQamvf2m9z2eFrh9Ljapnl2m2nu85zs6/Deb6jun/HlrKGz4qsjtWAqpyYfayz14KmutaEobrWhKG61oShutaEobrWhKG61oShutaEobrWhKG61oShutaEobrWhKG61oShutb/py4I/6o+Gf+sSy7/r1VB+rdbSvS9XVPtxV9a5s5jX9/Za2HZ4Wxy0uZthsrmbZfA3m6mtNhvsqjOdLybxXnGj72AzYO5idJ7t5nVd7uw1X2yvNR/q7vUf6u71H+ru9R/q7vUf6u71H+ru9R/q7vUf6u71H+ru9R/q7vUf6u71H+ru9T/py4I/6s/GP+sSy7/sVY/+LlaR/HAW07pyl1U4NZiVtnhaGDT525xzOtvhMPncJa633KksNh1sKPSdLuVzXXEiMI6y37HhM94xZXrd8ao0XjDvtB6ub3Qerm90Hq5vdB6ub3Qerm90Hq5vdB6ub3Qerm90Hq5vdB6ub3Qerm90Hq5vdD/qC4I/6w/GP+tTC39tFg89bxYQ+zFWEni0VpL2d9eUtLmZWXL7Gp2xO9vhr3ocpW04Hajq9l4r5/TdrmSz3bBhst5x37Jgct7yl7Mesmey3nJsMp2yb3Ldsm9y3bJvct2yb3Ldsm9y3bJvct2yb3Ldsm9y3bJvct2yb3Ldsm9y3bJvcv/qS4I/60/GP+uTC36t1Y58cBVPubMVEHa3FVB0uVcV8vsYmrD8Wd8vO1si7XlcJmu33WlpNI1r5jUdLeN0HS+hM54wn/MfsV8zInGfMyWxXzNpcR6zK7FesyuxXrMrsV6zK7FesyuxXrMrsV6zK7FesyuxXrMrsV6zK7FesyuxXrMrsX/qi4H/65AF/+wTS33ulQ07MVRN97WTzXT41RHy+xbXMPyYXC882aAtetrj67kcJyl3nGmm9pvrpHVcLWI0nK6gdb1vnzPesB5zoLBeM6NwXjPmsB5z6O/ec+jv3nPo795z6O/ec+jv3nP0795z6O/ec+jv3nPo795z6O/ec+jv3nPo7//qy4H/69BF/+yTizyviAu5M5KLNTThSzXL61NMw/NaYbv4YHS08maErupskaXkbJyb32ulkttrlrYbbGD1W+1fdNyuHnSd7p30n27ddKFvHTSj7t00pe7dNKXu3TSI7t00pe7dNKXu3TSI7t00pe7dNKXu3TSI7v/rS8G/rJCFvm4UCbrxkkk190/l8zqSzvD81NRu/lbZrT5Yneu8WmFpepokZvlZpuS4Gaiid1oqlPbaax92WyvedhvsnbWc7Nz1ni0cdV+tXDVhrVw1Yy1cNWMtXDVLVw1Yy1cNWMtXDVLVw1Yy1cNWMtXDVLVw1Yy1cNWMtXDVLX/ry8G/LRDFfO+Sh3e0zoWzudBKcTzS0C7+IRVtP5daK74ZXek8WOEmuthj5DnYZel42KdgeBkonzfZqV33WmodNxsqnHbb6tv23Osbdp3rWzafq1r2oOta9qDrWvag61r2oOta9qDrWvag61r2oOta9qDrWvag61r2oOta9qDrWvag638si8F+LhFE+fJOxDQ5DQYxPFDLrv7TUSz/1dXrf9gaKL5XnaY81yBju5cioXqXJF+512WeeVgmnXjY51y4mafb+FooG3ha6Jr4G6iaeByo2ffd6Rm33qkZt96pGbfeqRm33qkZt96pGbfeqRm33qkZt96pGbfeqRm33qkZt96pGbfeqT4ti8E8MA+C9TeJARf8Dkdu/tGMrP/UEas/1IXof9YZpb8V3KM9IZ7g/JXg3vvWII17FqNcOtckG3pX5Jq6GGUaOhklWfnZpZI52mXZOZsmGLmcJlh5XOZYeVzmWHlc5lh5XOZYeVzmWHlc5lh5XOZYeVzmWHlc5lh5XOZYeVzmWHlc5n0uy8C29Y

XAcftLQ68+j0hs/9JM6r/UUWf/1BUIP9QYYr/UGuA+1FzePdSeXL1VH5t81aCafFYhGbwW4Zk712
HYu9fiWHuYYIf7mSKXu1mi13taYxc7WyMXO1sjFztblxc7WyMXO1sjFztblxc7WyMXO1sjFztblxc7
WyMXO1sjFztblznxyAAyuodA735NBGz/0Eiqf9HMP3/R0GS/0dOh/9lWX7/SmJ2/0xpb/1Obmn7UH
Jl+Jl1YvIVd2D4V3le91l6XPdbe1v2XHta9l58WfVhfVf1Y35X9WV+V/Vlflf1ZX5X9WV+V/Vlflf1ZX5X
9WV+V/Vlflf1ZX5X9WV+V/Vlflf1ZX7N2AgAvvcoBbT/OBKp/zohnP88LpH/PTuF/z9Ge/9BUHL/Q1h
r/0ZdZf9JYmH/TGVe/05oW/9RaIn/U2tY/1VsVv5XbVX+WG5U/VpuU/1cb1L9X3BR/GBwUfxgcFH8
YHBR/GBwUfxgcFH8YHBR/GBwUfxgcFH8YHBR/GBwUfxgcFH8YHD/mCYH/5o0Fv+aQCr/lk4//5
dZUf+bYF//m2ds/5luevqVdlb0jinqS74eCnOuBiqPoe5Kp5naZruVzobDkcai5HCvs+RwtrPkcL+z5H
HJs910zbLVeNCyz3rQsc960LHPetCxz3rQsc960LHPetCxz3rQsc960LHPetCxz3rQsc960LH/mCY
H/5o0Fv+aQCr/lk4//5dZUf+bYF//m2ds/5luevqVdlb0jinqS74eCnOuBiqPoe5Kp5naZruVzobDkcai5
HCvs+RwtrPkcL+z5HHJs910zbLVeNCyz3rQsc960LHPetCxz3rQsc960LHPetCxz3rQsc960LHPet
Cxz3rQsc960LH/mCYH/5o0Fv+aQCr/lk4//5dZUf+bYF//m2ds/5luevqVdlb0jinqS74eCnOuBiqPoe5K
p5naZruVzobDkcai5HCvs+RwtrPkcL+z5HHJs910zbLVeNCyz3rQsc960LHPetCxz3rQsc960LHPe
tCxz3rQsc960LHPetCxz3rQsc960LH/mCYH/5o0Fv+aQCr/lk4//5dZUf+bYF//m2ds/5luevqVdlb0jinq
S74eCnOuBiqPoe5Kp5naZruVzobDkcai5HCvs+RwtrPkcL+z5HHJs910zbLVeNCyz3rQsc960LHP
etCxz3rQsc960LHPetCxz3rQsc960LHPetCxz3rQsc960LH/mCYH/5o0Fv+aQCr/lk4//5dZUf+bYF//
m2ds/5luevqVdlb0jinqS74eCnOuBiqPoe5Kp5naZruVzobDkcai5HCvs+RwtrPkcL+z5HHJs910zbL
VeNCyz3rQsc960LHPetCxz3rQsc960LHPetCxz3rQsc960LHPetCxz3rQsc960LH/mCYH/5o0Fv+a
QCr/lk4//5dZUf+bYF//m2ds/5luevqVdlb0jinqS74eCnOuBiqPoe5Kp5naZruVzobDkcai5HCvs+Rwtr
PkcL+z5HHJs910zbLVeNCyz3rQsc960LHPetCxz3rQsc960LHPetCxz3rQsc960LHPetCxz3rQsc96
0LH/mCYH/5s0Fv+aQCr/l04//5hZUf+cYF7/nGds/5ptefqWdlb0kHmS74mBm+uCiqPofJKq5neZruV
zoLHkcaiz43Cvs+NwtrTjccC043HKs9x1zrPTedCyzXvQss170LLNe9CyzXvQss170LLNe9CyzXvQ
ss170LLNe9CyzXvQss170LL/mSYH/5w1Fv+bQSn/mE4//5tYT/+gYFz/oWdq/6Btd/ucc4T1l3iQ8l9+
muylh6PogY+q5XuXr+R2nrPjdKe14nKvtuJyuLflic8O33XXLttR5zrbMfc61x33Ntsd9zbbHfc22x33Nt
sd9zbbHfc22x33Ntsd9zbbHfc22x33Ntsd9zbb/miYH/5w1Fv+cQSn/mU4//55XTf+jX1r/pWdn/6VsdP
yicoH2nXeN8Zd9mOyPhKLoh4yq5X+UsON6nLXhdqW34XSvueF0urnhdMi51nnMuM59zLjGf8y5
wH/LucB/y7nAf8u5wH/LucB/y7nAf8u5wH/LucB/y7nAf8u5wH/LucB/y7n/miYH/501Ff+dQSn/mk4//6
FXTp+mX1j/qWZl/6lscv2ncn74o3eL8p57lu2WgaDpjYmp5YSRsOJ9mrhbheKS54HWvu991vbzaeM
m70H3Lu8eAy7u/gcq8uoHKvLqByry6gcq8uoHKvLqByry6gcq8uoHKvLqByry6gcq8uoHKvLqByrz/
myYH/541Ff+eQSn/m04//6NXSv+pX1f/rGZj/61scP6scnz5qXal86R6k+6df53qllan5oqPr+KBmLbge
qO73nevvdt3vr7Sesm+yH/JvsCCyb65g8i/tlPlv7SDyL+0g8i/tlPlv7SDyL+0g8i/tlPlv7SDyL+0g8i/tlPl
v7SDyL//myYG/541Ff+eQin/nE4+/6VXSf+rX1X/r2Zh/rBrbf2vb3n7rXWE9ap6kPCKfprrm4Ok5pGMr
d+HlbbYfp+80nuqwM96uMHLfMfBwYHHwbqEx8GzhcfBr4XHwa+Fx8GvhcfBr4XHwa+Fx8GvhcfB
r4XHwa+Fx8GvhcfBr4XHwa+Fx8H/nCYG/581Ff+fQin/nk49/6dWSP+uX1P+sWZf/LJqa/uybnb4sX
OB8654jO6qfpfoo4Kh4JiJrNeNkbbOhJq9yIClwcR+scPBf8LEuYPFxLOGxcOuhsXEqofFxKqHxcS
qh8XEqofFxKqHxcSqh8XEqofFxKqHxcSqh8XEqofFxKqHxcT/nCYG/581Ff+gQij/oE48/6lWR/+wY
FH8s2Vd+rRpaPi1bXP1tHJ977N3iOqwfJLjqX+f2Z6Fq8+TjrXGi5a9voWgw7mDrMa2g7rGsoXExq
2IxMaoiMTGpYnExqWJxMalicTGpYnExqWJxMalicTGpYnExqWJxMalicTGpYnExqWJxMb/nSYG/
6A2Ff+gQij/oU07/6tXRf6yYVD6tGRa+LdoZfa4bG/xuXF57LI2gua2eY/drnue06SCqsiairW+kZK+toq
bxLCHpsish7TJqYjCyaWKwsiiisPlolrDx6CKw8egisPHolrDx6CKw8egisPHolrDx6CKw8egisPHolr
Dx6CKw8f/nSYG/6A2Ff+hQij/o006/6xXRPyzYE75tmNY9rInYvO8a2vuvnB06b92fOG8dl7YtHadzK
l+qsKghrS3l469rpCWxaeMocmji67Loly/y52NwcubjckLkmo3CyZqNwsmajcLJmo3CyZqNwsmajcLJ

mo3CyZqNwsmajcLJmo3CyZqNwsn/nSYG/6A2FP+hQij/pE06/65YQ/u0X0z3uGJV87xmXvDAamf
sw3Bu5cVxe9zDbY3SunOcx697qbymg7SxnYq9p5aSxZ6RnMqZkKjNlpG4zZSQwM2Uj8HLII/Byp
SPwcqUj8HKII/BypSPwcqUj8HKII/BypSPwcqUj8HKII/BypSPwcr/niYG/6E2FP+iQyj/pk04/7BZQfm2
Xkr1umFT8b9kW+zEaWLoynBn4cxretfKalzMwHCbwbV4qLasf7Oqpla8oJ2NxZaXI8uPlaPOi5ayzo
uWv86Mk8DNjZLBy42SwcuNksHLjZLBy42SwcuNksHLjZLBy42SwcuNksHLjZLBy42Swcv/niYG/6
E2FP+iQyj/p043/rJaP/i4XUfzvWBP7cNjVujKaFvj0Wtl3NdiedHRZYvGxm2au7x0p7Cze7Kkq4G8m
KSIXl2ekcuFnJ3OgZ2tz4Cdv86EmcDNhpbAzlaWwMyGlsDMhpbAzlaWwMyGlsDMhpbAzlaWwMy
GlsDMhpbAzlaWwMz/nyYG/6I2FP+jQyj/qU42/LRbPfa6XETwwF5L6chhUOPSaFLc22Rj1d9feMvY
Y4nAzmqZtcRwpqm7drGes3y6kayDw4WnjMp9pZjNeaaazneov818ocDMfp3By36dwct+ncHLfp3B
y36dwct+ncHLfp3By36dwct+ncHLfp3By36dwcv/nyYF/6M2FP+kQyf/rE80+rZZO/O9WkHsxVxF5M
9gSNzbZUvV4mRizuZjdsXgYoi52GaXrs5spKLFca+WvXe5ibZ9wX6yhsd2sJTLcrGky3C0u8t1rML
Kd6XCynelwsp3pcLKd6XCynelwsp3pcLKd6XCynelwsp3pcLKd6XCynelwsr/oCYF/6M3FP+IRCf/rI
Ay+LhYN/DAWDzny1k+3dlePdbiZUNP52dgx+xndb7oZ4az4WaWp9pno5vRbK2OyXG3gsN4vnfAg
cRww5DGbcChx2rDt8Zvu8TFcbHExnGxxMZxscTGcbHExnGxxMZxscTGcbHExnGxxMZxscTGcb
HExnGxxMb/oSYF/6Q3E/+mRCf+sVlv9LtVM+vGVDXg1FU01uBbPM/nYk/J7WhgwfJqc7fwaoSt6
GuUoeFooZPbaauG1my0etJzunLQfb9tz4vBa86cwWrPrsBqz8i/bMHHwWzBx8FswcfBbMHHwWz
Bx8FswcfBbMHHwWzBx8FswcfBbMHHwWzBx8H/oiYF/6Y3E/+nRSf6tVMr8L9SLeTNTyzX3IEvz+
daQsjYFXB9GVmuflodrHxbIW06W2SnONqno/da6iD2W2wetVytXPTerlw0oa6btKTum7TOrlt07O4
adPEumnTxLpp08S6adPEumnTxLpp08S6adPEumnTxLpp08S6adPEumnTxLr/oyYF/6c4Ev+rRiT
3uVEm68ZMJdraRiDQ5VA0yO5YSMD1Xlq5+2Nssfdne6nvaome6GeUk+NnnojeaKZ/22usd9hws
HLWd7Nv1oC0btaLtG7WI7Nu16aybtiysW7YsrFu2LKxbtiysW7YsrFu2LKxbtiysW7YsrFu2LKxbtiys
W7YsrH/pSYE/6k4Ev+wSB/xvkwe4dBCGtHjRSXI7VA5wPVXTbj8XWCx/WNwqvVmfpu/zIqU6GK
UiuRjnlHgZqN63WmndNxtq3Dac61t2nqua9mCr2rZjK5q2piuatqjrWrao61q2qOtatqjrWrao61q2qOt
atqjrWrao61q2qOtatqjrWrao63/pyYE/6s5Efm3Shjpx0IU1N84FsnsRyrA9VA/uPxXUrH/XmOp/GJyn
/VgfpTvX4mK6l6SgeZgmXvjY5514WahcN9qpG3eb6Zq3nSnaN17qGfdgghm3YyoZt6Vp2beladm3
pWnZt6Vp2beladm3pWnZt6Vp2beladm3pWnZt6Vp2belaf/qSYD/646EPG+RA/Y2SgJyuo8G8D1S
S+4/VFDsP9YVan/XWSe/FtylPVafYrwWoaB7FuNeelck3PnYJdv5WOabORnnGnja55n4nCfZeJ0n
2TieqBj4oKgYuKJoGLiiaBi4omgYuKJoGLiiaBi4omgYuKJoGLiiaBi4omgYuKJoGLiiaD/rCYC+rU7C
+PNLATM5y4OwfQ/ILj9SzOw/1JFp/9WVp3/VmSS/VVviPhVeX/0VoB48FiGcu5aimzsXI5p61+QZul
jkmTpZ5Ni6GqVYOhuIV/nc5Ze53mWXed+I13nfpdd536XXed+I13nfpdd536XXed+I13nfpdd536XX
ed+I13nfpf+sCUB7MlyAtDjHQPC8zUSuP1DJLD/TTWm/05Fm/9OVJD/TmCG/09qffxQcnb4Unhv9
VR9avRXgGbyWYNj8VyFYfBfh/vYohd72WJXO5oiVrublpZ7nGLWO11i1jtdYtY7XWLWO11i1jtdYt
Y7XWLWO11i1jtdYtY7XWLWO11i1jtdYv4tiIA09cJAMPxKAa4/DoVsP9GJaT/RTSZ/0ZDjv9GT4T/
R1p7/0pic/9MaWz+Tm5n/FFyY/pTdWD5Vndd+Fh4W/dbelr3XXtY9mB7V/ZifFb1ZX1U9WI+U/VtflP
1bX5T9W1+U/VtflP1bX5T9W1+U/VtflP1bX5T9W1+U/VtflP1bX7TxgYAxe8WALn8MAmv/zoWo/86
JJf/OzGM/z0+gv8/SHj/QVFw/0RZaf9HXmT/SmJf/0xIXP9PaFn/UmlY/1RrVv9WbFT+WG1T/lptUv5
dbIH9X29Q/WJvTv1lcE79ZXBO/WVwTv1lcE79ZXBO/WVwTv1lcE79ZXBO/WVwTv1lcE79ZXDEz
wQAuPwbAa7/Kwmi/y4Vlv8vIYv/Mix//zU2df84QG3/O0dl/z5NX/9BUVv/RFXV/0dXVP9KWVP/TFtR
/05cUP9RXU//U15N/1VfTP9WX0v/WWBK/1xhSf9eYUn/XmFJ/15hSf9eYUn/XmFJ/15hSf9eYUn/X
mFJ/15hSf9eYUn/XmH/jR8F/48tE/+NOST/ikg4/4xSR/+PWIT/j2Fh/4xobv+Gb3r6fnaF9niAjvJyiZXw
bZOb7mqbnu1no6HsZqqi7GWyo+xluqPsZsSj7GbPo+Zp06LfbNai2HDaodJy26HSctuh0nLbodJy2
6HSctuh0nLbodJy26HSctuh0nLbodJy26H/jR8F/48tE/+NOST/ikg4/4xSR/+PWIT/j2Fh/4xobv+Gb3r

6fnaF9niAjvJyiZXwbZOb7mqbnu1no6HsZqqi7GWyo+xlupPsZsSj7GbPo+Zp06LfbNai2HdaodJy26
HSctuh0nLbodJy26HSctuh0nLbodJy26HSctuh0nLbodJy26H/jR8F/48tE/+NOST/ikg4/4xSR/+PWIT
/j2Fh/4xobv+Gb3r6fnaF9niAjvJyiZXwbZOb7mqbnu1no6HsZqqi7GWyo+xlupPsZsSj7GbPo+Zp06L
fbNai2HdaodJy26HSctuh0nLbodJy26HSctuh0nLbodJy26HSctuh0nLbodJy26H/jR8F/48tE/+NOST
/ikg4/4xSR/+PWIT/j2Fh/4xobv+Gb3r6fnaF9niAjvJyiZXwbZOb7mqbnu1no6HsZqqi7GWyo+xlupPs
ZsSj7GbPo+Zp06LfbNai2HdaodJy26HSctuh0nLbodJy26HSctuh0nLbodJy26HSctuh0nLbodJy26H
/jR8F/48tE/+NOST/ikg4/4xSR/+PWIT/j2Fh/4xobv+Gb3r6fnaF9niAjvJyiZXwbZOb7mqbnu1no6HsZ
qqi7GWyo+xlupPsZsSj7GbPo+Zp06LfbNai2HdaodJy26HSctuh0nLbodJy26HSctuh0nLbodJy26H
Sctuh0nLbodJy26H/jR8F/48tEv+OOST/ikg4/45SRv+RWVP/kmFg/49nbf+Kbnn7gnWE9nt+jvJ1iJb
vcJGc7Wuan+xpoqLrZ6qj62axpOtmuqTrZ8Wk6mfRpONr06Tcbtej1HLaos9z2qPPc9qjz3Pao89z2
qPPc9qjz3Pao89z2qPPc9qjz3Pao89z2qP/jh8F/5AuEv+POST/jEg4/5JRRf+WWVH/l2Be/5Vmav+
QbHf8iXOC9oF6jfJ6hJbvd16c7G+XoetroKTqaamm6mixp+povKfqaMin5WvRp91v1KbVctilznXYps
h116flddenyHXXp8h116flddenyHXXp8h116flddenyHXXp8h116f/jx8F/5EuEv+QOiT/jUg4/5VQQ/+
aWE//m19b/5plaP+Wa3T9kHGA94h4i/OAgZXveYuc7HKVoupunqbpa6io6Gmxqehpvqroas2p327
SqdZy1qjPdtox3fVqcJ41arCeNWqwnjVqsJ41arCeNWqwnjVqsJ41arCeNWqwnjVqsJ41ar/kB8F/5
luEv+ROiT/j0g3/5hQQf+dV03/n15Z/59kZf+banL+lnB++I92ifOFfZPvfoic7HaSo+IwnKfobKeq52uyr
OdrwKzkbc+r2XLTq9B21arleNSrwXnTrLx60628etOtvHrTrbx60628etOtvHrTrbx60628etOtvHrTrbx
6063/kB8F/5MuEv+SOiT/kkc2/5tPQP+gV0v/o15X/6NkY/+gam//nG97+pV0hvSMepHwg4Sb7HuPo
ulzmajlb6Ws4m6wruBuvq7eb9Cu0nTSrsl4063CetKuu3vSr7d80a+3fNGvt3zRr7d80a+3fNGvt3zRr
7d80a+3fNGvt3zRr7d80a//kR8F/5QuEf+TOiP/IEc1/51PPv+jWEn/pl9U/6dIYP+lzmz7om939px0g/G
UeY7si4GZ54KMouF6lqncdKCcu2HGssNRxubHScsqxynbRscJ60bG7fNCxtX3QsrF+0LKxftCysX7
QsrF+0LKxftCysX7QsrF+0LKxftCysX7QsrF+0LL/kh8F/5QuEf+UOiP/lkc0/6BPPf+mWEf/qWBS/6t
mXfyqbGj3p3B08qN0f+yceIrmllCV4lqJoNmAkqnSeZyvvXanssl2s7TGdsK0wnnPtLt8z7S1fs+0r3/
OtKyAzrSsgM60rIDOtKyAzrSsgM60rIDOtKyAzrSsgM60rIDOtKyAzrT/kh8E/5UuEf+UOiP/mEYy/6J
QO/+oWUX/rGFP/a5nWvmvbWXzrXfw7al0euikelXhm36T2ZCFn9CHjanlf5ewwnuhtL56rba7eru3
uXzNt7N/zbeugc23qYHNt6eCzbengs23p4LNT6eCzbengs23p4LNT6eCzbengs23p4LNT6eCzbf/kh8
E/5UuEf+VOyP/mkYx/6RQOv+rWkP9r2FN+rJnVvWya2HvsXBr6q90deOqdoLboHqS0peCn8iNiqn
AhpKxuYGctrR+p7mxfrS6r3/FuquBy7qng8u5o4PLuaGEzLmhhMy5oYTMuaGEzLmhhMy5oYTMu
aGEzLmhhMy5oYTMuaGEzLn/kx8E/5YvEf+WOyP/nEYw/6ZROP+tWkH7smJK+LRmU/K2alzstm
5l5rVzb96vclHVPnaRy51/nsGUhq4j16xsYaXt6uDoruogq68pYO+vaKFyryfhcq8nYbKu5uGy7ubhsu
7m4bLu5uGy7ubhsu7m4bLu5uGy7ubhsu7m4bLu5uGy7v/kx8E/5YvEf+WOyP/nkcv/6hRN/6wWz/5
tGFH9rdkT++5aFfpvG1f4rptbtm0bIDPq3OPxaN7nbuag6iikoqxqYyTuKOInbyehqm/m4e3v5mlyL+
XiMi+lojJvZWlyryViMq8lYjKvJWlyryViMq8lYjKvJWlyryViMq8lYjKvJWlyrz/lB8E/5cvEf+XOyP/oEcu/
6pSNfyyXDz3tmBE87pjS+y+Z1LmwmXZ3sBmbdS5aH7KsHGOWKh4nLWgf6esmlwopKOuJqNm
L6Vi6TBkYuywY+MxMGOjMfAjovlv46Kyb6Oism+jorJvo6Kyb6Oism+jorJvo6Kyb6Oism+jorJvo6Ky
b7/lB8E/5cvEP+XOyL/oUct/6xTM/q0XDr1uV5A8L5hRunDZkzix2ZY2shfbM+/Zn3Etm6Nuq51m7C
mfKamn4OwnJiKuJKTk76LkJ/Ch5Ctw4WRvsOFkMbCho/HwleOyL+Hjsi/h47lv4eOyL+Hjsi/h47lv4
eOyL+Hjsi/h47lv4eOyL//lR8E/5gvEP+YOyL/o0gr/65TMfi2WzfyvF087MJfQebKZETez15W1M5cas
nFY3y/vGuMtLRymaqteKWgpn6vIZ+Ft4qajr6Cl5rCfZeoxHuYucN7l8bCfpTHwYCSyMCAksjAgJLI
wlCSyMCAksjAgJLIwlCSyMCAksjAgJLIwlCSyMD/lR8E/5kvEP+ZOyL/pUgq/bFUL/W5WTTvwFo3
6MhdOuHSYD/Z3FNVz9VZacTMYXq5w2eKr7xumKS0dKSZrXqujaeBtoKiib15oJXBdZ+jw3KhtMN
yocbCdvpHwHmYyL95mMi/eZjlv3mYyL95mMi/eZjlv3mYyL95mMi/eZjlv3mYyL//lh8E/5kvEP+aPC

L/qEko+7RVLPO8Vy/rxVgx4tBbMdndVz3S4VdTyd1YZ77UXXmzzGSIqMRqlp29b6KStnWshrB8tH
ushLtyqpC/bqqfwGursMBqrci/b6XlVnGgyb5xoMm+caDJvnGgyb5xoMm+caDJvnGgyb5xoMm+ca
DJvnGgyb7/lx8D/5ovEP+bPCH/q0ol+LdVKO/AVCrly1Qp29tYJdPiWzvL6FxRwuRdZbfdXHes1mC
Goc9llJblaCkKwXCpfrx2sXS4f7dst4y7aLecvGa5rbtku8a6abLLumury7trq8u7a6vLu2ury7trq8u7a6
vLu2ury7trq8u7a6vLu2ury7v/mB8D/5wwD/+fPR//rksi9LpSI+rFTyLd1U4f1OJWKc3oXznF7mFPvOt
iY7HnYnWl4V+Emdtgko7VZZ2Bz2qmdstxrm3le7Nnylm1Y8matmHKq7VgzMO1ZMPPTGa5zrVmu
c61ZrnOtWa5zrVmuc61ZrnOtWa5zrVmuc61ZrnOtWa5zrX/mR8D/50wD/+jPrz8skwd8L5OHeLOR
xnV4EoezOhVLsXvXT++9GJQtvNmYa3wZ3Kg7GKBk+dgj4XjYZp532Wib91sqWfbd61j2YWuYdm
Vr2Dapa5f2reuX9nSrGHM0q5hzNKuYczSrmHM0q5hzNKuYczSrmHM0q5hzNKuYczSrmHM0q7/
mx8D/58wDv+oPhj2uE0Y6cZFFNfcPBLN6EsixfBVNL32W0W2+mBWrvlkZqP4YnWX81+Ci+xfjX/n
YJZ242WebuBso2nedKZm3oCnZd6Np2TemqZk36ilY9+5pF/eyqZf3sqmX97Kpl/eyqZf3sqmX97Kp
l/eyqZf3sqmX97Kpl/eyqb/nB4C/6ExDv+uPxPwvkYQ3dQ0Cs7mPhbF8EwovfdUObX9Wkuu/19bpP
9eapn5XHeO8luCg+1ci3npXpNx5mOZa+NpnWficJ9k4XmhYuGDoWLhjaFh4ZmgYeKnn2Hjsp5h4
7KeYeOynmHjsp5h47KeYeOynmHjsp5h47KeYeOynmHjsp7/nx4C/6MxDfm1PwzmyTYH0elvC8bv
QRu9900ttf5VPq7/Wk+k/1lemf9Ybl75V3eE9FiAe+9ZiHLSXI5s6WCSZ+dmlmTmbJhh5XKZX+V6m
l7lgppe5YyaXeWYmV3mopld5qKZXeaimV3mopld5qKZXeaimV3mopld5qKZXeaimV3mopn/oh4B/
6oxCe+/OgTU3RwDx+41D733RCC1/k8xrv9VQqP/UIGY/1Jfjv9Ta4T6U3R79IV8c/JXgmzwWodn7l
6KY+xijWDrZ49e6m2QXOpzkVvqeZJa6oGSWeqKklngk5FZ6pORWeqTkVnqk5FZ6pORWeqTkV
nqk5FZ6pORWeqTkVnqk5H/pR0B+bQrAtzTDgDI6yYFvfc7E7X/RySt/080ov9MQ5f/TFGN/0xdg/9
OZ3r9T29y+lJ1a/dUemb1V35i81uBX/Jfg1zxY4Ra8GeGWPBsh1fwcYdW73elVe9+iFTvhYhU74WI
VO+FiFTvhYhU74WIVO+FiFTvhYhU74WIVO+FiFTvhYj/qRwA4MUKAMvoEgC+9zAlt8/Fqz/RSag
/0Q0lf9DQov/RU6B/0dZeP9JYXD/TGhp/05tZPxRcGD7VHNd+lh1Wvlbd1j4XnlW92J6VPdme1P2a
ntS9m58UPZ0fE/2eXxP9nl8T/Z5fE/2eXxP9nl8T/Z5fE/2eXxP9nl8T/Z5fE/2eXzZtgMAys4GAL/1IQG
1/zULqv86GJ//OiWT/zoyiP88Pn7/Pkl1/0FRbf9EWGb/R15h/0piXf9NZVr/UWdX/1RpVf9Wa1P/WW
xR/lxtUP5fbk/+Y25N/WZvTP1rb0v9b3BL/W9wS/1vcEv9b3BL/W9wS/1vcEv9b3BL/W9wS/1vcEv9b
3DJvgMAvdYEALP/IAKp/ywLnf8vF5L/MCOG/zlufP81OHL/OUFq/zxIY/8/TI3/QIJZ/0ZVVf9IWFP/S1
pR/05bT/9RXU7/VF5M/1ZfS/9ZX0n/XGBI/19hR/9jYUb/ZmJG/2ZiRv9mYkb/ZmJG/2ZiRv9mYkb/Z
mJG/2ZiRv9mYkb/ZmK9ywlAsOsEAKb/GgOc/yIKkP8IFIT/Jx55/ysob/8vMGB/Mjdf/zY9Wf86QVT/P
UVR/0BITv9DSkz/RktK/0hMSf9KTUf/TU5G/09PRf9SUEP/VFBC/1ZRQf9aUkD/XVJA/11SQP9dUk
D/XVJA/11SQP9dUkD/XVJA/11SQP9dUkD/XVL/gRoF/4EnEP9/NCD/e0My/4BMPv+CVer/glxW/3
5jYv93a27/cHV4/Wp/gPpkiob3YJSL9l2djvVbpZD1W62R9Fq1kfRav5H0WsmR81vWke1e25DnYd
6Q4WThj9pn44/VaeSP1Wnkj9Vp5l/VaeSP1Wnkj9Vp5l/VaeSP1Wnkj9Vp5l//gRoF/4EnEP9/NCD/e
0My/4BMPv+CVer/glxW/35jYv93a27/cHV4/Wp/gPpkiob3YJSL9l2djvVbpZD1W62R9Fq1kfRav5H0
WsmR81vWke1e25DnYd6Q4WThj9pn44/VaeSP1Wnkj9Vp5l/VaeSP1Wnkj9Vp5l/VaeSP1Wnkj9V
p5l//gRoF/4EnEP9/NCD/e0My/4BMPv+CVer/glxW/35jYv93a27/cHV4/Wp/gPpkiob3YJSL9l2djvVb
pZD1W62R9Fq1kfRav5H0WsmR81vWke1e25DnYd6Q4WThj9pn44/VaeSP1Wnkj9Vp5l/VaeSP1
Wnkj9Vp5l/VaeSP1Wnkj9Vp5l//gRoF/4EnEP9/NCD/e0My/4BMPv+CVer/glxW/35jYv93a27/cHV4/
Wp/gPpkiob3YJSL9l2djvVbpZD1W62R9Fq1kfRav5H0WsmR81vWke1e25DnYd6Q4WThj9pn44/V
aeSP1Wnkj9Vp5l/VaeSP1Wnkj9Vp5l/VaeSP1Wnkj9Vp5l//gRoF/4InEP+ANCD/fEMy/4JMPf+FU0n
/hVtV/4FiYf97am3/dHN3/W19gPlnilf3YpKM9V+cj/RdpJH0XK2S81u1k/Nbv5PzXMuT8lzYk+pg25L
kY96S3WbhkdZq45HRa+SS0WvkkfFr5JLRa+SS0WvkkfFr5JLRa+SS0WvkkfFr5JL/ghoF/4MnD/+
BNB//fkMx/4dKO/+Kukf/illT/4dgX/+CaGv/eW92/XJ6f/lrhlf2Zo+N9GGZkfNfo5TyXqyV8l21lvJdwJby

Xc6W7WDYleVk3JXeZ9+U1mviIM9s4pXLbeKVy23ilctt4pXLbeKVy23ilctt4pXLbeKVy23ilctt4pX/gx
oE/4QnD/+CNB//gklw/4pKOV+OUUX/j1hR/41fXf+IZmn/gG10/nh2fvlwgYf2aYyO82SXk/JhoZbxX6u
Y8V+1mPFfwpjwX9KY6GPZmN9n3ZfYa+CW0G7hI8lv4JjFcOCZxXDgmcVw4JnFcOCZxXDgmcV
w4JnFcOCZxXDgmcVw4Jn/hBoE/4UoD/+DNB//hUEu/41JOP+SUUP/k1hO/5JfWv+NZWb/hmty/n
1yffp1fYb2boiO82eUIPFjn5jwYaq72C1m+5hwpvsYtKa42famtlr3pnRb9+ZynDfmsNx3pu/ct6cv3L
enL9y3py/ct6cv3LenL9y3py/ct6cv3LenL9y3pz/hBkE/4YoD/+ENB//h0At/5FJN/+WUUH/mFIM/5dfW
P+TZWT/jWpv+4Vxevd7eoXzc4WO72yQlexom5npZaac52WxneVmv3jZ86c3WrbnNjv3pzKcd2c
xHLdnr103J+6dNfunTcn7p03J+6dNfunTcn7p03J+6dNfunTcn7p03J//hRkE/4coD/+FNB//ij8s/5R
JNf+ZUj//nFIJ/5xfVP+ZZWD6IGps9o1wd/GDd4Lte4GM6HOMIORul5rgaqKe3Wmt9tpuqDYasqg1
GzboMtx26DDdNugvXXbobd22qK0dtqitHbaorR22qK0dtqitHbaorR22qK0dtqitHbaorR22qL/hhkE/4
goDv+GNB//jT8r/5dKM/+dUjz/oFIG/6BfUfufZVz2m2po8ZRvdOuMdX/mhH+K4HuJk9t0k5vWb52g0
m2oos9ttKPMbsKkym/UpMJz2aO8dtmjtnfZpLF42KWueNilrnjYpa542KWueNilrnjYpa542KWueNilr
njYpa542KX/hhkD/4goDv+HNB7/kD8p/5pKMf+gUjr/pFpD/aVgTfekZVjyoWpj7Jxvb+aVdHrgjHyH2
YKfK9F6jzMDziix3KjpcRrxqfBcrunv3PMp7p116e0eNenr3nWp6t61qioe9aoqHvWqKh71qioe9aoq
HvWqKh71qioe9aoqHvWqKh71qj/hxkD/4koDv+INB7/kj8o/51KL/+jUzf/p1pA+qlgSvSpZITup2pe6K
RvaeGccnfZknmG0YmBksmBipzDe5OjvXedp7l2qKm2drSqtHfEqrF41aqse9WqqHzVqqV91aqifd
Wqon3VqqJ91aqifdWqon3VqqJ91aqifdWqon3VqqJ91ar/iBkD/4ooDv+JNB7/IEAn/59KLv+mUzX+q
1s9961hRvCuZk/qRwtY5KpuZNYib3XTmHaEypB+kcKlHpy6gY6ktH2YqbB6o6yteq6tqnu9rah80a2lf
tOtoX/TrZ5/062cf9OsnH/TrJx/06ycf9OsnH/TrJx/06ycf9OsnH/TrJx/06z/iBkD/4soDv+KNB7/lkAm/6F
LLP+pVDP7rls69LFiQe2zZknns2pS369pYtana3TNnnODxJZ7kLuOg5uzh4qkrYKTqqd/nq6kfqmwo
X+3sJ+AybCcgdGwmoHRr5iB0q+WgdKuloHSrpaB0q6WgdKuloHSrpaB0q6WgdKuloHSrpaB0q7/i
RkD/4soDv+KNB7/mEAK/6NLKv+rVDD5sVw28bVhPeq3ZETjuGdN27RiYdGsaXLlPHGBvpx4j7W
Uf5utjYekpYiPq5+Ema+ag6Wyl4Oxs5WEw7OThNCykoTQspGE0bGQhNGwkITRsJCE0bCQhNG
wkITRsJCE0bCQhNGwkITRsJCE0bD/iRkD/4woDv+LNB7/mkAj/6ZLKP6uVC32tF0z7rhfOOe8Yz7
fvWFM1rleX8ywZnHCqW6AuaF1jrCafJqnllOjno6Lq5eJILGRh5+0joettYulvbWKiM60iYfPtlqH0LKK
htCxiobQsYqG0LGKhtCxiobQsYqG0LGKhtCxiobQsYqG0LH/iHgD/4woDf+NNB3/nEai/6hMJvywV
Svzt1sv67xdM+TCYTbbxFdK0b5cXse1ZG+9rmt/tKdyjauheZmhmoCjmJSHq4+Pj7GlijJu1hlyot4G
MuLd/js22glzOtYKLz7SDitCzg4rQs4OK0LODitCzg4rQs4OK0LODitCzg4rQs4OK0LP/iHgD/40oDf+
PNRz/n0Eg/6pMJPmzVifwulkr6MFbLeDIWjTWyIFJzcNaXMK7Ym64tGI9r61vi6WndpeboXyikZuCq
oeWi7F/k5a2epKkt3iTs7d2lMe3d5LntnqPzrR7js+ze47Ps3uOz7N7js+ze47Ps3uOz7N7js+ze47Ps
3uOz7P/ixgC/44oDf+SNRv/oUEe/61Mlfa3ViTvlYl5cdYJdzRUDLS0E9Hx8lXW73CX2yzumZ8qbR
siZ+ucpaVqHigiqJ+qYCeHRB3m5G1cpqft2+br7dtnMO2b5rOtXKWz7R0IM+zdJTPs3SUz7N0IM+zd
JTPs3SUz7N0IM+zdJTPs3SUz7P/jBgC/48oDf+UNRN/pEEc/bBNHvO6Ux/pw1lf4dBUHNbcRjHM1
kxFwtBUWbfJXGqtwmJ6o7xoiJm2bpSOsHOeg6t6p3ingq5wpY2za6SbtGilq7Rmpr+0Z6XPs2uf0L
JsnNCybJzQsmyc0LJsnNCybJzQsmyc0LJsnNCybJzQsmyc0LL/jRgC/5ApDf+YNhf/p0EZ+bRNGu
++Txnky0wW2NxHG/sITS/G3k5DvNhSVrHSWGiny154nMVkhZG/aZKHum+ce7V1pHGyfatpslmv
ZbCYsGKxqLBhsruvYLLSrmSq0q9mptKvZqbSr2am0q9mptKvZqbSr2am0q9mptKvZqbSr2am0q//j
hgC/5EpDP+bNhX/q0lV9bhNFenFRxLb2EIN0eNOGMnoVCzA5VZbtuFWVKrcVmaf1lI1NBfg4nLZ
l9+xmmZdMJwoGrAeaZkvoWpYL+Vql3ApqpcwbmpW8LUqF+41algstWqYLLVqmCy1apgstWqYL
LVqmCy1apgstWqYLLVqmCy1ar/jxgC/5MpDP+gNhL9sEER8L5HDuDQOgnS4UAPyepPHMLvW
Cu67V0+sepeUaXnXGOZ4llyjN1bf4HZX4t21WSUbNjrm2TRdaBf0IKiW9GT01nTpKNY1LeiV9XS
oVrK2qJbwtqjW8Lao1vC2qNbwtqjW8Lao1vC2qNbwtqjW8Lao1vC2qP/kRgC/5UpC/+INQ73tkAL58

c5B9TeLwfK6kMTwvFPIrr0WDGz9F1CqvNfUZ/xXWGT7Vxuhupae3nnW4Zv5WCOZuRnIF/jcZhb
5H6aWeWOm1jln5pX5a6ZVubCmVfi25hX2N6aV9jemlfY3ppX2N6aV9jemlfY3ppX2N6aV9jemlfY3
pr/kxcB/5gpC/+sNAnwvjsF2NkZAcvoNQrC8kUYuvlQJ7P7Vzeq+1pHn/pYVpT4V2SI91ZwfPVWen
LyWIJp712JYuxkjV3qbZFa6XiSWemEk1jpkpJY6p+SV+qtkVfwrI9W69GPVuvRj1br0Y9W69GPVu
vRj1br0Y9W69GPVuvRj1br0Y//IRcB/58oBvmzLwPjzB8AzeYjA8LyOg66+UgdsV9RLKr/VDyf/1JLIP
9RWIn/UGR+/VFudPITd2v1Vn1k8luCX/BhhlvallyY7nGKVu16i1XthYtV7ZCLVO6cilTuqoIT77SJU+
+0iVPvtlIT77SJU++0iVPvtlIT77SJU++0iVPvtln/mBYB/6gjAu6/GwDQ3goAw/EtBrr6PxKy/0ohqf9O
MJ7/TD+T/0tMiP9KWH7/TGN1/05rbPxRcWX5VHZg91I6XPVefVj0ZIBW82qBVPJyglLyeoNR8oO
DUPKNG1DymYJP86GCT/Ohgk/zoYJP86GCT/Ohgk/zoYJP86GCT/Ohgk/zoYL/nBUA+LMQANDL
BwDE7xsAuvo0CbL/Qhao/0Qknf9FMpL/RD+H/0NLff9FVXT/SF5s/0tlZf9Oal/+Um5b/FZxWPtac1X
6X3VS+WV3UPhqeE/4cXhN+Hd5TPh/eUv4iXIL+JB5S/iQeUv4kHIL+JB5S/iQeUv4kHIL+JB5S/iQe
Uv4kHnopwQAz70EAMPTBgC5+yMCsP80C6b/Ohib/zslkP88MYX/Oz17/z5Hcv9BUGr/RFdj/0dcX
v9LYFn/T2RW/1NmU/9WaFH/WmpP/19rTf9jbEv+aG1J/m5tSP50bkf+e25G/oFuRv6Bbkb+gW5G/
oFuRv6Bbkb+gW5G/oFuRv6Bbkb+gW7PsgEawsYEALbbBAct/yIDpP8tDjN/MRiO/zljg/8zLnj/NTh
v/zlBZ/88SGD/QE1b/0NSVv9HVVP/SihQ/05aTv9RW0z/VV1K/1heSP9cX0b/YGBF/2VgRP9pYUL/
cGFB/3RhQf90YUH/dGFB/3RhQf90YUH/dGFB/3RhQf90YUH/dGHEvQIAt9ICAKv/CwCh/xwEI/8k
Dlz/KBaB/ykgdv8sKWz/MDFk/zM4Xf83Plf/O0JS/z5GT/9CSE3/RUpK/0hMSP9LTUb/Tk5E/1JPQ/9
VUEH/WFFA/1xSP/9gUj3/ZVM8/2hTPP9oUzz/aFM8/2hTPP9oUzz/aFM8/2hTPP9oUzz/aFO5ygA
Aq9sAAKD/CwCU/xMDif8aCX7/HhJy/yAaaP8IlmD/KShZ/y0uU/8xMk7/NTZL/zg4SP88Okb/PzxE/0
I9Qv9EPkD/Rz8+/0IAPf9MQTz/UEI6/1NCOF9WQzf/WkQ3/11EN/9dRdf/XUQ3/11EN/9dRdf/XUQ3
/11EN/9dRdf/XUT/dBgG/3EiDv9uMRz/aj8r/3JINv90T0H/c1hM/25gV/9oamL/YXVr/1uAcv9WjHf/Up
d6/IGgff5QqX7+T7F+/U+6f/1PxH/8T9B/+0/ffvVS5H7vVud96VnpfeNc63zdX+1822DtfNtg7XzbYO1
822DtfNtg7XzbYO1822DtfNtg7Xz/dBgG/3EiDv9uMRz/aj8r/3JINv90T0H/c1hM/25gV/9oamL/YXVr/
1uAcv9WjHf/Upd6/IGgff5QqX7+T7F+/U+6f/1PxH/8T9B/+0/ffvVS5H7vVud96VnpfeNc63zdX+1822
DtfNtg7XzbYO1822DtfNtg7XzbYO1822DtfNtg7Xz/dBgG/3EiDv9uMRz/aj8r/3JINv90T0H/c1hM/25g
V/9oamL/YXVr/1uAcv9WjHf/Upd6/IGgff5QqX7+T7F+/U+6f/1PxH/8T9B/+0/ffvVS5H7vVud96Vnpfe
Nc63zdX+1822DtfNtg7XzbYO1822DtfNtg7XzbYO1822DtfNtg7Xz/dRgG/3EiDv9uMRz/bT8q/3VHN
f94TkD/dlZL/3JfV/9raGH/ZHNR/15+cv9YiXj/VJV8/IKffv1RqH/9ULGA/VC6gPxQxYD7UNKA+IDhgP
NV5H/sWOd/5lvpfuBe637ZYe1+1mHtf9Zh7X/WYe1/1mHtf9Zh7X/WYe1/1mHtf9Zh7X//dhgG/3MiD
f9wMRv/cT0p/3IFM/99TT7/fFRJ/3hcVf9xZWD/anBq/2J6cv9chnn+V5F9/VScgPxTpoL7UrCD+1G6
g/tRxoP6UtWD9lThgu5Y5ILnXOeB4F/pgdpi64HSZOyC0GTtgtBk7YLQZO2C0GTtgtBk7YLQZO2C
0GTtgtBk7YL/dxgG/3QiDf9yMRv/dTwo/31DMv+BTDz/gVRH/31bU/93Y17/b2xp/2d3cv9ggnn9Wo5
+/FaagvtUpIT6U6+F+IO5hvhUxYb2VdSF8lfhheIb5lThX+eE22Pqg9NI64TMZuuGymfrhspn64bKZ+
uGymfrhspn64bKZ+uGymfrhspn64b/eBgG/3UiDf9zMBv/eDsn/4FDMP+GTDr/hlNF/4NaUP99YIz/d
Wln/210cf5lf3n7Xot/+VqXhPdYoYb1V6uH81e1iPFYwYjwWc+l7Vrgh+Rf5YbcY+iG1Gbqhs1n6ojG
aOmJxGnpicRp6YnEaemJxGnpicRp6YnEaemJxGnpicRp6Yn/eRgF/3YiDf90MBv/fDom/4VDLv+K
TDj/i1NC/4laTv+EYVn/fWhl/HRxb/hse3j1ZYeA8mCShe9dnYjtXKeK61yxiuldvYroXsuK5V/did5j5oj
UZ+iJzGjoisZq6lvAa+eNvmvnb5r542+a+eNvmvnb5r542+a+eNvmvnb5r543/ehgF/3giDf91MBv/fz
kk/4IDLp+PTDX/kFNA/49aSv+LYFb7hWdi9nxubfJzeHfubIN/6maOhudimYrkYaOM4mCujeBhuY3
eYseN3GPZjdVm5o3MaueNxWvmjr9s5Y+5buWQt27IkLdu5ZC3buWQt27IkLdu5ZC3buWQt27IkL
du5ZD/ehgF/3kiDP93MBv/gzkj/41DKv+TTDP/IVM8/5VaR/uSYFL2jGZe8YVsaex8dXTndH9+4m2J
ht5ollvaZZ+P12SpkdRktJHSZcKR0GbSkcxo5JHEbOSRvW7kkrhv45OzcOOUsXHjILFx45SxceOU

sXHjILFx45SxceOUsXHjILFx45T/excF/3oiDP94MBv/hjkh/5BDKP+XTDD/mIM5/JpaQ/aYYE7xIGV
Z641rZeaFc3DgfXx72nSFhtRuj43Qa5mSzGmkIMlprpXHaruWxGvKlsJs3pW7b+KVtXHhIrFy4Zesc
+GXq3Phl6tz4Zerc+GXq3Phl6tz4Zerc+GXq3Phl6tz4Zf/fBcF/3siDP96Lxr/iTkg/5RDJv+aTC3/nIM1
+Z9aP/KeX0nsm2VU5pVqX+COcWzZhHh60XuBhct1io7GcZSUwm6el75uqZm8brSauW/Dmrw1
pmzct+ZrnTfmqp135qmdt+apXbfmqV235qlt+apXbfmqV235qlt+apXbfmqV235r/fRcE/3wiDP98L
hn/jDke/5dDJP+eTCv8olQy9aRaOu6kX0PoomRN4p1oWtqUbmRi3V5yoN9hcN8ho69d4+VuHSZ
mrVypJyyccq+dr3O9na10zp2qdt2dpnfDNA43Z2geN2dnndnZ553Z2eed2dnndnZ553Z2eed2dnndn
Z553Z3/fRcE/3wiDP9/Lhj/jzkd/5IDlv+hTCj5plQu8qhaNuupXz7kqGRH3aJkWNSaa2nLkXN3w4
l6hLyCgo61fYqWsHmUm6x3n5+pdqqgpne2oaR4x6GiedugnnrboJx726CZe9yfmHvcn5h73J+Ye9y
fmHvcn5h73J+Ye9yfmHvcn5h73J//fhcE/30iDP+BLhf/kTkc/5xDIP+kTCX3qVQR761aMeeuXzjhrWB
D2KdgVs+faWfGI3B2vY93g7WJfo6ug4aWqH+PnaR8maGge6WjnXuxpJp8waSYfdWjln3Zo5R92q
OSfdqikn3aoZJ92qGSfdqhkn3aoZJ92qGSfdqhkn3aoZJ92qH/fxcE/34iC/+ELhb/kzka/55DHv6mTC
P0rFQn7LFaLOS0XzLcsllC06teVMqkZmXBnG50uJV1gbCPfl2oiYOWoYSLnZuBlaKXf6CllH6sppF
/u6aPgM+mjYDYpo2A2KWMgNmkiJDZo4yA2aOMgNmjiJDZo4yA2aOMgNmjiJDZo4yA2aP/fxcE/38
iC/+GLhX/ITkZ/6FDHPupTCDxsFQj6bVaJ+G5XCzYtlNAz69cU8WpZGS8omxzs5tygKuVeYyij3+W
m4qHnpSGkKSOg5unioKoqYiDtmGhMmohITWqIWD16eFg9ilhYLYpYWC2KWFgtihYLYpYWC2
KWFgtihYLYpYWC2KX/gBcE/4AiC/+lLhT/mDkX/6NDGvmsTB3us1Qf5rIXld6+UyvUu1A+yrRaUc
GtYmk3p2lXrqFwf6WbdoudIXyVIJCDnoyLjKSGiJaogYeqn6lsat8icSqe4nVqnyH1qh+htenfobYpn6
G2KZ+htimfobYpn6G2KZ+htimfobYpn6G2Kb/gRcE/4AiC/+LLhP/mjkV/6ZDGPavTBnrt1Ma479UG
9rFRynPwE49xrlXT7yzX2CyrGZwqadtfaChc4mXm3mUjpaAnYWSH6R9j5Kpel2fq3WOraxzj7+rco/
VqnSN1al2i9and4rXp3eK16d3itend4rXp3eK16d3itend4rXp3eK16f/gRcD/4EiC/+NLhH/nTkT/6IDFf
OzSxXovE8V38ZNFdXLQijLxUw7wb9VTbe5XV6ts2RupK1qe5qob4iRonWTh558nH2Zg6N1lo6oc
JWbq2yVqqtqlruqaZjTqmuU1alukdanb5DWp2+Q1qdvkNanb5DWp2+Q1qdvkNanb5DWp2+Q1qf/
ghYD/4liC/+QLhD/oDkR/KxCEe+3ShHkwkgP29A8FNDQQCbGy0o5u8VSS7G/WlyoumBsnrRmeZ
SvblWKqnGQgKZ4mnaif6FuolmmaJ+XqGWfpqljoLeoYqHO2Se16dnmdimaJfYpmiX2KZol9imaJf
YpmiX2KZol9imaJfYpmiX2Kb/gxYD/4QiCv+ULg7/pDgO+LBBDeq9RgvfzD0l1NoyEsRYPSTA0kc2t
s1PSavIVlqiwl1pl71id423Z4ODs22Oea9zl2+se55nqoiYqqUpF+qpKVe7WkXKzLo16p2qJgo9mj
YaHZo2Gh2aNhodmjYaHZo2Gh2aNhodmjYaHZo2Gh2aP/hBYD/4UiCv+YLGv/qDcL87Y+CeXFO
QXY2yoDzeE7EMTfQiG620Y0r9ZMRqXRULEbzVhmKmdedlbCY4B7vmiKcbpvkmm4d5lit4KdXbeR
n1u4oZ9ZuLKeWLnJnVm33pxbr92eXKzcnlys3J5crNyeXKzcnlys3J5crNyeXKzcnlys3J7/hhYC/4gi
Cf+dLQj9rTQH7Lw3BN3TIADO4zEEexuhEDr3mTB+05FAxquFQQ57dUFST2FRjiNNZch7PXntzy2
OFaslqjWLHc5Jcxn+WWceOI1fln5dVybCWVMnGIVTJ4pRWvuGWV7rhl1e64ZdXuuGXV7rhl1e64
ZdXuuGXV7rhl1e64Zf/iBUC/44iB/+iKgX2sy4C5MgiANDhHAHG7DYHvu1HErfuUR+v7VgupetXP5
noVVCN5FRfgOFVbHXeWXZr3F9/Y9pmhlzZb4pY2XyNVdqLjIPbnl5S3K2NUd3BjFDd3otS0eaMU
s3njVLN541SzeeNUs3njVLN541SzeeNUs3njVLN543/iHUC/5UgBP+pJQHtvB0A0tUJAMfsJgK+8j
wLtvRjF6/1UiWm9FM0m/NTQ5DxUICF71Ndee5UaG7sVnFk61t4Xepifljq4FV63eDUuyGg1HtloN
R7aaCUO62gU/vzIBO6uWBTuXmgk7I5oJO5eaCTuXmgk7I5oJO5eaCTuXmgk7I5oL/jRQB/5waAfi
xQDtxwcAyOoPAL71LwS2+UEPrvtJHKX8TCqb/E04kPpMRoX6TFJ6+U5dcPIQZmb4U21e+Fdy
WPhddIP3ZnlQ9m97TvZ7fEz2iXxL9pd8S/ele0r3tXpJ+Mt5SfjXeEn413hJ+Nd4SfjXeEn413hJ+Nd4
SfjXeEn413j/kBMB96UMANO6BQDHZgYAvfUcALX9NQet/z8TpP9DIJr/RS2P/0Y6hf9GRnr/R1Fw/
0IZZ/9LYV//T2ZZ/1RrVP5ablH8YXBO+2hyS/txc0n7fHRI+4d0R/uUc0f7oXNG+69yRvy1ckb8tXJG/
LVyRvy1ckb8tXJG/LVyRvy1ckb8tXL/mgoA068CAMbBBAC81QYAs/8iAqv/Mwqi/zkWmP88lo7/Pi6

D/z86eP9ARG7/QU1I/0RUXv9HWVn/TF5U/1FhUf9WZE3/XGZL/2JoSP9paUb/cWpF/3pqQ/+DakL
/jmpC/5tpQv+gaUL/oGIC/6BpQv+gaUL/oGIC/6BpQv+gaUL/oGnWpgAAyLgCALzMAwCx6wkAqf8i
A6D/LQyW/zlXjP81loH/Ni12/zg3bP86P2P/PEZd/0BMV/9EUFL/SFRP/01XTP9RWUn/Vlph/1tcRP
9hXUP/Z15B/25eP/91Xz7/fl89/4hfPP+NXzz/jV88/41fPP+NXzz/jV88/41fPP+NXzz/jV/KsQAAvcMC
ALHVAgCm/xAAnf8eBJP/JgyJ/yoWfv8tIHT/Lylq/zExYf80Ofr/OD5U/zxCUP9ARkz/REhK/0hKR/9L
TET/UE5C/1RPQP9ZUD7/XIE9/2RRO/9pUjr/cF14/3hTN/97Uzf/e1M3/3tTN/97Uzf/e1M3/3tTN/97Uz
f/e1O/vAlAss8AAKXdAACa/w4AkP8WB1b/Hgt7/yITcf8IG2b/KCNd/yopVv8vL1D/MzNN/zc3Sf86OU
b/PjtD/0I9Qf9FPz//SUA9/0xBO/9RQjn/VUM4/1IDNv9eRDT/Y0Qz/2pFMv9sRTL/bEUy/2xFMv9sRT
L/bEUy/2xFMv9sRTL/bEW0yAAAp9gAAJr0AACO/wgBgv8NA3f/Ewhs/xgOYv8cFVn/HxtR/yQgTP
8oJEj/LCde/zAqQf80LD//Ny49/zovO/89MDj/QDE3/0QyNf9HmZP/SzQx/080MP9TNS7/VzUt/1w2L
P9eNiz/XjYs/142LP9eNiz/XjYs/142LP9eNiz/Xjb/Zhgl/2AiDv9bLxj/Wzwk/2NDLv9ITDj/Y1VD/15ftf9
Yalb/UnZe/02CY/9JjWf/R5dq/0aha/9Fq mz/RbNs/0S9bP9EyGz/RNVs/0XkbP9G7Wv5SvBr807yau
1R9GrnVPVp4Vf2auFX9mrhV/Zq4Vf2auFX9mrhV/Zq4Vf2auFX9mr/Zhgl/2AiDv9bLxj/Wzwk/2NDL
v9ITDj/Y1VD/15ftf9Yalb/UnZe/02CY/9JjWf/R5dq/0aha/9Fq mz/RbNs/0S9bP9EyGz/RNVs/0XkbP9
G7Wv5SvBr807yau1R9GrnVPVp4Vf2auFX9mrhV/Zq4Vf2auFX9mrhV/Zq4Vf2auFX9mr/Zxgl/2EiD
v9bLxj/Xjsj/2ZCLf9oSzf/ZINC/2FdTP9baFb/VHRe/0+AZP9KjGj/SJZr/0egbP9GqW3/RbJu/0W9bv9
FyG7/RdZu/0bkbF5I7W33TO9s8FDybOpT9GvkVvVr3Vj2bN1Y9mzdWPZs3Vj2bN1Y9mzdWPZs3
Vj2bN1Y9mz/aBgI/2lhDf9dLxj/Yjki/2pALP9tSTb/bFJA/2ZbS/9gZVX/WXBe/1J9Zf9NiWn/SpRt/0ieb/
9HqHD/R7Fw/0i7cP9lxXD+SdJw/ErhcPIL7G/yT+9u61PybuRX9G3eWfVv1lv1b9Zb9W/WW/Vv1lv
1b9Zb9W/WW/Vv1lv1b9Zb9W//aRcH/2MhDf9eLhj/Zjch/24/Kv9ySDT/cVE//21ZSv9mY1T/Xm1d/1d
5Zf9RhmR/TZJu/0yccP9LpXL+S65y/Uu3c/xMwnL5Tc9y907ecvRP6nHtUu9w5lbycN9a83DYW/Rx0
F31c9Bd9XPQXfVz0F31c9Bd9XPQXfVz0F31c9Bd9XP/ahcH/2QhDf9gLhj/ajYg/3M+Kf93RzP/d1A
9/3NYSP9sYFL/ZWtc/112Zf9XgmV+Uo5w/FCZcvpPonT4T6t190+0dfVQvnX0Uct08VLbdO5T6XPo
VvBz4Fvycthc83PRXvR1yl/0dspg9HbKYPR2ymD0dspg9HbKYPR2ymD0dspg9Hb/axcH/2YgDf9h
Lhf/bjQf/3g+J/99RzD/fU86/3pXRf90XID/bGhb/WRyZPpdfmv3Wllx9VWVdPNTnnbxU6h371Oxd+5
Uu3fsVch36lbYduZY53XhW/B12F7xdTbF8nfKYfN5w2LyesNi8nrDYvJ6w2LyesNi8nrDYvJ6w2Lyes
Ni8nr/bBYH/2cgDP9jLrf/czQe/309Jf+CRi7/g043/4BWQv97XU38dGVY+GxvYvRkemvwXoVx7Vq
QdutYmnjpWKR651iteuVZuHrjWsV54VvUeN1c5XjZX/B4z2Hxeshi8XvCZPF9vGXwfrxl8H68ZfB+v
GXwfrxl8H68ZfB+vGXwfrxl8H7/bRYH/2ggDP9nLBb/dzQc/4E9l/+HRiv/iU40/4dVPvuDXEn2fGNV8
nNrYO1sdmnpZYBx5WGLd+JelnrfXaB83V2qfddtH3ZXcF91I7QfdNg4n3PYfB9xmTwfsBm74C6Z
+6BtWjugbVo7oG1aO6BtWjugbVo7oG1aO6BtWjugbVo7oH/bhYG/2kfDP9rKhX/ezMa/4Y9If+MRij/
jk4x/I5UOvaKW0XxhGFQ631oXOZ1c2fhbn1w3GeHd9hjkX3UYZuA0WC1gc9hr4LMYbuCymLJgsh
j24LEZeuCvmftg7hp7ISzauyFrmvrha5r64Wua+uFrmvrha5r64Wua+uFrmvrha5r64X/bxYG/2ofC/9v
KRT/fzMZ/4o9H/+QRSX/k00t+JRUNvGRWkDsjWBL5oZnVuB/cGLZdnlv026CeM5pi3/KZ5WDx2W
fhcRlqYbBZrSHv2bCh71n0oe6aeaHtWvqh7Bs6oirbemlp27piadu6YmnbumJp27piadu6YmnbumJp
27piadu6Yn/cBYG/2sfC/9yKRP/gzMX/448HP+URSL7mE0p9JIUMe2YWjrnlF9F4Y5mUdmGbWD
RfXVuynZ9eMVwhoDAbZCFvWuaibpqplq3aq6LtWu7i7Jry4uwbd+LrG7oi6hv54ykcoEoMoXHnjKFx
54yhceeMoXHnjKFx54yhceeMoXHnjKFx54z/cRYG/2weC/92KRL/hjMW/5E8Gv+YRR/4nEwl8J5T
LOmeWTTjnF4+25RiTtKMal/KhHJsw316eL13goC4c4uHs3CVi7Bvn46tbqmPq2+1kKlwxJCncNiPp
HLlj6Bz5Y+dc+WPmnTlj5p05o+adOaPmnTmj5p05o+adOaPmnTmj5p05o//chUF/20eC/95KBD/iTI
U/5Q8GP6bRBz1oEwh7aNSJ+akWC7eoVk71ZpfTM2SaF3Fi29rvYN2d7Z9foCweYalq3WQjahzm
pGkc6SSonOwk6B0vpOedNGTm3Xjk5h245OWduOSIHbkkpR25JKUduSSIHbkkpR25JKUduSSIH

bkkpR25JL/chUF/24eC/97KA//jDIS/5c8FvyfRBnypEsd6qhRluKqVyjapVU50Z9dSsiXZlu/kG1pt4p0
drCDe4CqfoKlpHuLj6B4IZOcd6CVmXerl5Z3uZeUeMuXknnhlpB54ZaPeeKVjXnill154pSNeeKUjX
nill154pSNeeKUjXnill154pT/cxUF/28eCv9+KA7//jIR/5k7FPmiQxbvqEoa56xQHd+uUSXVqll3zKNb
ScOcZFm7lmtoso9xdKuJeH+khH+lnYCHkjh9kJWTe5uYkHunmY57tJqLe8aaiXzdmYh84JilfOCX
h3vhlod74ZaHe+GWh3vhlod74ZaHe+GWh3vhlod74Zb/dBUF/3AeCv+AKA3/kTIQ/5w6EfaiQhPsq
0kW5LFOGNuySCTRIa1yKdZR7+hYle2m2lmpVvc6aPdX6einyll4WDkJCCjJaLf5aah3+jnIR/sJy
Cf8GcglIDYnICA3puAf9+ZgH7gmIB+4JiAfuCYgH7gmIB+4JiAfuCYgH7gmIB+4Jj/dRUF/3EeCv+DJ
wz/kzEO/586D/SoQRDpr0cR4bZMEta3QyLNsk40w6xXRbqmX1WxoGdkqZttcaGVc32YkHmHklt/k
lmHiJeDhZKbf4SennuErJ55hLyed4XSnXeE3Zx5g96beoLfmXqC35l6gt+ZeolfmXqC35l6gt+Zeol
mXqC35n/dRUF/3leCv+GJwv/ljEM/al5DfCrPw3mtEUN3btDENK7QSDJtkwyv7FVQ7arXVOtpmRi
pKFqcJubcHuTlnaGipF8j4KNhJd7i46cdomannKJqJ9wirifbovNnm+K3Z1xiN6cc4bfmnOG35pzht+a
c4bfmnOG35pzht+ac4bfmnOG35r/dhUE/3MeCf+JJwr/mTAK+qU3Cu2vPQniuUAl2clzDs7APx/Eu
0owurZTQbGxWIGorGFgn6dobpaibXqNnXOEhJh5jnuVgJZzkoqbbpGWnmqRpZ9okbSfZpLJnma
S3Z1pj96ca4zfmmuM35prjN+aa4zfmmuM35prjN+aa4zfmmuM35r/dxUE/3YeCP+MJgj/nC8l9qk1B
+m0OAXewDYD08gvDcnFPB2/wUcutbxQP6y3V0+jsl5ema5kbJCpaneHpG+CfaB1jHSdfJNsm4aZ
Z5qTnGOaoZ1hmrGcX5vGm1+b35phl9+aZJPfmWST35lkk9+ZZJPfmWST35lkk9+ZZJPfmWST35
n/eBUE/3odB/+PJQb/oC0F8q0xBOS6LwLZzB0Bzc4sC8PMORq6yEQrsMNNPKa/VEydultbk7Vha
YqxZnWArWt/dqlxiG2neZBmpYKVKSPmF2knplbpa6YWqXDI1mm3ZZboeGWXZzh12b4Zddm+
GXXZvhl12b4Zddm+GXXZvhl12b4Zf/ehQE/38cBv+UlwT9pCkD7LlpAd3EGQDR0xcAx9UpCb3TN
hiz0EEpqsjJOaDIUEmWw1dYjL9cZoO6YnF5t2d7b7RthGexdYtgsH+PW7CMkliwnJJXsKySVbHA
kVSx2pBWreWQWKbkkVim5JFYpuSRWKbkkVim5JFYpuSRWKbkkVim5JH/fBQD/4UbBP+ZiQL3
qSIB5boWANLNCADJ2hQAwN0uB7fdOBWt2j4lo9ZFNpnSTeApZlJVhpcYYnvGXW1xw2J2aMBpf
mC/cYRbvnuIV76JilS+mYpTv6qKUcC9iVDA2lhRvOqIU7PpilOz6YpTs+mKU7PpilOz6YpTs+mKU
7PpilOz6Yr/fhQD/4sZAv+eGgDvsBIA08IAMnSCADA4ilBueY5BrHmRBKo5EcineJJPLeSkKH20
5QfNhTXXLUWGdp0l5wYdBld1vPbXxWznH/Us+GgFDQl4FP0KiATtG7f03S1X5N0O19TsXugE7F
7oBOxe6ATsXugE7F7oBOxe6ATsXugE7F7oD/gBMC/5IVAPOIDADTtwYAyccFAMDZCQC46isCs
e08CqnuRRWh7Uoil+xNMIzqTj6B6E9LdeVRV2vkVGF41ppW+JhblbhanJS4XV1UOKDdk7jk3ZN
5KR1TOS0dEvlynNK5OVySdvwdUnb8HVJ2/B1SdvwdUnb8HVJ2/B1SdvwdUnb8HX/hRIB/ZoLAN
SsBADJvAQAv84GALftFgCw8i8FqPU8DqD2QhqW9UUnjPRHNILzSUB38kpLbfFNVGPxUVxc8V
ZiVvBdZ1HxZWpO8W9sS/F7bUnyim1l85psSPSpa0f0umpG9NBpRvPtaEbz7mhG8+5oRvPuaEbz
7mhG8+5oRvPuaEbz7mj/jwsA2KQBAMu0AwDAXQAtdUGAK72HgGn+zAHnv04EpX9PR6L/UA
qgf1CNXf9Qz9t/UZIZPxJUFz8TFZW/VJbUf1YXkz9X2FI/mdjRf5xZEP/fWVB/4xkQP+bZD//qmM//7
pjPv/SYj7/02l+/9NiPv/TYj7/02l+/9NiPv/TYj7/02LdmwAAzawAAMG9AgC2zgQAq+0KAKT/IQKc/y0
Kk/8zfIn/Nx9//zkqdf88NGv/Pjxi/0FEW/9ESVT/SE5Q/01SS/9SVUf/WFdD/19YQP9nWj3/cVs7/31b
Ov+KWzn/l1s4/6VaN/+1Wjf/tlo3/7ZaN/+2Wjf/tlo3/7ZaN/+2Wjf/tlrQpQAawRYBALfHAgCr1glAov8R
AJj/HwOQ/ygLh/8tFX3/MB9z/zMoaf82MGD/OTdZ/zw8Uv8/QU7/Q0RJ/0dHRf9MSkL/UUs//1dNPP
9eTjr/Zk84/29QNV95UDT/g1Az/49QMv+dUDL/nVAy/51QMv+dUDL/nVAy/51QMv+dUDL/nVDEsA
AAucEBAKzRAACg4QAAlv8QAYz/GQSD/yELef8mE2//KRtm/ywjXf8vKVb/My9Q/zYzS/86N0b/PTI
D/0E8QP9GPj3/Sj87/1BAOP9VQjb/W0l0/2JDMv9qRDD/ckQu/3tELP+HRCz/h0Qs/4dELP+HRCz/
h0Qs/4dELP+HRCz/h0S6uwAArSSAAKHZAACU9QEAIv8LAX7/EgR1/xgJa/8dD2L/IRZZ/yUcUv8ol
Uz/LCVH/y8pQ/8zKz//Ny09/zsvOv8+MTf/Qjl1/0czM/9LNDD/UTUu/1c2LP9dNiv/Yzcp/2o3J/9zNyf/
czcn/3M3J/9zNyf/czcn/3M3J/9zNyf/czewxgAAo9QAAJTeAACI+gAAfP8EAXH/CgNm/w8HXP8TC1

P/GA9N/xwUR/8gF0L/JBs+/ycdOv8rHzj/LyE1/zliM/81IzD/OSQu/zwLP9AJir/RSco/0koJ/9PKCX/V
Ckk/1oplv9hKiL/YSoi/2EqLv9hKiL/YSoi/2EqLv9hKiL/YSr/VxsK/1AID/9ILxX/TDgd/1NAJv9VSjD/VFQ
5/05fQv9Jakr/RHRQ/z+BVP88jlf/O5IZ/zujWf87rFr/O7Va/zy+Wv88yFr/PdNa/z7gWf8+61n/P/VZ/U
H7WPdE/VfyR/5X7Uv/VuIM/1fpTP9X6Uz/V+IM/1fpTP9X6Uz/V+IM/1f/WBsK/1EkD/9ILhX/Tjcd/1U/
Jv9YSC//VIM5/1FeQv9LaUr/RnNQ/0F/Vf8+jFj/PZha/z2iW/89qlv/PbNb/z69W/8+xlV/P9Jb/z/fW/9A6I
r+QfRa+0L7WfVG/VjwSf5Y6kz/WOZN/1nmTf9Z5k3/WeZN/1nmTf9Z5k3/WeZN/1n/WRoJ/1IkD/9J
LhX/UjYc/1o9Jf9dRi7/W1A4/1dbQf9QZ0r/SnJR/0Z8Vv9CiVn/QJRb/0CfXf9AqF3/QLBd/0G6Xf9Bw
13/Qs5d/kPcXfE6Fz6RPNc90b7W/FJ/VrrTf5a5U/+W+BQ/lzgUP5c4FD+XOBQ/lzgUP5c4FD+XO
BQ/lz/WhoJ/1MkD/9LLhX/VjQb/148JP9iRS3/YE83/1xZQP9WZEn/T3BR/0p6V/9Ghlv/RJFd/0OcXv
9DpV//RK5f/0S3X/5FwF/8Rctf+kbZX/hH5I71SPFd8kn6XexN/FzmUP1d31L9X9pT//aU/5f2IP+X9p
T//aU/5f2IP+X9pT//WxkJ/1QjDv9NLRT/Wjla/2M7I/9nRCz/Zk41/2JXP/9cYkn/VW1R/094V/9Lglz/
SI5f/0eZYP1HomH8R6ti+ki0YvllvWH4Schh9UrWYfJL5GDvTPBf7U36X+ZR/F/fU/xh2FT9YtNW/W
PTVv1j01b9Y9NW/WPTVv1j01b9Y9NW/WP/XBkJ/1YjDv9RLBP/XzAZ/2k6lf9tQyr/bUwz/2IVPf9iX0
f/XGIQ/1V1V/1QgF37TYph+UyVY/dLn2T1S6hk9EywZPJNumTxTcVk707TY+xP4mLoUO9i5IL5Yd
5U+2PWVvtlz1j8ZstZ/GfLWfxny1n8Z8tZ/GfLWfxny1n8Z8tZ/Gf/XRkJ/1ciDv9VKhL/ZC8Y/285H/9z
Qif/dEsx/3BTO/9qW0X9Y2ZP+IxxV/ZXfF70U4Zi8VGRZe9Qm2btUKRn7FCtZ+pRt2foUsJm51PQZ
eRU4GXgVu1k3Vb4ZdRY+mfnWvtpx1v7asNc+2vDXPtrw1z7a8Nc+2vDXPtrw1z7a8Nc+2v/Xxgl/1
kiDv9aKBH/ai8W/3Q4Hf95QSX/ekkt/3hRN/xyWUL3a2JM82RtVu9eeF3sWYNj6VaNZ+dVI2nkVaF
q4lWqauFWtGnfV79p3VfNadpY3mnWWuxp0lr3aspb+mzEXfptvl75brtf+W+7X/lvu1/5b7tf+W+7X/lv
u1/5b7tf+W//YBgl/1ohDf9eJRD/by4V/3o3G/9/QCL/gUgq/H9QM/d7Vz7xdF9l7WxpU+hldFzkYH5j4
FyJaN1ak2vaWZ1t2FmmbtVZr27TWrpU0VrHbs9b2G7LXelux172b8Ff+HG7YPdytmL3crNj9nOzY/
Zzs2P2c7Nj9nOzY/Zzs2P2c7Nj9nOzY/YRcl/1shDf9jJA//dC4T/383GP+FPx//h0cm+lZPL/GDVjnsfV1E
5nZmT+FvcFnbaHpj1mKEatJfjm7PXpdxzF2hcsdpqnPHXrRzxV7AdMNf0HTAYeJzvWLxdLhj9XW
yZPV2rmX0d6tm9HerZvR3q2b0d6tm9HerZvR3q2b0d6tm9Hf/Yhcl/1wgDf9oJA3/eC0R/4M2Fv+KP
hv7jEYi841NKuyLVDPmhlS+4H9kSdl3bFfSb3VjzWl/a8hmiHHEY5J0wWKbd79ipXi8Yq54umO6eb
hijHi2ZNt4s2bseK9m83mqapJ6p2nyeqRp8nqkafJ6pGnyeqRp8nqkafJ6pGnyeqRp8nr/YxYH/14g
Df9slwz/fCwP/4c1E/+OPhj3kkUe75NMJeiSUy3hjlK32oZhR9J+aVXLd3JixHF6a79sg3K7aYx3uGe
VerVnn3yyZ6l9sGe0fa5own2saNN9qWnnfaZq8H6ia/B+n2zwp1t8H6dbfB+nW3wfp1t8H6dbfB+n
W3wfp1t8H7/ZBYH/18gDP9vlwv/gCwO/4s1Ef2SPRXzlkQa65hLIOSYUSfclFY01IxeRMuFZ1PEfm
5gvXh2a7dyfnOzb4d5r2yQfaxrmn+pa6SBp2uvgaVsvlKjbM2BoG3hgZ1u7oKab+6CmG/ugZZv7oG
Wb+6Blm/ugZZv7oGWb+6Blm/ugZZv7oH/ZRYH/2AfDP9ylgr/gysM/440D/qWPBLwm0MW6J5JGu
CeTyHYmVMypJcQsaLZFG+hGxft35zarF5enOrdYJ6p3KLf6NwIYKgb5+Enm+qhZtwtoaZcMeGI3
HchZVx64WTcuyFkXLshJBy7ISQcuyEkHLshJBy7ISQcuyEkHLshJBy7IT/ZhUH/2leC/91Ign/hisL/5
EzDfeZOg/tn0ES5KNHFdyiSB/SnVEwypdaQMGQYk+5impdsoRwaat/d3Olen57oHeHgZt1klWYc5
qHIXOmiZNzsomRc8KJjnTWiY106YmLdOqlnTqh4l064aJdOuGiXTrho1064aJdOuGiXTrho1064b/Z
hUH/2UdC/94lQj/iSoJ/5QyC/SdOQzqoz8O4adEENimQx7Ook8uxZxYPr2VYU20kGhbrYpuaKaFd
HKfgHt6mXyDgZR5jlaQd5aKjXeijlp2royld72MhnfRjIR46lyDeOiKg3fpiYN36YiDd+mIlg3fpilN36YiD
d+mIlg3fpilN36Yj/ZxUG/2ccCv97lQf/iykl/pcxCfGgNwnnpzwK3qw+DdOqQRzKpk0swaBWPLiaX0y
wlWZaql9sZqGKcnGZhXh6k4F/go1+h4ilfJKMhHudjoF6qo9/e7mPfXvMj3t85l58e+eNfHvoi3x66lp8
euiKfHroinx66lp8euiKfHroinx66lr/aBUG/2obCf9+1Ab/jigG+5ovB+6jNAfjqzgh2rA0DM+uPxrGqksqv
aVUOrSfXUqsmmRYpJVqZZyQcHCUi3V5jYd8goaDhliAgY2NfH+ZkHl/ppF2f7WRdIDIkXOA4JB0f
+aOdX7njXV+54x1fueMdX7njHV+54x1fueMdX7njHV+54z/aRQG/24bCP+BIAX/kScF+J0tBeunMA

TgrzED1bQwCsuyPRnCrkkoualSOLCkWkiooGFWn5toY5eWbW6PkXN4h415gX+Jglh5homOdIW
VkXCFopJuhbGSblbEkmqG3JFrheWPbYPmj6C541ugueNboLnjW6C541ugueNboLnjW6C543/a
hQG/3EaB/+EHwT/ICUD9KApA+erKgLctScB0bguCce2Oxe9skYmtK5QNqyqWEWjpV9UmqFIYZK
ca2yJmHB3gZN2gHiQfYdxjYaNblyRkWiMn5JmjK6SZl3AkWKN2ZBjjOWPZonmjmeH541nh+eNZ
4fnjWeH541nh+eNZ4fnjWeH543/axMG/3UaBv+HHQP/lylC8KQkAeKwIADWuxkAzL0rCMK7ORW
5uEQksLRNNKewVUOerFxrLadiXoyjaGqDn210eptzfXKYeYVrloKLZZWOj2GUnJBflauQXZW9j1y
W1Y5clOeNXpDnjWCO54xgjeMYI7njGCO54xgjeMYI7njGCO54z/bBMF/3kZBf+LHAL8mx4A66
gbANy3DADQwBUAxsMoBrzBNhOzvKEiqrKMaG3UkCYs1hOj69eW4arZGd9p2lxdKRvemuhdoJk
n3+HX56Li1uemYxZn6mMWJ+6i1ag0olWn+mJWJrpiVqX6Ylal+mJWpfpivqX6Ylal+mJWpfpivqX6
Yn/bhMF/34YA/+QQQH3nxYA3K4JANK7CADJxhAAwMkkBLbJMhCuxj0fpcNGLpu/Tj2Su1VLibdb
WH+0YGN2sGVtba5rdmWrcnxfqnyBWqmlhFeql4VVqqeFU6q4hFKrz4NSquiCUqXshFSh7IRUoe
yEVKHshFSh7IRUoeYEVKHshFSh7IT/cBIF/4MWAv+VFADfpQgA0rIGAMnABQDBzQoAuNEgA7
DRLg6nzzkcns1CK5TJSjmLxVBHgcFWVHi+XF9vvGFoZrlncF+3b3ZZtnh6VbaFfVO2IH1Rt6R9T7
e2fE64zXtOt+d6TbLxfE+u8H1PrvB9T67wfU+u8H1PrvB9T67wfU+u8H3/cxIE/4oSAPGbCgDTqgU
AyrCFAMHFBQC40wgAsdsdAancLAug2jYYltg+J43URTWD0UxCec5RTnDLV1lNyV1iYMdjaVnGa
25UxnVxUcaCc0/GknRNx6JzTMe0ckvly3FLx+VwScT3ckq+9nRKvvZ0Sr72dEq+9nRKvvZ0Sr72d
Eq+9nT/fBAC/5ELANaiAgDLsAMAwr4DALjMBgCv3QsAquQnAqPmNwmb5j4UkeRClobhRTB730
g9cd1NSGjbUIJg2VhaWdhfYFTYaGRQ2HJnTdiAaUvZj2IK2qBoSdqxZ0jbx2ZH2uFIRtjyZkXT+Wh
F0/loRdP5aEXT+WhF0/loRdP5aEXT+Wj/hQsA3JkAAM6pAgDDtwIAucUDAK/TBgCo6RcAoe0rBJ
rvNg2S7jwYiO5BJH7sRC9060c6aupKQ2HpTktZ6VRSVOlbV1DpZFPm6W5cSul6XUnqiV1I65ldR
+ypXEfsultG7M9ZRevpWUHo91tB6PdbQej3W0Ho91tB6PdbQej3W0Ho91vyjwAA0aEAAMWxAQ
C6vwlAsM0EAKbaBgCf8x0BmPYrB5D3MxCH+Dgbff8JXP3Py9q9kM4YfZGQFn2SkZU9IBLT/Z
WT0v2XVJH92ZURPdxVUL4fVVA+lxVQPmcVT/5q1Q/+rxTPvrQUj365FE9+uRRPfrkUT365FE9+u
RRPfrkUT365FWHmgAAx6sAALy5AQcxyAlAptUDAJz4DwCU/B0Cjf8oCYT/LxJ7/zMccf83JWj/Oi
1g/z40WP9BOIP/Rj9O/0tCSf9QRUX/V0hC/15KP/9nSzz/cUw6/31MOP+MTDf/m0w3/6ILNv+4SjX/x
0o1/8dKNf/HSjX/x0o1/8dKNf/HSjX/x0rKpQAavbQAALPDAACn0QAAM9wBAJL/EQCJ/xoDgP8jCn
f/KRJy/y0aZf8xl3/NChW/zguUf88Mkv/QDZH/0U5Q/9KO0D/Tz08/1Y/Of9dQDb/ZUE0/29BMf97Qi/
/iElu/5RCLf+iQS3/rkEt/65BLf+uQS3/rkEt/65BLf+uQS3/rkG/rwAAtL4AAKjMAACb2AAAJ+8CAIb/Dg
F8/xYEc/8dCWrlhBh/ycWWf8qHFP/LiFN/zImSP82KUT/OixA/z4uPf9DMDn/RzE2/0wzM/9TNDD/
WjUt/2I1K/9rNij/dTYn/382Jf+MNI/ITYk/5U2JP+VNI/ITYk/5U2JP+VNI/ITa2uQAAqsgAAJ3UAA
CP3QAAg/gAAHn/CQFv/xAEZf8VB1z/GgxU/x4QTv8jFUj/JxID/yscQP8uHzz/MiE5/zYInF85JDL/PS
Uv/0ImLP9HJyr/Tsgn/1MpJf9bKSL/Yyog/2sqHv90Kh3/fCod/3wqHf98Kh3/fCod/3wqHf98Kh3/fCqs
xAAAantAAAJDaAACC4QAAdv0AAGv/AgFh/wgEWP8NB1D/EgpJ/xYNQ/8aDz7/HhI6/yITNv8IFTP/
KRYw/ywXLf8vGCr/Mhko/zYaJf86GyP/Pxsg/0QcHv9KHRz/UR0a/1ceGP9fHhf/ZB4X/2QeF/9kHhf/
ZB4X/2QeF/9kHhf/ZB7/Sh0L/0ImEP85MhX/PzQX/0U7Hv9IRSf/R1Av/0NcN/8+aD3/O3RC/zeBRf8
2jUf/NZhl/zaiSf82qUn/NrJJ/ze6Sf83w0n/OM1J/zjaSP855kj/OvBH/jv4R/w7/0b6PP9G9j//RfBB/0fuQ
v9H7kL/R+5C/0fuQv9H7kL/R+5C/0f/Sh0L/0MmEP87MBT/QjIW/0k5Hv9MRSf/S1Av/0daN/9DZj7/
PnJD/zT+Rv85ikn/OZVK/zmfS/85p0v/Oa9L/zq4S/86wUv/O8tK/zvXSv485Er8Pe5J+j73Sfg+/0j2P/9
H8UL/SOtE/0nqRf9K6kX/SupF/0rqRf9K6kX/SupF/0r/Sx0L/0QmEP8+LhT/RjEW/005Hf9QRCb/UE
8v/0xZN/9HZD7/QnBE/z98SP88iEr/PJNM/zydTP88pU3/PK1N/z21Tf89vkz/PshM/T7UTPs/4kv4Q
O1L9kH2SvNB/kryQ/9J7UX/S+ZH/0zIR/9N5Uf/TeVH/03IR/9N5Uf/TeVH/03/TBwL/0UIEP9CLRP/Si
8V/1I4Hf9WQiX/VU0u/1FYNV9MYj7/R21E/0N5Sf9AhUz/P5BN/z+aTv8/ok//P6tP/UCzT/xAu077Qc

VO+ULRTvZD3030Q+tN8UT1TO5F/kvtRv9M50j/TuBK/0/fSv9Q30r/UN9K/1DfSv9Q30r/UN9K/1D/
TRwK/0YID/9GKxP/Ti0U/1g3G/9cQST/W0os/1hVNf9SYD7/TWpF/0h1Sv9FgU3+Q4xP/EOWUfpD
n1H5Q6hR90SwUfZEUfH1RcJQ80bOUPFH3U/uSOpP6kj1TuhK/U7mSf9Q4Ev/UthN/1PWTv9T1
k7/U9ZO/1PWTv9T1k7/U9ZO/1P/TxwK/0ckD/9KKBL/VCwT/141Gv9jPyL/Ykgq/19SM/9ZXTz/U2h
E/E5ySvILfU/3SIhR9UiTU/NlnFTySKRU8EitVO9JtVPtSr9T7EvLUuIM2ILmTOhR4070UOBN/VPdT
f9V1U//Vs5R/1fNUf9YzVH/WM1R/1jNUf9YzVH/WM1R/1j/UBSk/0kkD/9PJxD/WysS/2U0GP9qPR//
akYo/2dQMf5hWTr6W2RD9IVvSvJReVDwToRT7U2OVetNmFbpTaFX6E2qVuZOslbkT7xV41DJV
eFR2FTdUudU2ILzVdZS/FJSUv9Zy1P/W8VV/1zDVf9cw1X/XMNV/1zDVf9cw1X/XMNV/1z/URsK/0
ojD/9UJA//YSoQ/2wzFv9xPBz/ckQk/W9NLfhqVjzY2BB711rSetXdVDnVH9V5FKKWOJSIFngUp1
Z3IKmWdxSr1raU7la2FPGWtZT1FrSvEvaz1byWstW+1zHVv9ewVj/X7tZ/2C6Wf9guln/YLpZ/2C6
Wf9guln/YLpZ/2D/UxoJ/0wjDv9Zlg7/ZykO/3lyE/93Ohn/eUMh+HdLKfJzUzPta1w952VnR+NfcU/e
W3tV21iFWthWj13VVple0laiX9BWql/OVRgzFe/YMpYzWDHWd9gxFruYMBa+mK8Wv9jt1z/ZLJd/
2WxXf9IsV3/ZbFd/2WxXf9IsV3/ZbFd/2X/VBoJ/08iDv9flgz/bCgM/3cxEf99ORb7f0Ed839JJex7US7
mdVk44G9kQ9pobU3VYndW0F6AXM1cimDKW5Njx1qcZMVbpWXDW65lwVu5Zb5cx2W8XdhluV
7pZbZe9mayXv1ormD9aKph/GmpYfxpqWH8aalh/GmpYfxpqWH8aalh/Gn/VRkJ/1MgDf9jIqV/cScL
/3wvDv+DNxP2hj8Y7oZHH+eDTijgf1cy2XdgQNJwaUzMaXJXx2V7XsNihGO/YI5mvV+XaLpfoGq4
X6lqtmCzarRgwGuyYdBrr2Lja6xi8mupYvpspWT6baJI+m2hZfptoWX6baFI+m2hZfptoWX6baFI+m
3/VhkJ/1YeDP9oIar/diYJ/4EuDPyINg/yiz0U6oxFGuKLTCLbhlQu0n5dPct3ZkvEcW5Wv2x3X7pof2
W2Zohps2SRbLBkmm6uZKRvrGSub6pkum+oZclwpmXdcKNm7nCGZ/hwnWj4cJpo+HCZaPhwm
Wj4cJlo+HCZaPhwmWj4cJlo+HD/VxgJ/1kdC/9sIAn/eiUI/4UtCfiMNAzukDsQ5pJCFd6RSB3Vi1Es
zYVbO8V+Y0m+eGtVuHNzX7Jue2aua4NrQ2mMb6holXKlaJ9zo2ipdKFotXSeacN0nGnWdJpq6nS
YavZ0lWr2dJN9nSSa/Zzkmv2c5Jr9nOSa/Zzkmv2c5Jr9nP/WBgl/1wbCv9vHwf/fSQG/4grB/WQM
gnrITkM4pg/ENmWRBvQkU8qyltYOcCEYUe4fmhTsnlvXqx0d2ancX9to26HcZ9tkHWcbJp2mmyle
JhssHiVbL54k2zReJfT5niPbfR4jm30d4xt9HeMbfR2jG30doxt9HaMbfR2jG30doxt9Hb/WRgl/18aCf
9yHgb/gCMF/ospBfKUMAfnmTYI3p07C9SaQRnMIU0ow5BXN7uKX0WzhGZSrH9tXaZ6dGahdntt
nHODc5hxjHeUcJZ5km+ge49vrHyNb7p8i3DMfllw4nyHcPJ7hnDyeoVw83mFcPN5hXDzeYVw83
mFcPN5hXDzeYVw83n/Whcl/2IZCP91HgX/gylE+44nBO6XLQTknTIF2qEzCtCePxfHmksmv5RV
NbaPXUOvimRQqlVqW6GAcWWbfHdtlXh/dJB1h3mNdJF8iXOcfodyqH+Ec7V/gnPHf4Bz3n9/dPB
+f3PxfX5z8Xt+c/J7fnPye35z8nt+c/J7fnPye35z8nv/Whcl/2YYB/94HQT/hiED+JEIAuuaKQLgoiwC1
qQwCcyiPRXDnkkku5ITM7OUW0Grj2JOo4poWpyGbmSVgXRtj317dlp6g3qFel1+gXeYgH52pIJ8
drKCenfDgnh32oF3eO6Ad3fvf3h28H14dvB9eHbwfXh28H14dvB9eHbwfXh28H3/Wxcl/2kYBv97H
AT/iR8C9ZQiAeieJAHdpiUB0qguB8imOxS/okcit55Rma+ZWT+nIGBMn5BmWJeLbGOQhnJsiYJ4
dIN/gHp9fYI/eXuUgnZ7oIRze66EcXu/hG981YNvfOuCcHvugHF6739xeu9+cXrvfnF6735xeu9+cXr
vfnF6737/XBYI/2wXBv9+GwP/jB0B8pceAOShHQDZqhsAzqssBsSqORK7pkUgs6JPL6ueVz2imV
5KmpVkvPORamGLjG9rg4h1c3yFfHp2goWAcYGGq26AnYVrgauFaYG7hWeB0YRngemDalDug
Wp+74Bqfu9/an7vf2p+739qfu9/an7vf2p+73//XRYH/28XBf+BGQL+jxoA7poXAN+IEgDTrRcAya8p
BcCuNx3q0Mer6dNLKajVTqen1xllptiVI6XaF+Gk21pfo9zcnaMeXlviYF/aoiMg2aHmYVjh6iFYyi4
hGCIzoNfiOaCYIbugWKE7oBjg++AY4PvgGOD74Bjg++AY4PvgGOD74D/XhUH/3MWA/+FFwH6k
hYA6Z8OANaqCADOsRMAxLMmBLuyNA+ysEAacq1KKqGpUjiZplIFkaFfUoidZV2AmpmpneJZwch
CTdndpkX59Y5CJgV+PlOndj6WDW5C1glqQy4FZkOSAWY7vgFyL739ciu9/Xlrvf1yK739ciu9/Xlrvf
1yK73//YRQG/3gVAv+JFAD2lg8A16MHAM+tBwDltQ4Av7gjA7a4Mgyttj0ZpbNHJ5ywTzWUrFZCi6
hcToOIYlp6omdkcp9tbGqcc3Rjmnx5XZmGfVqZIH9XmaN/VpmzflSayH1UmeJ8U5jxfFWT8XxWkv

F8VpLxfFaS8XxWkvF8VpLxfFaS8Xz/ZhEF/30SAf+NDwDamwUA0acFAMmxBQDAuwkAuL8fAbC
/LgqovToXn7pDJJa3SzKOTfI/hbBYS3ytXlZ0q2Rfa6hqaGSmcG9dpXI0WKSEd1WkknITpKF5UaS
xeFCkxndQpOB1TqPydk+e9HdQnfR4UJ30eFCd9HhQnfR4UJ30eFCd9Hj/bA8D/4IPAO2SBwDTo
AQAYqsEAMG2AgC5wQQAscYaAarHKgihxjYUmcM/IZDARy6HvU47frpURnW3WIFttV9aZbNmYI
6xbWhYsHZtVLCBb1Gwj3FPsJ9wTrCvcEywxG5MsN5tS6/xbkqq+HBLqfhwS6n4cEup+HBLqfhwS
6n4cEup+HD/dA0C/4gJANIYAQDMpQIAw7ECALq8AQcXxwUAqc8TAKLQJQWa0DEQks47HYjL
Qyp/yEo2dsZQQW7DVUtlwVtUXsBiW1i+aWBTvnJkUL5+Zk2+jWdLvp1nSr+tZkm/wWVlv9xkSL3
wZEW6/mdFuP5nRbj+Z0W4/mdFuP5nRbj+Z0W4/mf/fAkA418AANCfAADFqwEAu7cAALLCAwCp
zQYAOngLAJrcIQOS3C0Nito2GYDYPIv31UUwbtNKO2bRUERe0FZMWM9dUIPOZldPzm9aS857
W0nPilxlz5pcR9CrW0bQv1pF0NpZRc7tWELM+1tCy/5cQsv+XELL/lxCy/5cQsv+XELL/lz8hAEA1Z
cAAMimAAC9sgAAtL4BAKrKBACg1AcAmeMXAJPMJwSM5zILg+Y5FXrPiBw5EMqZuNHNF/hTT
xY4VNDU+BaSE7gYktL4GxOSeF4T0fhHlBG4pZPReOmTkTjuE1D485MQuLnTEDf904/3/pPP9/6
Tz/f+k8/3/pPP9/6Tz/f+k/cjgAAy6AAAL+uAAC2ugAAq8YCAKDRBACW4AgAkO4cAYnwJwaB8TA
OePE1F2/wOiFm8D4pXu9DMFvSDdR7008Te9UQEnvXENG8GVFQ/BvRkHxfEZA8YtGP/KbRj/y
q0U+8rxEPfPQQzzy6kl88e9CPPhvQjzx70l88e9CPPhvQjzx70LQmQAAwqkAALe2AACttwAAoc
4BAJbYAgCN9Q4AhfgcAn76JQh1+ywPbfsxF2T7NR9c+zomVvs+K1D7QzBL+0g0R/xON0P8VTIA
/F07Pf1mPTv9cD05/n0+N/6MPjf/nD02/6s9Nf+5PDT/zzs0/9Q7NP/UOzT/1Ds0/9Q7NP/UOzT/1Dv
EpAAAubIAAK6/AACjywAAItQAAIrdAACc/xAAev8ZA3H/lAhpyYOYf8rFVn/MBtT/zQgTv84JUUn/P
ShF/0lrQf9HLT3/TS86/1QxN/9bMjX/ZDMy/280L/97NC3/iTQt/5c0LP+IMyz/tTMs/7kzLP+5Myz/uTM
s/7kzLP+5Myz/uTO7rgAAsLsAAKTIAACX0gAAitoAAH7wAAB2/wwBbf8VBGT/Gghc/x8MVf8kEU//
KRVJ/y0ZRf8yHEH/Nh89/zohOv8/lzf/RCU0/0kmMf9QJy7/Vygr/2ApKf9qKSb/dSkk/4lql/+OKSL/my
ki/58plv+fKSL/nyki/58plv+fKSL/nymyuAAAsUAAJjPAACL2AAAFd8AAHL4AABp/wcBYP8OBFf/F
AdQ/xkKSv8dDUT/IRBA/yUSPP8pFDj/LRU1/zEXMv81GC//ORks/z4aKf9DGyf/SRwk/1Edlf9ZHh//
Yh4d/2weG/91Hhn/gR4Z/4QeGf+EHhn/hB4Z/4QeGf+EHhn/hB6owgAAms0AAIzVAAB93QAAb+Q
AAGX+AABc/wABU/8GA0z/CwZF/xEIP/8VCzr/GQ02/x0OMv8hEC//JBES/ycSKf8qEib/LRMj/zEUlf
81FB7/OhQc/0AVGf9HFRf/ThUV/1YVE/9eFRH/ZxUR/2oVEf9qFRH/ahUR/2oVEf9qFRH/ahX/PCA
M/zQpEP8yMRP/NzMV/zk5GP88Qx//PE8m/zlbLP82ZzH/M3Q1/zGAN/8wjDj/MZY5/zGfOv8xpzr/M
q46/zK2Ov8zvjn/M8c5/zTSOf813zj+Neo4/Db0N/o3/Df4N/829jj/NvU4/zfyOf848jn/OPI5/zjyOf848jn/
OPI5/zj/PR8M/zUpEP80LxL/OTEU/z03F/9AQh7/QE0m/z1ZLf86ZTL/N3I2/zV+OP8zjir/NJQ7/zSdO
/80pTv/Naw7/zW0O/82vDv/NsU6/zfQOv033Tr7OOk5+TnzOfY6+zj0Ov848zv/OPI6/zruO/877jv/O+
47/zvuO/877jv/O+47/zv/Ph8M/zYoEP83LRL/PS8U/0E2F/9FQB7/REsm/0JXLf8+YzL/O283/zh7Ov
83hZv/N5E8/zebPf83oz3/OKo9/ziyPf85ujz+OcM8/DrOPPo72zv3O+c79TzyOvI9+jrwPv857z3/O+0
9/zzpPv8+6T7/Puk+/z7pPv8+6T7/Puk+/z7/Px8M/zcoEP87KxH/QS0T/0c0Fv9KPh3/Skkl/0dVLf9D
YDP/QGw4/z14O/87gz3/O44+/zuYP/07oD/8O6g/+zyvP/o9tz74PcA+9z7LPvU+2D3yP+U970DwP
OxB+jvrQf896UD/PuhA/0DiQv9B4kL/QeJC/0HiQv9B4kL/QeJC/0H/QB4M/zgoEP8/KBH/RyoS/00y
Ff9RPRz/UUck/05SLP9KXTP/RWk5/0J0Pf1AgD/6P4pB+T+UQfc/nUH2QKVB9ECsQfNBtEHYqB1
A8ULIQO9D1T/sROM/6ETvPuZG+T7jRf9B4UT/Q99E/0TaRv9F2kb/RdpG/0XaRv9F2kb/RdpG/0X
/Qh4L/zonEP9EJhD/TSkR/1QxE/9YOxr/WEUi/1VQKv9RWjL8TGU5+UhwPvZGe0HzRIZD8USQR
PBEmUTuRaFE7UWpROtGsUPqRrpD6EfFQudl0kLkSeJB4EruQd1K+EPaSF9F2Ej/R9RJ/0jPSv9
Jz0r/Sc9K/0nPSv9Jz0r/Sc9K/0n/Qx0L/z4ID/9Jlw7/VCgP/1wwEv9gORj/YEig/11MKPpYVzD1U2E4
8k9sPu5Md0LrSoJF6UmMRudJIUfIsp5H5EqmRuJLrkgbTLdF30zCRd1M0EXaTeBF1k/tRdJO+Ej
PTf9KzE3/TMIN/03ET/9OxE//TsRP/07ET/9OxE//TsRP/07/RBwL/0liDv9OIQ3/WycN/2MuD/9oNxX/

aUAc+mZJJfRhUi3uW1426lZoPeZSc0PiUH1H4E+HSd1PkUnbTppK2U6iS9dOq0vVT7NL00++S9
FQy0vPUdtLy1LqS8dS9k3EUv9PwIH/Ub5S/1K5U/9SuVP/UrIT/1K5U/9SuVP/UrIT/1L/RhwL/0cgDf
9VIAz/YiYM/2osDf9vNRL7cT0Y9G9GIO1qTynnZFkz4l9lO91ab0PZVnhl1VSCTNJTio7PU5RQzV
OdUMtTpVHJU65Rx1S4UcVUxFHDVdRRwFblUr1W81O5Vv1Vt1b/VrNX/1awWP9XsFj/V7BY/1e
wWP9XsFj/V7BY/1f/RxsK/0seDP9aHwr/ZyQK/3ArC/92Mg/2eDsU7ndDG+dzTCThblYu2mhgOdR
hakPPXXRky1p9T8dYhIPFWl9VwleXVsBXoFe+WKIXvFiyV7pZvle4Wc1XtVrfV7Jb71ivWvtarVr/W
6pb/1unXP9bp1z/W6dc/1unXP9bp1z/W6dc/1v/SBsK/08cCv9gHgn/bSMI/3YpCPt8MAvxfzgQ6X9A
FuJ8SB7ad1Mq0nBdN8xpZkLGZG9LwWB4Ub5egVa7XYlZuFySW7Zcm1y0XKRcslytXbBduF2uX
cZdq17ZXalf6l2mX/depF//X6Bg/1+eYf9fnmH/X55h/1+eYf9fnmH/X55h/1//SRsK/1MaCf9kHQf/cSIG
/3onBveBLgJthTUM5lY9EdyDRRnTfk8nzHdZncVxYkG/a2tLuWdzUrVkfFiyYoRcr2GNX6xhlmCqY
Z9hqGGpYqZhs2KkYcFiomLSYp9j5WKdY/Rjm2P/Y5hj/2OWZP9jlmT/Y5Zk/2OWZP9jlmT/Y5Zk/2
P/ShsK/1YYCP9oHAb/dSEF/38lBPOFKwXpjiI4lw5DNaJQRfOhE0lxn5XMr94Xz+4cmhKsm5vU65
qd1mqah9epmaIYqRlkWShZZpln2WkZp1lrmeaZbtmGbmZ5Zm4WeUZvFnkmb9Z5Bm/mePZv5
mj2b+Zo9m/maPZv5mj2b+Zo9m/mb/SxoK/1oXB/9sGwX/eR8D+4MjA++JKAPljy4F3JEzCdKOPxX
JiUsjwYRUMLI+XT2yeWVlRHRsUqdwC1qjbXtgn2uDZJtqjGeZaZVplmifapRoqmuSaLdrkGjHa45p3
WuMae9rimn7a4lp/GqlafxpiGn8aYhp/GmlafxpiGn8aYhp/Gn/TBoK/10WB/9vGgT/fB4C+IYhAuUNJ
AHhkykC15UwB82SPRPEjkkhvllSLrWEWzuuf2JHp3ppUaF2cFqcc3dgmHB/ZpRuh2mRbJFsjmyb
boxrpm+Ja7Nvh2zDb4Vs2G+DbOxvgmz6boFs+m2BbPtsgWz7bIFs+2yBbPtsgWz7bIFs+2z/TRoJ/
2EVBv9yGQP/fxwB9YkeAeiRIADdlylA05kuBsmWOxLakkcful5RLLGJWtmphGBFon9nUJx7bVm
Wd3RgkXR7Zo1yg2uJclxvhm+XcYNvo3KBb69yf2+/cn1v03J7b+lxe3D4cHpv+W96b/luem/5bnpv+
W56b/luem/5bnpv+W7/TxgJ/2QVBf91GAL/ghoB8owaAOSUGgDZmxsAz5wrBcWaORC8l0UdtJJP
Kq2OVzeliV5DnoVITpeBa1iRfHFgi3l4Z4Z2f2yCdIhxfnOTc3tzn3V5c6t1d3O7dXVzz3Vzc+Z0c3P2c
3Nz93F0cvhwdHL4cHRy+HB0cvhwdHL4cHRy+HD/Uhcl/2cUBP94FwH/hRcA748VAOCXEgDUnh
cAyyqApBMGeNw65m0MbsZdNKKmTVjWhjl1BmopjTJOGaFaMgm5fhn51Z4B7fG17eYRyd3iPdX
N3m3dxd6h3bne3d213y3Zrd+N2a3j0dGx39nNtdvdybXb3cm1293JtdvdybXb3cm1293L/VRUH/2o
TA/97FQH8lBQA65IPANibCQDQoRQAxqMnA72iNQ21n0EZrZxLJqWYVDODlFs/lpBhS46LZlWHh
2xegIRyZnqBeW10f4Fyb32Ldmt8l3hpfKV4Z3y0eGV9x3djfeB3ZH3ydWV89nRmevdzZnr3c2Z693
NmevdzZnr3c2Z693P/WBMG/24SAv9+EwD4iw4A2pUHANKeBwDLpBEAwqckArmmMwuxpD8Xq
aFJJKGdUTGZmVk9kZVfSlqRZFOCjimpceopvZXOHdmxthX5yalSIIdmSDIHhHg6J4X4OxeF6DxHd
dg912XIPwdV6B9nRfgPZzX4D2c1+A9nNfgPZzX4D2c1+A9nP/XBAF/3lRAf+BDwDnjggA1JkGAM
2hBgDFqA0AvashAbSrMAmsqTwVpKZGIZyJTy6Un1Y6jJxcRoWYYIB9lWdadZJtYm2Pc2pnjXtwY
YuFdF6LkXZbi592WYuudliLwXVXi9t0Vorvc1eJ93NYhvdYwIb3cliG93JYhvdYwIb3cliG93L/YA0E/3
YOAfqFCgDZkgMAz50EAMelBAC/rAgAt68eAa+wLQiorjkSn6xDH5epTCuPpIM3h6NZQ3+fX013n
GRXb5pqX2iXcWZhlnlsXJSCcFiUj3JWIJ1yVJStclOUwHFSINlwUZPtcFCS+HBSj/hwUo/4cFKP+H
BSj/hwUo/4cFKP+HD/ZgwD/3oLAOSKAgDTlglAyaEDAMGqAgC4sQMAsbUaAKq2KgaitTYQmrN
AHJKwSCiJrVA0gapWP3moXElxpWFSaaNnWmKhbmFcoHZnV5+AalSejWxRnptsUJ+rbE6fvmpN
n9ZpTZ7saUuc+mpMmfrTJn7a0yZ+2tMmfrTJn7a0yZ+2v/bAoB/X8GANiPAADNmwEAw6YBALq
uAACxtgAAqrsUAKO9JQScvTINILs8Glu4RSSDtKwwerNSOnKxWERqr11NY61kVFysaltXqnNfU6
p9YICqimROqplkTKqpY0uqvGJKqtRhSanrYUen+WNHpf9kR6X/ZEel/2RHpf9kR6X/ZEel/2T/cwcA
4oYAANGVAADGoQAAvKsAALS0AACqVAEaOsMMAJvGIAKUxi0KjMQ3FYTCQCB7wEcrc75NN
Wu8Uz5jullGXblgTVe4Z1NSt29XTrZ6WUy2h1pKt5daSbenWke3uVhGt9FXR7bpV0S0+FICs/9bQr
P/W0Kz/1tCs/9bQrP/W0Kz/1v/ewAA2l0AAMqbAAC/pwAAtrEAAKy6AACjwwMAmMwGAJLQGQG

M0ScHhNAyEHZPOhtzzUlla8tIL2PJTjdcyFU/VsdbRVHGY0INxmxMSsZ3TkjGhU9HxpRPRcelTkt
Ht01Dx85MQ8boS0HE900/wv9PP8L/Tz/C/08/wv9PP8L/Tz/C/0/ggwAAz5UAAMKjAAC4rgAAr7gA
AKTBAQCaygQAKNMHAIIncFACD3iQEe94uDHPdNhZq2z0fYtpDKFvZSS9V2FA2UNhXO0zXXz9I
12IBRth0Q0TYgUND2ZFDQtmIQkHatEFA2spAQNnjPz7W8kE81PxDPNT8QzzU/EM81PxDPNT8
QzzU/EPUjgAAxp4AALurAACxtgAApsAAAjvJAQCQ0QMAhtoHAIHoGgF66SUFcuotDGrpNBRi6T
kcW+g/l1ToRCIP6EsuSuhSMUfoWjRE6GM3QuluOEDpejg/6ok4P+uZNz7rqjc967s2POvRNTvr6D
Q55/c1Oef3NTnn9zU55/c1Oef3NTnn9zXKmAaAvacAALOzAACovgAAncAAJHPAACF2AIAffAM
AHbzGgJv9CMGZ/UpDF/1LxNZ9TUZUvU6Hk31PyNI9UUnRPZLKkH2Uiw+9lsuPPdkLzn3bzA3+
HwwNviLMDb5mzA1+aovNPm8LjP5zS4y+eltMvniLTL54i0y+eltMvniLTL54i2/owAAtbAAAKq7AAC
exgAAks4AAIXVAAB53gAAcvsNAGv9GANj/h8HW/8IC1X/KhBP/y8VSv80GUX/ORxB/z4fPv9DITv/
SiM4/1EkNf9ZJjL/Yicw/20nLv96Jyz/iScs/5gnK/+nJyr/tCYq/8MmKv/DJir/wyYq/8MmKv/DJir/wya3r
QAARLkAAKDEAACTzAAAhtQAAHjcAABu7QAAZv8LAV//EwNX/xkGUf8fCkv/JA1G/ygQQf8tEz3/
MRU6/zUWNv86GDP/QBkw/0YaLv9MHCv/VBwo/14dJv9oHiT/dR4i/4Melf+QHiH/nB0g/6cdIP+nH
SD/px0g/6cdIP+nHSD/px2utgAAosIAAJXLAACH0gAAeNoAAGvhAABh9wAAWv8FAVP/DANM/xl
GRv8YCED/HAs8/yANOP8IDjT/KRAx/ywRLv8wEiv/NBMo/zkTJf8/FCL/RRQg/00VHf9WFRv/YBU
Z/2sVF/93FBb/ghQV/4wUff+MFBX/jBQV/4wUff+MFBX/jBSlwAAAI8kAAIjRAAB52QAAa98AAF3
mAABU/gAATv8AAUf/BANA/woFO/8PBzb/Fagx/xgKLv8cCyv/lAwo/yMNJf8mDiL/KQ4f/ywPHP8x
Dxn/NhAX/zwQFP9DEBH/TBAO/1UQDP9eEAr/ZxAJ/28PCf9vDwn/bw8J/28PCf9vDwn/bw//LyMN/
yctD/8qLxH/LjIS/y43Ff8wQhj/L04d/y5blv8sZyX/K3Qo/ymAKv8qiyv/KpQr/yudK/8rpSv/LKsr/yyyK/8t
uiv/Lclq/y7MKv8u2Sr+L+Up+y/vKfkW+Cj3Mf8o9TH/KPQx/ynzMf8q8zH/K/Mx/yvzMf8r8zH/K/Mx/yv
/MCIM/ygtD/8sLRD/MDAS/zM2Ff80QBj/NEwd/zJYlv8wZSb/LnEp/y19K/8tiSz/LZls/y6bLf8uoy3/L6
os/y+wLP8wuCz/MMAs/zHKK/0x1iv6MuMr+DPuKvYz9yrzNP8p8jT/KvE0/yvwNP8t7zT/Le80/y3vN
P8t7zT/Le80/y3/MSIM/yksD/8wKxD/NC0S/zg0FP86Phj/OUod/zhWl/81Yif/M28q/zF6LP8xhi7/MZA
u/zKYLv8yoC7/M6cu/jOuLv00ti78NL4t+zXILfk10y32NuEs9DfsLPE39ivuOP4r7Tj/Lew3/y7qN/8v6jf
/MOo3/zDqN/8w6jf/MOo3/zD/MilM/ywqD/80KA//OSoR/z4yE/9APBf/QEcD/z5TI/87Xyj/OGss/zd3Lv
82gi/+Noww/DaVMPs3nTD5N6Uw+DisMPc4szD2Obsv9TnFL/M60S7xO98u7jvrLes89S3oPP4u5z
z/MOU7/zLkO/8z4zv/M+M7/zPjO/8z4zv/M+M7/zP/MyEM/zAnDv85JQ//QCgQ/0UwEv9HORb/R0Q
c/0RPI/9BWyn/Pmct/DxzMPk7fjL3O4gy9juSM/Q8mjPzPKlz8T2pMvA9sDLvPrgy7T7CMew/zjHqQ
Nww5kDqMONB9TDhQP0y3kD/NNxA/zbbP/832j//N9o//zfaP/832j//N9o//zf/NSAM/zUkDv8+lg7/Ry
cO/00uEf9PNxT/T0Eb/0xMlv1JVyJ5RWMu9UJuMfJBeTTwQIQ17kCONexBljXqQZ416UKmNedCr
TTmQ7U05US/M+NEyzPhRdoy3UbpMtpG9DXWRf031ET/OdFE/zrPRP88z0T/PM9E/zzPRP88z0
T/PM9E/zz/NiAM/zohDf9DHwz/TiUN/1UtD/9XNRL/Vz4Y/FRIIPZQUyfxTF8t7UlqMupHdTXnRn835
UaJOONGkjjhR5s430ejN95IqjfcSLI32ki8N9IlyTjWSNg40krnOM9K8zrLSvw8yEn/PsZJ/z/ESf9Ax
En/QcRJ/0HESf9BxEn/QcRJ/0H/OB8M/z4fDP9KHgv/VSQL/10rDP9gMhD9YDsV9I1FHe9YTyXqVF
os5VBIMuFOcDfeTHs520uEO9hLjjzWS5Y91EuePdJLpj3QTK4+zky3PsxMwj7KTdE+x07iPsRP8D
/ATvtCvk7/Q7tO/0S6Tf9FuU7/RbIO/0W5Tv9FuU7/RbIO/0X/OR8M/0McC/9QHqN/XCIJ/2QpCv9n
MA33aDgR72ZBGehhSyHiXVcp3VliMdhUazjTUnU80FB/P81QiEHLT5BCyFCYQ8dQoEPFUKIEw
1CxRMFRvES/UcpEvFLbRLIT60W2U/hHs1L/SLFS/0mvUv9KrIP/Sq5T/0quU/9KrIP/Sq5T/0r/Oh4L/
0cZCf9WHAj/YiEH/2onB/tvLQnxcDQN6W89FOJrRxzbZ1MI1GFdMM5cZznJWHA/xVZ5Q8JVgkBA
VlpHvISTSLxUm0m6VaRKuFWsSrZVtkq0VsNKslbUSq9X5kusV/RMqVf/TadX/06lV/9OpFj/TqRY/0
6kWP9OpFj/TqRY/07/PR0L/0sXCP9cGgb/aB8F/3AkBfZ1KgbsdzEJ43c5Dtt0RBfTb04jzGIZL8ZjYz
jAX2xAvFx0RblbfUm2WoVMtFmNTbFZlk6vWZ5PrImoT6xasVCqWr5QqFrOUKVb4VCiW/BRoFz8

Up5c/1KcXP9Sm1z/Uptc/1KbXP9Sm1z/Uptc/1L/QBsK/08VB/9hGQX/bR0E/nUiA/F7JwTnfS0G3n4
0CdV7QBTNdkshxXBWLB9rXze5ZmhAtGNwR7BgeEytX4BPq16IUahekVOMXZpUpF6jVaJerVW
gXrlVnl/IVZtf3FaZX+1Wl1/6V5Vf/1eUX/9Wk1//VpNf/1aTX/9Wk1//VpNf/1b/QxkJ/1QUBv9IGAT/cR
wC+nofAu1/lwLigygd2YQwB8+BPRLHfEkfv3dTK7lyXDaybWQ/rWlsR6lmc02lZhtRomODVaBijFe
dYpVYm2KeWZliqVqXYrRaIWLDWpNi1lqQYupbj2L4W41i/1qMYv9ai2L/Woti/1qLYv9ai2L/Woti/1r/
Rhcl/1gTBf9pFgP/dRoB9n0cAemDHgDehyIB1lguBsqGOxDCgkYcun1QKbN4WTStc2E+p29oR6J
scE6eaXdTmmh/V5hmm1qVZZBck2WaXZBlpV6OZbBfjGW/X4pl0l+lZedfhmX2XoVI/16EZf9dhGX/
XYRI/12EZf9dhGX/XYRI/13/SRUI/1sSBP9sFQL/eBcA8oAYAOWGGADZixsAz40RbcaLOQ+9h0Q
atoJOJ69+VzKoeV89onVmRp1ybE6YbnNUIGx7WZBqg1yNaYxfi2iWYYhooWKGaK1ihGi7Y4Joz
WOAaORjf2j0Yn5o/2F9aP9gfWj/X31o/199aP9ffWj/X31o/1//SxQH/18RA/9vFAH/ehQA74MTAOGJ
EQDVjhGAY5ApBMKPNw26jEIYsodMJauDVTCKf1w7nXtjRZd3aU2Sc3BUjnB3Woluf16Gblhig2yS
ZIBrnWV+a6lmfGu3ZnlryWZ4a+Bmd2vyZHZs/mN2a/9idmv/YnZr/2J2a/9idmv/YnZr/2L/ThMG/2lQ
A/9yEgD8fREA64YNANmMCQDQkhUAx5QnA76TNQu2kEEXroxLI6elUy6ghFo5mYBhQ5N8Z0y
NeG1UiHV0WoNye19/clRje2+OZnhvmmh2b6ZpdG+0aXFvxmhwB91ob2/wZ29v/GVvb/9kb2//ZG9
v/2Rvb/9kb2//ZG9v/2T/UREF/2UPAv90EAD3gAwA2ogGANOQBwDMIRIAw5clArqXMwqyID8Vq5
BJlaONUSyciVg3IYVfQo6BZUulfWtTgnpxWn13eGB4dYBlDHSKaHFzImpuc6NrbHOwa2pzwmpoc
9lqaHPuaGh0+2doc/5maXP/ZWlz/2Vpc/9laXP/ZWlz/2X/VA4F/2gOaf93DQDoggcA1owFAM+TBg
DHmA8AvpsiAbabMQivmD0Tp5VHH5+STyqYjlc1kYpdQlqGY0mDg2hSfYBuWXD9dWBxe31lbXm
HaWl5kmtmeJ9sZHitbGJ5v2thedVrYHnraWB4+Wwhd/5nYnf+ZmJ3/mZid/5mYnf+ZmJ3/mb/VwwE
/2wMAfx6CQDbhQMA0Y8EAMqWBADCnAsAup8gAbKfLgernToRo5pFHJuXTiiUk1UzjZBbPYWM
YUd+iWZQd4ZsWHGDcl9rgXplZoCDaWJ/j2tff5xsXX+rbFt/vGtaf9JqWn/qaVp++Ghbff5nW3z+Z1t8
/mdbfP5nW3z+Z1t8/mf/WwoD/28KAO59BADWiQIAZMDAMWaAwC9oAcAtaMcAK6jLAWmojgP
n6BCGpedSyWPmVMwiJZZO4CTX0V5kGROco1pVmuLcF1liXdiYleAZ1yHjGIzhppqV4apalaGum
IUh9BoVIboZ1OF92dUhP5mVIP+ZISD/mZUg/5mVIP+ZISD/mb/YAgC/3MHAN2BAADRjQEAYJcC
AL+eAQC3pAMAsKcYAKmoKASipzUNmqU/F5KjSCKKoFAtg51WOHuaXEF0l2JKbJVnUmaTbVlf
kXVfWpB+Y1aPimVUj5hmUo+nZIGPuGVQj85jT4/mY06O9mNOjP9jTov/Y06L/2NOi/9jTov/Y06L/2
P/ZQcB93cCANiGAADMkgAAwpsAALqjAACxqQAAqawTAKOuJQKcrjEKIKw8FI2pRR+Fp0wqfaV
TNHWiWT1uoF9GZ55kTWCCa1Ram3JZVpp8XVKZiF9QmZZgTpmlX02Ztl5MmcxdS5nlXEqX9V5l
lv9eSJX/X0iV/19lIf9fSJX/X0iV/1//awMA4H0AANGMAADGlwAAvaEAALSoAACqrgAAorlNAJy0IAG
WtC0lJrM4EYaxQRt+r0kmdq1PL2+rVThnqVtBYahhSFumaE5VpXBSUaR5V6kVdMpJRYs6Wk
V0mltFZlpcpVSKTkVEai81ZEof9XRKD/WESg/1hEoP9YRKD/WESg/1j/cgAA2YMAAMuSAADAnQ
AA6YAAK6tAACktAAAmrkGAJS8GgCOvSkFh7w0Dn+6PRd3uUQhb7dLKmi1UTNhtFc6W7JdQF
WxZEZRswxKTbB2TEqwg05IsJFOR7GhTkaxsk1FscdLRLDiSkOu8kxBrP9OQKz/TkCs/05ArP9O
QKz/TkCs/07ieQAA0YsAAMSyAAC6owAAsawAAKezAACdugAAksADAlrFEgCFxyIDfscuCnfGNx
NvxD8cZ8NGJGDCTCxawFlzVb9ZOFC/YT1MvmlASb5zQka+gENFv49DQ7+fQ0K/sEJBv8VBQb/
gQEC98UA+u/5DPbr/RD26/0Q9uv9EPbr/RD26/0TZgwAAyJMAAL2gAACzqgAAqrMAAJ+6AACV
wQEAisgEAH/QCAB60xoBdNMnBm3TMA5m0jkWX9FAHlnQRyRTz00qTs9UL0rPXDJHzmU1RM
9wN0LPfDhBz4s4QNCcNz/QrTY90MI1PdDdND3O7jQ6zPo3Osv9NzrL/Tc6y/03Osv9NzrL/TfOjQA
AwJwAALaoAACtsQAAoroAAJbBAACLyAAAAGM8EAHXyCABw4BgAa+EkBGTiLQId4TMQV+E6FI
HhQRxN4EghSeBQJUXgWCdD4WEqQOFrKz/hdys+4oYrPeOWKzzjpyo747kpOuTQKDni5yg33/
UqNt/5Kzbf+Ss23/krNt/5Kzbf+SvElwAAuKUAAK+wAACkuQAAMMEAAIzlAACAzgAAddYCAGzq
CgBn7BkBYe4iBVruKQpU7i8OTu81E0nvOxdF70EbQe9IHj7vTyA88FciOvBhlzjxbCQ28XkkNfKIJ

DTzmSMz86kiMvO7IjHzziEx8+YgMPLrIDDy6yAw8usgMPLrIDDy6yC7ogAAsa4AAKa4AACawQA
AjccAAIDOAAB01QAAad0AAGL2DABc+BcCVvkeBVD6JAIL+yoMRvsvD0H7NBI+/DkUOvw/Fjf8R
hg1/U0ZMv1VGjD+Xxst/mocLP94HCr/hxwq/5cbKf+nGyj/tRon/8gaJ//PGif/zxon/88aJ//PGif/zxqzqw
AAqbYAAJzAAACPxwAAAgc4AAHTUAABn3AAAXegAAff/CQBS/xICTP8ZBUb/HgdB/yMKPf8oDD
n/LQ42/zEPMv82ES//OxIs/0ETKv9IEyf/UBQl/1oUlv9mFCD/cxQf/4lTHv+REx7/nhMd/6wTHf+wEh
3/sBld/7ASHf+wEh3/sBKrtQAAAn78AAJHGAACDzQAAAdNQAAGfbAABa4gAAUvQAAEz/AwFG/ws
CQf8RBDz/FgY3/xsINP8gCTD/JAot/ygMKf8rDCb/Lw0j/zQOIP86Dh7/QQ8b/0kPGP9SDxb/Xg8T/2
oPEf93DhH/gw4Q/5AOEP+UDhD/IA4Q/5QOEP+UDhD/IA6hvQAAk8YAAITNAAB11AAAZ9sAAFn
hAABO6AAARvwAAEH/AAE7/wMCNv8IAzH/DQUt/xlGKf8WByb/Gggj/x4JIP8hCR3/JAoa/ygKF/8s
CxT/MQsR/zgLDv8/Cwv/SAwl/1MLBf9dCwP/ZwsB/3MLAf92CwH/dgsB/3YLAf92CwH/dgv/licM/xst
Dv8hLQ//ly8Q/yM1Ev8kQBT/JE0W/yNaF/8iZxr/lXMc/yF/Hf8iih7/lpMe/yObHv8koh7/JKge/yWvHv8l
th7/Jr0d/ybGHf8n0R3+J94c+yjqHPko9Bv3Kfwb9Cn/G/Mp/xzyKf8d8in/Hvlp/x7yKf8e8in/Hvlp/x7/ly
YM/x4sDv8ILA7/Jy4Q/yg0Ev8pPxT/KUsW/ydYGP8mZRv/JXEd/yV9H/8lhx//JpAf/yeZH/8noB//KKY
f/yitH/8ptB//Kbsf/irEHvwqzx76K9we+CvpHfUs8x3yLfsc8C3/He8t/x/uLP8g7Sz/le0s/yHtLP8h7Sz/le
0s/yH/JCYM/yMqDv8oKQ7/KysP/y8zEf8wPRT/L0gX/y1VGf8sYR3/K24f/yp5IP8qhCH/K40h/yuWlF
8snSH9LKQh/C2rlfstsSD6Lrkg+S7CIPgvzB/1MNof8zDnH/Ax8R7tMvse6zH/IOox/yLoMf8j5zH/JOc
x/yTnMf8k5zH/JOcx/yT/JiUM/ycoDf8tJg3/MSgP/zYwEf84OhT/NkUX/zRRGv8yXh7/MWoh/zB1lv0v
gCP7Mloj+jCTI/gxmiP3MqEj9jKol/UzryL0M7Yi8zS/lvE0yiHvNdch7DbllOk28CDmN/oh5Db/l+I2/yX
hNv8m3zb/J982/yffNv8n3zb/J982/yf/JyUM/yskDf8ylg3/OSYN/z4uD/8/NxP/PkIW/zxNGv85Wh/8N2
Yi+TZxJPc1fCX0NoYl8zaPjvE3lyXwN54l7jilJe05rCXsObMk6jq8JOk6xyPoO9Qj5DvjluE87yPePP
kl2zv/KNg7/ynWO/8q1Tv/K9U7/yvVO/8r1Tv/K9U7/yv/KCQM/zAhDP84Hwv/QSQM/0YsDv9INBH/
Rj4V/0NJGvIBVR/1PmEj8T1sJu88dyfsPll06jyLKOG9kyjnPZso5T6iJ+Q+qSfiP7Em4UC5Jt9AxCXe
QNII2kHiJtZB7ifTQfkqz0D/LM1A/y7LQP8vykD/MMpA/zDKQP8wykD/MMpA/zD/KiMM/zUeC/8+HA
r/SCIK/04pDP9QMQ//TzoT+EtEGPJlUB7tRVwk6UNoJ+ZCcynjQn0q4EKGKt5DjyrdQ5cq20OfK9l
DpivYQ64r1US2K9NEwSvSRM4rz0XeLMtG7CzHRvgvxEX/McJF/zPARf80v0X/NL9F/zS/Rf80v0X/
NL9F/zT/LiAL/zobCv9FGwn/UCel/1YnCF9ZLwz4WDcQ8VRAFupQSxzlTVcj4EtjKNxJbivZR3gt1Ue
BL9NHijDRR5lwz0eaMc1loTHLSKkxykixMchJuzHGSccyxErYMsBL6DK9S/U0ukr/NrdK/zi2Sv85tE
r/ObRK/zm0Sv85tEr/ObRK/zn/Mx0K/z8XCP9MGQf/Vx8G/14IB/xhLAnyYDMN6l48EuNaRxndV1M
g1lJeKNFPaC3OTnlxyk17M8hMhDXGTlw2xEyUNsJNnDfATaQ3vk2sN7xNtTi6TsE4uE7QOLVP4
ziyT/E6r0/8PK1P/z2sT/89qk//PqpP/z6qT/8+qk//PqpP/z7/NxsK/0MUB/9SGAX/XR0E/2QiBfZoKAbs
aS8J42c3DdxkQxTUX08fzVpaKMhXZC/EVG00wFJ1N71Sfjq7UYy7uVGOPldRlj21UZ49s1KnPb
FSsD6wUrs+rIPKpqtT3T6oVO0/plT5QaNU/0KhVP9CoFT/QqBU/0KgVP9CoFT/QqBU/0L/OxgJ/0g
SBv9YFgT/YxsD/WofAvBuJAPmcCoF3W8zCNRsPxLMZ0sdxWJWJ8BeYC+7W2g2t1hxOrRXeT2
xVoE/r1aJQa1WkUKrVplDqVaiQ6hXq0OmV7ZEo1fERKFY1kSfWOIFnFj2RppY/0aZWP9GmFj/Rp
hY/0aYWP9GmFj/RphY/0b/PhYl/00RBf9dFQP/aBgB+G8cAetzIAHhdiUC13YvBs5zPBDGb0gbv2p
SjblIXC+0YWQ2r19sPKxddECpW3xDpluERaRajEaiWpRHoFudSJ5bp0mcW7JJmlu/SZhb0UqW
W+VKk1v0SpJb/0uQW/9KkFv/SpBb/0qQW/9KkFv/SpBb/0r/QRMH/1EQBP9hEwL/bBUA9HMXAO
Z3GQDbeh0A0XssBch5OQ7AdUUZuXFPJLNsWC2taGE2qGVoPaRicEKHYXdGnmB/SJtfh0qZX5
BMI16ZTZVeo02TXq5OkV67To9ezE6NXuFPi17yT4le/U6lXv9OiF7/Tohe/06lXv9OiF7/Tohe/07/RB
EG/1UOA/9IEQH/bxIA8HYSAOF6EgDWfhkAzlApBMN/Nwy7e0lXtHdMlq5yVSyobl41omtlPJ5obE
OaZnNHI2R7S5Rig06RYotPj2GVUY1hn1KLYapSiWG3U4ZhyFOEYd5Tg2HvU4Jh/FKBYf9RgGH/
UYBh/1GAYf9RgGH/UYBh/1H/Rw8F/1kNAv9oDwD9cg4A6nkLANI9CQDRghYAx4QnA7+DNAu3g

EAVsHxKIKl4UyqjdFs0nXBIPJhtaUOUanBlkGh3TY1mf1CKZYdTh2SRVIVkm1aDZKdWgGS0V35
kxFd8ZNpXe2TtVnpk+IV5ZP9UeWT/U3lk/1N5ZP9TeWT/U3lk/1P/SQ0F/1wMAf9qDADzdAoA2ns
GANSBBwDMhhMAw4glArulMgmzhT4TrIFIHqV9USieeVgymHZfO5NyZkKObmxlimxzTYZqe1KC
aYNVgGiNV31nmFI7Z6RaeGewWnZnwVp0Z9Zac2frWXNo+VhyaP9Xcmj/VnJo/1ZyaP9Wcmj/Vn
Jo/1b/TAwE/18LAf9tCgDldwUA1n8FAM+FBgDliRAAv4wiAbeLMAiviTwRqIZGHKGCTyaaf1YwlHp
dOY53Y0GJc2llhHBwTn9ud1N8bX9XeGyJWnVrIFxza6BccWutXW5rvV1sa9JcbGvoW2tr91pra/9Z
a2v/WGtr/1hra/9Ya2v/WGtr/1j/TwkD/2lJAPlvBgDbegIA0oIEAMulBADDjA0Au48gAbOPLgasjToQp
YpEGp2HTSSWg1QukH9bOlP8YUCEeWdHfnZtTnlzdFN1cXxYcXCFW25wkV5rb51faW+qX2dvul
5lb89eZG/mXWRv9lXkb/9aZG//WWRv/1lkb/9ZZG//WWRv/1n/UgcC/2UHAO5yAgDYfQEAzoYCA
MaMAwC/kAoAt5MdALCTLAwokjgOoY9CGJqMSyKtIFlsjIVZNoWBXz5/fmVGeXtrTXN5cVNud3l
YanaCXGd1jV9kdZpgYnWoYGB0t19edcxfXXTkXl109FxddP9bXnT/Wl50/1pedP9aXnT/Wl50/1r/V
gYC/2gFAN91AADTgQAAYokBAMKPAQC6kwYAs5YbAKyXKgSlljYMnZRAFpaRSSCPJlAqilpXNI
GHXTx6hGJFdIJoTG5/blJofXZYY3x/XGB7il5de5dgW3ulYFI7tF9Ye8leV3rhXVd681xXev5bV3n/Wl
d5/1pXef9aV3n/Wld5/1r/WgUB/GwBANt5AADPhQAAXY0AAL2TAAC1lwlArpsXAKecJwOgmzMK
mZk+E5KWRx2Kk04ng5BVMXyOWzp1i2BCb4lmSmiGbFBihXNWxoN8WlqCh11XgpReVYKjXIS
Csl1SgsZcUYLfw1GB8ltRgP1aUYD/WIGA/1pRgP9aUYD/WIGA/1r/XwMA7m8AANZ+AADKiQAA
wZEAALiYAAcwnAAAqJ8SAKKhlwGboTAlIj87EY2cRBuGmkwkfpdTLneVWTdwkl4/aZBkRmOO
ak1djXFSWlt6VIWLhVISi5JaUlUhWk+LsFIOi8RYTYreV0yJ8FdLiPxXS4f/V0uH/1dLh/9XS4f/V0uH/
1f/ZAAA33QAANCDAADFjgAAvJYAALOCaACqoQAAoQNAJynHwGWpi0Gj6U4DoejQReAoUkh
eZ9QKnGdVjNrm1w6ZJliQV6XaEhZlm9NVJV4UVGUg1NOIJBUTJSfU0uUr1JKIMNRSZTdUEiT8
FFHkfxSRpD/UkaQ/1JGkP9SRpD/UkaQ/1L8aQAA2XsAAMulAADAkwAAt5sAAK6iAACkpwAAm6
oHAJWsGgCPrSkEiKw0C4GrPRR6qUUDcqMjMumUi5kpFg1XqNePFmhZUFUoGxGUJ91SUyfg
UtKn45MSZ+eS0efrktGn8FJRp/bSEWd7klDnPtLQpv/TEKb/0xCm/9MQpv/TEKb/0zjcAAA0oEAAM
WPAAC7mQAAsqEAAKioAACerQAak7EAAlyZFACHtCQCgbQvCHqzORByskEZA7BIIWSvTihel
QvWK1bNVOsYjpPq2k+S6tzQEmqfkJHq4xDRaubQkSrrEFDq79AQqvZP0Kp7T9Ap/pBPqb/Qz6m
/0M+pv9DPqb/Qz6m/0PbeQAAYokAAL6WAAC1oAAArKgAAKKuAACXswAAjLcAAIK7CwB+vh0A
eL4qBXG9MwxqvDwUY7tDG126SSJXuVAoUrIWLWU64XjFKuGY0R7dvN0S3ezhDulk4QbiZOECA
qTc/uLw2PrjVNT627DQ8tPg3O7P/ODuz/zg7s/84O7P/ODuz/zjRggAAw5EAALidAACvpwAApa4A
AJq0AACPuQAAL4AAHjEBAByyBQAbcoiAmfKLAhhyTUOW8k9FVXIRBtQx0ogTMdRJEjHWSH
FxmIqQsZsLEDHdy0/x4YtPsiWLTzIpyw7yLkrOsjSKjvG6ik5xPcrN8L/LTfC/y03wv8tN8L/LTfC/y3ljA
AAu5oAALKIAACorgAAAnrUAAJK6AACGvwAAesUAAG/MBABk1AkAYNgYAFzYJARX2S0JUtg1D
k3YPBNJ2EQXRdhLG0LYUx0/2FwfPdhnlTvZcyE62YEiOdqRITjboiA327QfNtvLHzbZ5B401/EfMtX
6ITLV+iEy1fohMtX6ITLV+iG/lwAAtKMAAKutAACgtQAAILsAAIjBAAB7xgAAb8wAAGTTAwBa3Qc
AWOUXAFPMlQNO5ykGSecvCkXoNg5B6DwRPuhDEzvoSxU56VQXN+leGDXqaRg06nYZM+uF
GDLslhgx7KcXMOy5Fy/szxYu7OUVLenzFS3p8xUt6fMVLenzFS3p8xW3oQAARawAAKO1AACWv
AAaiclAAHzHAABvzQAAY9QAANbAABT8QkAT/IVAUr0HQNF9CQGQfUpCD31Lws69jUNNVY7
DjP3QRAX90kRLvhREiz4Wxlq+WcSKPI1Eif6hBlN+pURJvumESX7thAk+8gQl/vaECP72hAj+9oQl/
vaECP72hCwqgAapbQAAJm8AACLwgAAfcgAAHDOAABj1AAAV9sAAE3hAABJ/AcARf8QAUD/F
wM8/x0FOP8iBjX/Jwgx/ywJLv8xCiv/Ngso/zwMjF9DDSL/TA0g/1YOHv9iDRz/cA0a/4ANGv+QDBn/
nwwZ/6wLGP+4Cxi/uAsY/7gLGP+4Cxi/uAunswAAm7wAAI3CAAB/yAAAcM4AAGLVAABW3AAA
S+IAAEPuAAA/wEAOv8JATb/EAlY/xUELv8aBSv/HgYo/yMHJP8mByH/Kgge/y8JG/81CRn/PAoW
/0QKE/9OChD/WwoN/2gKC/92CQv/hAkK/5EICv+cCAr/nAgK/5wICv+cCAr/nAieuwAAkMIAAIHJA

AByzwAAY9UAAFxdAABK4wAAQOgAADn4AAA0/wAAMP8BASv/BwIn/wwDJP8RBCH/FQQe/xg
FG/8cBRj/HwYV/yMGEv8nBw7/LAcL/zMHCP87BwX/RQcB/1EHAP9dBwD/aAcA/3MHAP99BgD/f
QYA/30GAP99BgD/fQb/FSkM/xQrDP8YKw3/GS0N/xgzD/8ZPxD/GUwR/xhZEv8YZhL/GHMR/xh+
Ev8ZiBP/GpET/xuYE/8bnxP/HKUS/xyrEv8dsRL/HbgS/x7BEv8eyxH9H9gR+x/kEfgg7xD2IPgQ8yH
/EPIh/xHxlf8S8CH/E/Ah/xTwl8U8CH/FPAh/xT/FikM/xgpDP8cKQz/HSsN/x4yD/8fPRD/Hkks/x5W
E/8dYxP/HXAT/x17FP8ehRT/H44U/x+WFP8gnRT/IKMU/yGpFP8hrxP+lrYT/SK/E/wjyBP5I9US9y
TiEvQk7hLxJfcR7yX/Eu0l/xTsJf8V6yX/Fusl/xbrJf8W6yX/Fusl/xb/FygM/xwnDP8hJgz/ligN/yYwDv8
mOhD/JUYS/yRTE/8jYBT/lmwU/yJ4F8jghb/JIsW/iSTFvwlmb7JaEV+ianFfkmrRX4J7QV9ye8FP
YoxhT0KdlU8SngE+4q7BPrKvcT6Sr/Fecq/xbmKv8Y5Cv/GeMr/xnjK/8Z4yv/GeMr/xn/GSgM/yEIC/
8mlwv/KSYM/y4vDv8vOBD/LUMS/ytPFP8qXBX/KWgW/Sl0F/spfhj5KocY9yqQGPYrlxf0K54X8yyk
F/lsqxfxLbEW8C25Fu8uwxbtL88V6i/eFecw6xXkMPYW4TD/GN8w/xrcMP8b2jH/HNox/xzaMf8c2jH
/HNox/xz/HiUL/yYhC/8rHwv/MiQL/zYsDf84NQ//NkAS/zRLFP0xWBb5MGQY9jBvGfMwehrxMIMa7
zGMGu4xlBrsMpsZ6zOhGeozqBnoNK8Y5zS3GOY1wBjkNc0X4jbcF9436hfaNvUa1zb+HNm3/x7
RNv8f0Db/IM82/yDPNv8gzzb/IM82/yD/liML/yodCv8xGwn/OilK/z8pC/9AMQ7/PzwR+zxHFPY5Uxf
xOF8Z7jdqG+s3dRzoN38c5jilHOQ4kBzjOZcb4TmeG+A6pRveO6wa3Tu0Gts7vhrZO8ob1zvaG9
M96RzPPPQeyzz+lcg8/yPHPP8kxTv/JcU7/yXFO/8lXtv/JcU7/yX/Jx8K/zAaCf85GQj/Qh8l/0cmCf9
JLgv7RzcP9ERCE+5BTRfpP1oa5D5lHeE+cB7ePnoe3D6DH9o+ix/YPpMf1j6aINQ/oSDSP6gg0U
CwIM9AuSDNQMUhy0HTlCdC5CHDQvEjwEL8Jr5B/ye8Qf8oukH/KbpB/ym6Qf8pukH/KbpB/yn/Kx
wJ/zUWCP9BFwb/Sh0G/1AjB/1RKwn0UDMM7E08EOVJSBXgR1Qa20VgHtZEayHTQ3Qj0EN9J
M5DhiXMRI0lykSVJchEnCbGRaMmxUWrJsNFtCbBR8nv0bNJ7xH3ye5R+4ptkf6K7NG/yyxRv8ts
Eb/LrBG/y6wRv8usEb/LrBG/y7/MBkl/zoSBv9lFgX/URsE/1chBPdZJwXtWS4l5FY3DN1TRBLWT0
8a0E1bIMxLZSTISW4nxUl3KcNJgCrBSYcrv0mPK71Jliy7Sp4sukqmLLhKryy2SrktEvHLbJL2S2v
TOkurEz3MKIL/zGnS/8ypUz/MqVM/zOITP8zpUz/M6VM/zP/NBYH/0APBf9OFAP/WBgC/V4dAvFg
lgPmYCKE3V8zCNRcPxDNWEsZx1VWIMJSYCa+UGkqu09yLbLOei+2ToIwtE6JMbnOKTGxTpkYr
06hMq1PqjKrT7QyqU/BM6dQ0jOIUeU0oIdzNZ9R/jadUf83nFD/N5xQ/zecUP83nFD/N5xQ/zf/OBM
G/0UNBP9TEgL/XRUB+GMZAepmHQHgZyMB1mYuBs1kOw7GYEcXv1xSlpZXCe2VmUssIVtM
K9TdTKtU3w0q1OENalTjDanU5Q3pVOdN6NTpjihVLA4n1S8OZ1UzDmbVeA5mFTwOpZU/DuVV
P87IFT/O5NU/zuTVP87k1T/O5NU/zv/PBAG/0oMA/9YEAH/YREA8mcTAORqFQDZbBwAz20rBM
drOAy/aEQWuGROH7NgWCeuXWEtqltpMqdZcDWkWHg4old/OqBXhzueV488nFeYPZpYoT2YV
6w+lle4PpRXxz6SV9w/kFftP45X+j+NV/8/jFf/P4tX/z+LV/8/i1f/P4tX/z//Pw0F/08LAv9cDQD9ZQ0A7
WoMAN9tCwDTcRcAyXMoA8FyNQq5bkEUs2pLHa1mVCaoY10tpGBIM6BebDecXXM6mlx7PZdb
gj+VW4tAk1qUQZFanUKPWqhCjVq0Q4taw0OJWthDh1rrRIZa+EOEWv9DhFr/Q4Ra/0KEWv9Ch
Fr/QoRa/0L/QgoE/1IKAf9fCgDxaAgA220FANVxCADNdRQAxHglArx3Mgm1dD4SrnBIG6htUSSia
VosnWZhM5lkaTiWYXA8k2B3P5BefkKXodEi12QRYldmkaHXaVHhV2wR4NdweIBXdNlf13oSH
5d90d9Xf9GfF3/Rnxd/0V8Xf9FfF3/RXxd/0X/RAgD/1YIAf5iBwDiagIa1nIEANB2BgDJeREAwHwjA
bh8MAexeTwQqnZGGqNyTyOdb1crmGteMpRoZTiQZmw9jGRzQYliekSGYYNHhGCMSYFglkp/Y
KFLfWCtS3tgvEt5YM9Ld2DIS3Zg9Up1YP9JdWD/SHVg/0h1YP9IdWD/SHVg/0j/RwYD/1kGAPVIA
wDbbgEA0nUDAMt6BADEfQ4AvlAgAbSALgatfjoOpntEGJ93TSGZdFQqk3BcMY5tYjiKamk9hmv
QoJmd0Z/ZX9JfGSIS3pkkk13Y55OdWOqTnNjuU9xY8xPb2PjTm9j801uZP5MbmT/S25k/0puZP9
KbmT/Sm5k/0r/SgQC/1wEAOpnAADYcQAAznkCAMd+AwC/gQsAulQeALCELAWpgjgNon9CFpt8
Sx+VeVloj3VZMlpyYDeFb2Y9gGxsQnxqc0d4aXtLdWiETnJnj1BwZ5tRbWenUWtntlFpZ8lRaGfgU
Whn8k9nZ/1OZ2f/TGdn/0xnZ/9MZ2f/TGdn/0z/TQMB/14CAN9qAADTdQAAyn0BAMOBACQ7hAg

AtIcbAK2IKgSmhzYLn4RAFJiBSR2RfVAmi3pXL4V3XTaAdGM9enFpQ3ZvcEhybnhMbmyBT2tsi1
JobJdTZmylU2Rrs1Nia8VTYWvdUmFr8FFHa/xPYWv/TmFr/05ha/9OYWv/TmFr/07/UQIB/GEAAN
xuAADQeAAAx4AAAL+FAAC3iAUAsIsYAKmMJwOizQJm4k+EpSGRxuNg04kh39VLYF8WzV7e
mE8dXdnQnB1bUhrc3VMZ3J+UGRxiFNhcZRUX3GiVF1wsFRccMNTWnDbU1pw7IFacPtQW3D/
T1tw/09bcP9PW3D/T1tw/0//VQEA8WQAANhyAADMfAAAw4QAALuJAACzjAEaq48VAKWQJQKe
kDEII448EJCLRRmJiEwig4VTK3yCWTN2gF86cH1IQWt7a0dleXJMYXh7UF13hVJbd5FUWXefVF
d3rINVdsBTVHbYUIR27VFUdfpQVHX/T1R1/09Udf9PVHX/T1R1/0//WAAA4mgAANR2AADlgAAA
v4gAALaNAACukAAAppMRAKCVlgGalC8Gk5M5DoyQQheFjkogfotRKHiJVzBxhl03a4RiPmWCa
URggG9KW394Tlh+glBVfo9SU36dUIF+rFFQfr5QT37VT059609OfPIPTnv/Tk57/05Oe/9OTnv/Tk5
7/07/XQAA3m0AAM97AADEhQAAuo0AALKSAACplAAAOZcNAJuaHgCVmiwEjpg3DleWQBSAIE
gdeZJPJXOQVS1sjls0ZoxgO2CKZkFbiG1GVod1SIOGgE1QhoxOToabTk2Gqk1MhrxMSobTS0qF
6ktJhPhLSYP/S0mD/0tJg/9LSYP/S0mD/0v2YQAA2HIAAMqAAAC/igAAtPEAAK6XAACjmgAAm50
HAJWfGgCPoCgDiZ8zCYKdPRF7m0UZdJIMIw2XUiInllgwYZReNluTZDxWkWtBUpB0RU+QfkdM
kltlSpCZSEmQqUdlkLtGR5DSRUaO6UVFjfdGRlz/R0SL/0dEi/9HRIv/R0SL/0fkZwAA0nkAAMSGA
AC6kAAAAspcAAKmcAACeoAAAIKMBAl2IFAClpiQBggYvB3ulOQ50o0EWbqJIHWegTyRhn1UrW5
1bMVacYjZSm2k6TppyPkuafEBImolAR5qYQEaaqD9Emrk+Q5rRPUOZ6T1BI/c/QJb/QECV/0BAIf
9AQJX/QECV/0DdbwAAy4AAAL+MAAC2lQAARZ0AAKOiAACZpgAAi6kAAISrDQCArR4Aeq0qBH
StNAptrD0RZ6pEGGGpSx9bqFEIVqdYKlGnXy9NpmYySaVvNUelejdFpYc3Q6aWN0KmpjZBprc1
QKXONEck5zM+o/U2PKH/Nzyh/zg8of84PKH/ODyh/zjUeAAAxlcAALmTAACwnAAAp6MAAJ2pA
ACSrAAAhq8AAHqzBQB1tRcAcbyIAmu2LwdltTgNX7Q/E1m0RhIUs00eULJTI0yyWydlSWlqRbFs
LEKxdy1BsYQuP7KTLT6yoy09srUsPLLLKzyw5So6r/QrOK3/LTit/y44rf8uOK3/Ljit/y7LgQAAvZAA
ALSbAACrowAAoaoAAJavAACKsgAAfrYAAHK6AABpvg0AZcAdAGHBKANcwTEIV8A5DVLAQBJ
NwEcXSb9OG0a/Vh5Cv14gQL9olj6/cyM8v4AjO8CPlzrAoCI5wLEhOMDIIDe/4iA2vflhNLv+lzS7/y
M0u/8jNLv/lzS7/yPCjAAAt5gAAK6iAAClqgAAmbAAAI21AACBuQAAdb0AAGnCAABdyAUAV80R
AFXOHgFRzygETc8xCEnPOAtFz0APQs9HEj/PTxQ8z1gWOs9iGDjQbRg30HsZNtGKGDXRnBg0
0a0XMtHDFjHR3hUxzu4WL8z6GC7M+XguzPsYLSz7GC7M+xi6lgAAsKEAAKeeqAACcsQAAkLYA
AIO7AAB2vwAAasQAAF7KAABU0AMAS9glAEjdEwBF3h8BQt8oAz/gLwY84DclOuA/CjfhRww14
VANM+JaDjLiZQ8w43IPL+SBDy7kkg4t5KMOLOW2DSvkzQ0p4+QMKOD0DSfg9g4n4PYOJ+D2
Difg9g6zoAAAqqoAAJ+yAACTuAAAhbwAAHjBAABrxgAAXssAAFTRAABK2AAAQ+MFAEHsEgA
+7RsBOu4iAzfuKQQ07y8GMe81By7wPAgs8EQJKvFNCijyVwsm8mMLJPNxCyP0gAoi9JIKIfSkC
SD1tQkf9MglHvXgCB715Age9eQIHvXkCB715AisqQAAorIAAJW4AAClvgAAecMAAGvIAABezQA
AU9MAAEaAABA3wAAO/EDADj5DQA1+hUBMfsbAi78IQMr/CYEKP0BSX9MAYi/jcGIP4+Bx3/R
wcb/1EIGf9eCBb/bAcV/3wHFP+OBxT/ngYT/60GEv+9BRL/wQUS/8EFEv/BBRL/wQWksQAAI7kA
Alq+AAB7xAAAAbMkAAF7PAABS1QAAR9sAAD3hAAA15gAAMfoAAC7/BgAq/wwBJ/8TAST/GAli/
x0DHv8gAxv/JQQZ/yoEFv8wBRP/NwUQ/z8FDf9KBQr/VgUH/2UFBf90BQT/hAQE/5IEA/+gBAP/o
wQD/6MEA/+jBAP/owSauQAAjL8AAH3FAABuygAAX9AAAFHXAABG3gAAO+MAADLoAAAr8QA
AJ/8AACT/AAAh/wMBHv8JARv/DQEY/xICf8WAhL/GQMP/x0DC/8iAwj/JwQF/y4EAf82BAD/QQ
QA/00EAP9aBAD/aAMA/3UDAP+CAwD/hAMA/4QDAP+EAWD/hAP/BywK/w0pCv8PKAv/DSol/w
0yDP8OPQz/DUoN/w1YDf8NZQz/DXEM/w58C/8Qhgr/EY4K/xGVcf8SnAn/eqIJ/xOoCf8TrQn/FL
QJ/xS7Cf4VxAj8Fc8l+hbdCPcW6gj0F/QH8hf8B/AX/wjvGP8K7hj/Cu4Y/wwuGP8L7hj/C+4Y/ww/Cio
K/xEnCv8UJgr/EigL/xQwDP8VOw3/FEgN/xRVDf8TYg3/E24N/xR5DP8Vgww/FosL/xeTC/8Xmgv+
GKAK/RimCvwZqwr7GbIK+hm5Cvkawgn4G80J9RvbCflb6AnvHPMI7Bz8Cesd/wrqHf8M6R3/Dec

e/w3nHv8N5x7/Dece/w3/DycK/xYkCv8Zlwr/GiYK/x0uC/8dOQ3/HEUO/xtRDv8aXg7/GmsO/xp2Df
4bgAz9HIgM+x2QDPodlwz5Hp0M9x6jDPYfqQv1H68L9CC3C/MgvwvylcoK7yHZCuwi5grplvIK5iP
7C+Qj/w3jl/8O4CT/D94l/xDeJf8Q3iX/EN4l/xD/FCUK/xshCf8eHwn/liQK/yYsC/8mNgz/JEEO/yNND
/8iWg/9IWYP+iFyDvgifA72l4UO9CONDvMklA7xJZoN8CWgDe8lpw3uJq0N7Ca0DOsnvQzqKMg
M6CjWC+Qp5QvhKfEM3in7D9sp/xDYK/8S1Sv/EtQq/xPUKv8T1Cr/E9Qq/xP/GilK/yAdCf8jHAj/Kyl
J/y8qCv8wMwz/Lj0O/ytJD/oqVRD2KWIQ8iltEPApdxDtKoAQ7CuJEOorkA/pLJcP5yydD+YtpA/ILqo
O4y6yDulvuW7gL8YN3jDUDdsx5A3WMPAQ0zD7Es8x/xTNmf8VyzH/Fsox/xfKMf8XyjH/F8ox/xf/H
h4J/yUZCP8rGAf/NB8H/zgmCP85Lwr+NzkN9zRED/lyUBHtMVwR6TFoEuYxchLkMnwS4jKEEuA
zjBHeNJMR3TSaEds0oRHaNagR2DWvEdY1uBHUNsMR0jbQEs834RLLOO4Uxzj6F8Q3/xjCN/8a
wDf/G783/xu/N/8bvzf/G783/xv/lxsl/yoVB/80Fgb/PBWG/0AjBv9BKwj2QDQL7z0/Duk6SxDkOVcS4
DhjE9w5bRTZOXcV1jmAFdQ5hxbSOo4W0TqVFs86nBbNO6MWyzuqF8o8shfIPL0Xxj3KF8M92x
e/PuoZvD73G7k9/x23Pf8etT3/H7U9/x+1Pf8ftT3/H7U9/x//Jxch/zARBv88FAT/RBkE/0kgBPhKJwX
vSC8l50U5C+BCRQ/aQVES1EBdFtE/ZxjNP3EZyz95Gsg/gRvHQIkxbUCQHMBNlXzBQZ4cwEGI
HL5CrRy8QrcdukLEHbhD1B21Q+YeskPzIK9D/iKtQv8jrEL/JKtC/ySrQv8kq0L/JKtC/yT/LBMG/zYN
BP9DEgP/SxYC/lAbAvFRlgPnUCkE3k0zB9ZLPw3QSUwTykdXGMZGYRvCRWsewEVzH71FeyC
8RYMhukWKlhbGkSG3RpkItUahlrNHqSKxR7lJr0e+l6lIziOrSOEkqEjwJaVI/CejSP8ooUn/KKBI/yig
SP8ooEj/KKBI/yj/MRAF/z0LA/9JDwH/URMB91YWAOpXHADgVilB1IYuBc5UOwzHUkcTwU9SGb
xNXB65TGUhtktul7RLdiWySn0msEqFJq5LjCesS5Qnq0ucKKILpCinTK4opUy5KaNMMyCmhTdspn
k3sK5tN+SyaTf8smEz/LZhM/y2YTP8tmEz/LZhM/y3/NQwE/0lJAv9PDAH/Vg4A8VoQAONbEwDY
XBsAzl4qBMZcNwu/WkMSuVdOGBRUWB+wUmEjrVFpJqtQcSmoT3gqpk9/K6VPhyyjT48toVCXL
Z9QoC2dUKoum1G1LplRwy+XUdYvIVDpMJJQ9zGRUP8xkFD/MY9Q/zGPUP8xj1D/MY9Q/zH/O
AkD/0clAf9TCQD2WQkA5F0IANpeCQDRYhcAyGUnAr9kNA5YUARsl5KGa1bVB+pWF0lplZlKa
JVbCugVHMunVR7L5tUgjCaVloxmFSTMpZUnDOUVKYzklOxM5BTvzSOU9E0jFPmNYpT9TWIU
/81iFP/NYdT/zWHU/81h1P/NYdT/zX/OwYD/0sGAP5WBgDIXQlA2WIEANJlBgDLA BMawmsjAbpq
MQezZzwPrWRHF6dhUB+jXlklNlxhKptbaC2YWW8wllh2MpNXfjSRV4Y1j1eONo5WmDeMVql4iVa
tOldWuzmFVs05g1bjOYJW8zmAVv45gFb/OH9W/zh/Vv84f1b/OH9W/zj/PgMC/08EAPRZAQDcY
QAA02cCAMxqBADFbBAAvXAhAbVwLgaubToNqGpEFqJnTR6dZFYkmGFeKpVfZS+RXWwyj1x
yNYxbejeKWol5h1qKOoVZIDuDWZ88gVmqPX9ZuD19Wco9e1nfPXpZ8T15Wf08eFn/PHhZ/zt4W
f87eFn/O3hZ/zv/QgEB/1ICA0dcAADYZQAazmwBAMdvAwDAcQwAuHQeALF0LAWqcjcMpHBC
FJ5sSxyYaVMjk2ZaKo9jYS+LYWgziF9vN4VedjqCXX48gF2HPn5ckT97XJtAeVynQHdctUF1XMZ
Bc1zcQXJc70BxXPs/cVz/PnFc/z5xXP8+cVz/PnFc/z7/RQAB/1QAAN9fAADTaQAAYnAAAMNzAQ
C8dQkAtHgcAK15KgOndzUKoHU/EppxSBuUblAij2tYKYpoXi+GZmUzgmNsOH5iczt7YXo+eGCD
QHZgjUJ0YJhDcmCIRG9fskRtX8NEa1/ZRGtf7UNqX/pCal//QWpf/0BqX/9Aal//QGpf/0D/SQAA+Fc
AANxjAADQbQAAX3QAAL93AAC4eQYAsHwZAKp9JwKjfdMJnHk9EZZ2RhmQc04hinBVKlvtXC
6AamlzfGhoOHhmbzx0ZXdAcWSAQ29kikVsY5VGamOiRmhjr0ZmY8BGZGPWRmRj60Vky/IEZG
P/QmRj/0Jky/9CZGP/QmRj/0L/TAAA71oAANlnAADMcQAaw3cAALt7AAC0fAMArH8WAKaBJQK
fgDEHmX47D5J7RBeMeEwfhnVTJoByWS17b18zdm1mOHJrbD1uanRBa2l8RGhohkZlaJJlY2ifS
GForUhfZ71IXmfTR11n6UZdZ/hFXWf/RF1n/0NdZ/9DXWf/Q11n/0P/TwAA5F0AANRrAADJdAAA
wHsAALh/AACwgAAAqIMTAKKFlwGchC8GIYI5DY+AQhWlfUodgnpRJXx3Vyx2dV0ycXNjOGxxaj
1ob3FBZG55RWFtg0debY9JXG2cSVttqklZbLtlV2zQSFds6EdXbPdGV2v/RVdr/0RXa/9EV2v/RFd
r/0T/UgAA4GEAANFvAADFeAAAvH8AALSDAACrhAAApIcQAJ6JIACYiS0FkYc3C4uFQBOEgkgb
foBPI3h9VSpye1swbHlhNmd3ZzxidW5AXnR2RftzgEdYc4xIVnOZSVVzqEhTcrhUnLOR1Fy5kZR

cfVFUXH/RVFw/0RRcP9EUXD/RFFw/0T7VgAA3GYAAM1zAADBfQAAuYMAALCHAACniQAAAn4
sLAJmNHQCTjSoDjYw1CYaKPhGAiEYZeYZMIHOEUydtglkuZ4BfNGJ+ZTldfGw+WXt0QIV6fUVT
eolGUXqXR096pkZOerZFTXnMRUx55ERLePRES3f/Q0t3/0NLd/9DS3f/Q0t3/0PtWgAA1msAAMh
4AAC+gQAAtYgAAKyMAACijgAAMZAHJSSGQCokycCiJlyB4GROw57j0MWdI1KHW6LUCRoiV
YrYoddMV2GYzZYhGk7VINxP1CCe0FOgodDTIKVQ0uCpEJJgrVBSILKQEEB40BHgPNARn/+QE
Z+/0BGfv9ARn7/QEZ+/0DjYAAA0XEAAMR9AAC6hgAAsY0AAKiRAACdkwAAk5UCAI2XFACImC
MBgpgvBXyXOAx1IkETb5RIGmmSTiBjkVQnXY9aLFiOYTFUjWc2UIxvOUyLeTxKi4U9SluTPUeL
ozxFi7M7RlvJOKSK4jIcIfM7QYf9PEGH/zxBh/88QYf/PEGH/zdzZwAAy3cAAL+DAAC1jAAArZMA
AKSXAACYmQAAjJsAAIWdDgCANx4Ae58qA3WfNAlvnT0PaZxEFmObSxxdmVEiWJhYJ1SXXix
PlmYwTJZuM0iVeDVGIYQ2RZWSNkOVojVCibI0QZXHM0CU4TI/k/lzPpH9NT2Q/zY9kP82PZD/Nj
2Q/zbVbwAAxX4AALqJAACxkgAAqJkAAJ6dAACTnwAAhalAAHykBwB3phgAc6clAW2mMAZopjg
LYqVAEVyKrdXo00cU6JUIU6hWyVKoWloR6BrK0SgdSxColEtQaCQLUCgoCw/oLArPaDFKj2g
3yo8nvErOpz8LDmb/y05m/8tOZv/LTmb/y3NdWAAv4YAALWQAACsmQAAo58AAJikAACNpgAAg
KkAAHOsAABsrAAaa8fAGSvKgNfrzMHWq47DFWuQhFRrUkWTkXQGkmsVx1FrF8gQqxolkCrc
iQ+rH4kPayNJDysnSM7rK4iOqzCITmr3SA4qe8hNqj7IzWn/yQ1p/8kNaf/JDWn/yTFgQAAuY4AAL
CYAACnoAAAnaYAAJKqAACGrQAAea8AAGyyAABhtgYAXLgWAFq5lwFWuSwEUbk1B025PAAtJ
uEMPRrhKEkK4UhVAuFoXPbhjGTu4bho5uHsaOLmJGje5mho2uasZNbm/GDS42Rcztu0XMbX5
GTC0/xowtP8aMLT/GjC0/xq9iwaAs5YAAKqgAAChpwAAlawAAImwAAB8swAAcLYAAGO5AABY
vgEAT8MJAEzFFwBKxiMBR8YsA0PGNAZAxzslPcdDCzvHSw05x1QPNsddEDXHaBEzyHURMsi
EETHJIBAwyaYPLsm6Dy3l0w4txuoOK8T3DyrD/RAqw/0QKsP9ECrD/RC2lQAArZ8AAKSoAACZr
gAAjLIAAH+2AABYuQAAZbwAAFnBAABPxxgAARswEAD3SCQA71BQAODufADfWKAe11zADM9
g4BDHYQQUv2UoHLtlUByzaXwgr2mwIKtt7CCncjAco3J4HJ9yxBiXcyAYj2ulGIltjwByHW9wch1vc
Hldb3ByHW9wevngAap6gAAJyvAACPtAAAgrgAAHS7AABmvwAAWsQAAE/JAABFzgAAPdQAA
DXaBAAx5AwAL+UXAC3mIAAr5ycBKeguAifoNQMM6T4DJOphBCLqUQQg610FH+xrBR7sewQd
7Y0EG+2gBBruswMZ7cgDF+3eAxs6wIW7OsCFuzrAhbs6wKppwAAAnq8AAJK1AACeugAAdr4A
AGfCAABaxwAATswAAETRAAA71gAAM9wAACvhAAAp8gcAJ/QRACX1GAAj9R4BIPYkAR73KQ
lc9zACGfg4Ahf5QQMV+UsDEvpYAxD7ZwMO+3gDDfyKAgz8nAIL/a0CCvy9AQn8ywEJ/MsBCfzL
AQn8ywGhrwAAILYAAle7AAB4vwAAacQAAFrJAABOzwAAQ9QAADnaAAAw3wAAKOMAAACPtA
AAg/QAAHv8IABz/DgAa/xQAF/8ZART/HgES/yMBDv8pAQv/MAII/zkCBf9EAgl/UQIA/2ACAP9xA
QD/ggEA/5MBAP+hAQD/qgEA/6oBAP+qAQD/qgGXtgAAibwAAHrBAABRxxgAAW8wAAE3SAABC
2AAAN94AAC7iAAAm5gAAHuoAABr3AAAX/wAAf8AABP/AwAR/wkADf8NAAr/EQAH/xYBBP8b
AQD/IQEA/ygBAP8wAQD/OwEA/0gBAP9XAQD/ZgEA/3UBAP+DAQD/jAAA/4wAAP+MAAD/jAD/
ASkJ/wUmCf8FJgn/ASgJ/wEwCf8CPAn/AkkJ/wJWCP8CYwj/Am8H/wN5Bv8EgwX/BYsF/waSBP8
GmQT/B54E/wekA/8HqgP+B68D/Qi2A/wlvgl7CcgC+QnVAvYK4wLzCu8C7wv4Au4L/wLsDP8D6
w3/BOoN/wTqDf8E6g3/BOoN/wT/BSYJ/wokCP8KIwj/ByUI/wouCf8KOQn/CUYJ/wITCf8JYAj/CGw
H/wp2B/8LgAb/DlgF/wyQBf0Nlgt8DZwE+w2iBPoOpwP5Dq0D+A60A/cPvAP2EMYD9BDSA/AR4
QPTEe0C6hL4A+gT/wTnE/8F5RT/BulW/wbiFv8G4hb/BulW/wb/CiQJ/w8hCP8QIAj/ESQI/xMsCf8T
Ngn/EkMJ/xFQCf8RXAn/EWgl/hFzB/wSfAf6E4UG+BSMBvcUkwX2FZkF9BWFbPMVpQTyFqsE8
RaxBPAXuQTuF8ME7RjQBOoY3wPmGewD4xr3BOEA/wbeG/8H2x3/CNkd/wjZHf8l2R3/CNkd/wj/
ECEI/xUeCP8WHAf/GyIH/x0qCP8dMwn/Gz8K/xpLCv4ZWAR6GWQJ9xlvCPQaeQjyG4EH8RuJB+
8ckAbuHJYG7B2cBusdogXqHqgF6R6vBecftwXmH8EF5CDOBOEg3gTdlesF2iL3B9Yj/wjSJP8K0
ST/C88k/wvPJP8LzyT/C88k/wv/FR4l/xoZB/8cGAb/JB8G/ycnB/8nMAj/JTsJ+yNHCvYiUwrylI8K7iJ

qCewidAnpl30l5ySFCOYkjAfkJZMH4yWZB+EmnwfgJqYG3yetBt0ntQbbKL8G2SjMBtYp3AbSKus
Izir2Csor/wzIK/8Nxiv/DsUr/w/FK/8PxSv/D8Ur/w//GhoH/x8UBv8lFQX/LRwF/zAjBv8xLaf6LzYl8yxB
Cu0rTgroKloK5SplCulrbwrfLHkJ3SyBCdstiAnZLY8J2C2WCdYunAnULqMK0i+pCtEvsQrPMLsKz
TDHCssx1grGMucMwjL0Dr8y/hC9Mv8RuzL/Eroy/xO6Mv8TujL/E7oy/xP/HxYG/yQQBf8vEwT/Nhk
D/zkfBPs5JwXyODEG6jU8COQzSAreMIQL2jNgC9YzagzTM3MN0TR7Dc80gw7NNYoOyzWQDs
o2lw7lNp4OxjelDsQ3rA/DOLUPwDjBD7440A+7OelQuDnwErU5/BSyOf8VsTj/FrA4/xewOP8XsDj/
F7A4/xf/lxIF/ysLBP83EAL/PhUC/0EbAvNBIGLpPyoE4Tw1Bto7QgjTOk4MzzpZD8s6ZBDIO20RxTt
1EsM7fRLBPIQTvzyLE748kRO8PZgTuz2gFLk9qBS3PrAUtT68FLM+yhWxP9wVrj/sF6s/+RmpPv
8apz//G6U//xulP/8bpT//G6U//xv/KA0E/zlJAv8+DQH/RBEA+EcVAOtHHAHhRSMB2EUvBNBEPAn
JQ0gOxEJUEcBCXhS9QWcVukFvF7hBdx3QX4YtUKFGLNCjBmyQpMZsEKbGa9DoxmtQ6waq
0O3GqlExRqnRNYbpETpHKFE9h6eRP8fnUT/H5xE/x+cRP8fnET/H5xE/x//LAKD/zkHaf9ECQD/S
QsA8UwOAONLEgDYSxwAz04qA8dNNwjAS0MOu0pOE7dIWBA0R2lZsUdqG69HcRytR3kdq0eA
HalHhx6oR44epkeWH6Vlnx+jSKgfoUiylJ9JwCCdSdEgmknkIZhJ8yKWSP8jIej/JJNi/ySTSP8kk0j/
JJNi/yT/MAUD/z4FAf9lBgDwTQUA3U8FANIPCADQUxcAx1YmAr9VMwe5Uz8Os1FKE69PVBirTI
0bqE1IHqZMbCCkTHQhokx7lqBMgiOeTlkjnUyRJJtMmiSZTKQII0yuJZVMuyWTTMwmkUzhJo9M
8SeNTP0ojEz/KltL/yiLS/8oi0v/KltL/yj/NAEC/0MCAPdMAQDfUQAA1IYDANBxBgDJWRMAwF0jAb
lcMAayWjsMrVhGE6hVUBmkVFgdoVJgIJ5RaCOBUW8lmVB2JpdQfSeVUIUoIE+NKJJPlimQUKA
qjk+rKoxPtyqKT8griE/dK4ZP7yyET/ssg0//LIJP/youCT/8rgk//K4JP/yv/OAAB/0cAAOIPAADZVwAA0
FwBAMpeAwDDXw4Au2MfALRjLQWtYTgLP15DEqJcTBidWVUdmlhdIpZWZCWUVWsnkVRyKY9
TeSuNU4Asi1OJLYpTki6lUpwuhlKnL4NStC+BUsQvf1LZMH1S7DB8Uvowe1L/L3tS/y97Uv8ve1L/
L3tS/y//PAAB/koAAOBTAADUXAAAY2IAAMRkAQC9ZAsAtmccAK9oKgSoZjUKomRAEZ1hSReY
X1ldlFzZlpBaYCaNWWcpilduLlhXdS6FVn0vg1aFMIFVjjF//ZkyfVWkM3tVsTN5VcE0d1XVNHVV6
jR0VfgzdFX/MnNV/zFzVf8xc1X/MXNV/zH/QAAA9U0AANxYAADQYQAAX2cAAL9pAAC5aQgAsW
wZAKttJwOkbDMlnmk9D5hnRhaTZE8cjmFWlopXfSaHXWQqhFtrLYFacjB+WxkyfFmBNHpZizV3
WZY2dVmhNnNYrjdxWL43b1jSN25Y6DdtWPc2bVj/NWxY/zRsWP80bFj/NGxY/zT/QwAA6k8AANI
cAADMZQAAw2sAALtAAC0bQQAARXAXAKdxJQKgcDEHm47DpRrRBWPaeWbimZTIYVjWiaBY
WErfV9oLnpebzF3XXY0dVx+NnJciDhwXJM5blyfOmxcRdpqXLs6aFzPOmZb5jlmXPU4ZlZ/N2Zc/z
ZmXP82ZlZ/NmZc/zB/RgAA5FMAANvgAADJaQAawG8AALhxAACwcQEAqXQUAKN1lwGddS8G
lnM5DJBwQhOLbUoahWpRIIBoWCZ8Zl4qd2RILnRiazJxYXM2bmB7OGtghTppYJA7Z2CcPGVgq
Txix7k8YV/NPGBf5DxgX/Q6YF/+OWBf/zhgX/84YF//OGBf/zj/SAAA4FcAANFkAADGbQAavHMA
ALR1AACtdQAAPxcRAJ95lACZeS0Ek3c3C411QBKHckgYgW9PH3xtVSV3a1wqcmliL25naTNqZ
nA3Z2V4OmRkgjxiZl09YGSZPI5kpz5cZLY+WmPKPVlj4j1ZY/M7WmP9Oipi/zlaYv85WmL/OVpi/z
n6SwAA3VsAAM5oAADCCQAauXYAALF5AACpeQAaOxSNAJt9HgCVfSoDj3w1CYl6PhCDd0YXf
XVMHXdyUyNycFkpbW5fLmltZjNka203YWp1OI5pfzxbaYo+WWmWP1hppT9WaLQ+VGjHPINo4
D1TaPI8VGf9O1Rn/zpUZ/86VGf/OIRn/zrvTAA2V8AAMpsAAC/dQAAtnoAAK5+AACKfQAAnYAK
AJeBGwCRgigCi4EyB4V/Ow5+fUMVeHpKG3N4USJtdlcnaHvDLWNzYzJfcWo2W3ByOVhvfDxVb
4c+U2+UPIJvoj5Qbrl9T27FPU5u3jxObfA7Tm38O05s/zpObP86Tmz/Ok5s/zrmUwAA1GQAAMZw
AAC8eQAAs38AAKqCAACfgQAAMlQGAJKGFwCMhyUBhoYwBoCFOQx6g0ESdlFIGW5/Th9ofV
UIY3tbK156YTBaeGg0VndwN1N2eTpQdoQ8TnaRPE12oDxLdbA7SnXDOKl13DpJdO85SHP7OU
hy/zllcv85SHL/OUhy/zniWQAA0GkAAMJ1AAC4fgAAr4QAAKaHAACbhwAAkokCAIyLEwCHjCEA
gYwtBHULNngp1iT8Qb4dGFmmGTBxkhFliXoNYJ1qBXyxVgGYwUX9tNE5+dzdLfol4SX6POEh+nj
hHfq43Rn3BNkV92jVEfO42Q3v6NkN6/zZDef82Q3n/NkN5/zbcXwAAym8AAL56AAC0gwAArIkAA

KKLAACWjAAAJl4AAIWQDgCAkh0Ae5lpAnWRMwdvkDwNaY9DE2SNSHlfjFAeWotWI1WKXChRi
WMsTYhrL0qHdTFHh4AzRYeOM0SHnTJDh60xQofAMEGG2TBAhe0wP4P6MT6C/zl+gv8yPoL/
Mj6C/zLVZgAAxXUAALqAAACwiQAAqI8AAJ2QAACRkgAAhZQAAH2WCAB4mBgAdJkIAW6YLw
VplzgKY5ZAD16VRxVZIE0aVZNTIHCSWiJMkmEmSZFpKUaRcytEkH4sQpGMLEGRmytAkawqP
pC/KT2Q2Ck9ju0pO435KzqM/yw6jP8sOoz/LDqM/yzObgAAv3wAALWHAACsjwAAo5QAAJiWAA
CMmAAAFpsAAHWdAQBunxIAa6AfAGagKwJhoDQHxJ88C1ieQxBTnkkUT51QGEucVxxInF4fRjt
nIUkbcSNAm3wkPpuKJD2bmiM8m6oiO5u9ITqb1SE5mewhOJf4IzaW/yQ2lv8kNpb/JDaW/yTHdw
AAuoQAALCOAAColgAAnpoAAJKcAACGngAAeqEAAGulAABkpgkAYKgYAF2pJAFZqS4DVKg2
B1CoPgtMp0UPSadMEkWNuXVCplsXP6ZjGT2mbRs7pnkbOqaHGzmnls4p6caN6a6GTWm0hg1
peoYM6P3GjKh/xsyof8bMqH/GzKh/xu/gAAAtlwAAKyVAACjnQAAMKEAAIyjAAB/pgAAc6kAAGas
AABZrwAAU7AOAFGxHABOsicBS7lvAOiyNwZEsj8JQbJGCz6yTQ48sIYQOrJfETiyaRI2snUTNbK
DEjSykxlysQQRMBk2EDCyZhAvsecPLq/1ESyt/xlsrf8SLK3/Eiyt/xK4igAAr5UAAKedAACdpAAAkqg
AAIWraAB4rQAAa7AAAF6yAABStgAAR7oDAEO8EQBBvRwAP74mAD2+Lgl7vjYEOL8+BTa/Rg
c0v08IMr9YCTDAYwovwG8KLcB9CizBjQorwZ8JKsGxCCjAyAgnv+MHJr3zCCW7/golu/4KJbv+Ci
W7/gqykwAAqp0AAKGIAACVqgAAia4AAHuxAABtswAAYLYAAFW6AABKvQAAQMIAADfHBAAY
ygwAMMwYAC/NIQAtzSoALM4yASrPOglpz0QCJ9BOAybQWQMl0WUDI9F0AyLSHAMh0pcDINK
qAh7SwAlc0dsCHM/tAhvN+AMbzfgDG834AxvN+AOsnQAApKYAAJisAACMsAAAFrQAAHC3AABi
ugAAVr0AAErBAABAxgAAOMoAAC/PAAAO1AQAlDKIAB7cEQAE3hoAHd4jAB3fKwAc4DQAG+E+
ABriSgAZ41YAF+NkARbkdQAV5YcAE+WaABLIrgAQ5cUADeTdAAvh8QAL4fEAC+HxAAvh8QC
mpgAAm60AAI+yAACBtgAAcrKAAGO9AABWwQAASsUAAEDKAAA3zgAALtMAACbYAAAF3AAA
GeABABjsCQAW7RIAF4ZABPvHwAS8CcAEPAAvAA7xOAAL8kQACfNRAAf0YAAG9HEABPWF
AAP1mAAB9KsAAPO9AADy0gAA8tMAAPLTAADy0wCerQAAkbMAAIO4AAB0vAAAZb8AAFFEA
ABKyQAAP84AADXTAAAs2AAAjNwAAB3gAAAW4wAAEuwAABD5AQAO+wgADPwNAAAn9EwA
H/hoABP4gAAL/KAAA/zEAAP88AAD/SgAA/1oAAP9sAAD/fwAA/5EAAP+gAAD+rwAA/rAAP6wA
AD+sACUtAAAhRkAAHe9AABnwQAAWMcAAErMAAA+0gAAM9gAACrdAAAh4QAAGuQAABPnA
AAM6gAACfYAAAF/AAAF/wAAA/8CAAD/BwAA/wwAAP8SAAD/GQAA/yEAAP8qAAD/NAAA/0IAA
P9SAAD/YwAA/3QAAP+CAAD/kQAA/5EAAP+RAAD/kQD/ACYH/wAjB/8Alwf/ACUG/wAuBv8AOg
b/AEcF/wBUBf8AYQT/AGwD/wB3Av8AgAL/AlGc/wCPAf8AlQH/AJoB/QCgAfWApQD6AKsA+QCx
APcAuAD1AMEA8wDMAPEA2wDvAOgA7ADzAOoA/QDpAP8A6AD/AOYC/wHkA/8B5AP/AeQD/
wH/ACMH/wlgB/8AHwb/ACMG/wAsBv8ANwb/AEQF/wBRBf8AXQT/AGkD/wB0A/8AfQL+AIUC/A
GMAfsBkgH5AZgB+AGdAfcBowH2AqkA9QKvAPMCtgDxAr8A7wLKA00D2QDqA+cA5wTzAOQG
/QDjCP8B4Ar/Ad0M/wLdDP8C3Qz/At0M/wL/BCAH/wcdBv8GHAb/ByEG/wkqBv8JNAb/CEAG/wd
NBf8GWgX+BmUE+wdwA/kleQP3CIEC9QmJAvQJwHyCpUB8QqbAfAKoAHvCqYB7QqsAOwLt
ADrC7wA6QvIAOcM1gDiDOYA3w7yAdwQ/QHZE/8C1RT/A9QU/wPUFP8D1BT/A9QU/wP/Ch0H/
w0aBv8MGAX/Ex8F/xQnBf8TMQb/EjwG/xFJBfsQVQX3EGEE8xBsBPERdQPvEn4C7RKFAusTjA
LqE5IB6BOYAecUngHmFKQB5BSqAeMVsqHhFboB3xXGAd0W1QDZF+UB1RnyAtEb/APNHP8
EzBz/Bcoc/wXKHP8Fyhz/Bcoc/wX/EBog/xMVBf8WFQT/HRwE/x8kBP8eLQX+HDGf9xtEBfIaUAX
uGVwE6hpnBOcacQPIG3kD4xuBAuEciAlfHI4C3h2VAtwdmwLbHaEB2R6oAdgerwHVH7gC0x/D
AtEg0gLOlelCySPwBMUK/AXDJP8HwST/B78k/wi/JP8IvyT/CL8k/wj/FRUF/xgOBP8gEgP/JhgD/yg
gA/4oKQT1JjME7iQ+BegjSwXjllcF4CNiBNwkbATaJHUD2CR9A9UIhATTJooE0iaQBNAAnlwTOJ50
EzSikBMsoqwTJKbMExyq+BcUqzAXCK90Fvzsb7ss+Qm4LP8Ktiz/C7Us/wu1LP8MtSz/DLUs/wz/
GhAF/x4JA/8pDgL/LhQC/zAbAvYwlwLsLi0D5Cw4BN4qRATZK1EF1CxcBtAsZgbOLW8Hyy53B8k

ufgfHL4UHxi+LCMQwkQjCMZglwTGfCL8ypgi9Mq8luzK5Cbkzxgm3M9gJtDTpC7A09gyuNP8OrD
P/D6sz/w+qM/8PqjP/D6oz/w//HgsE/yYIAv8xDAH/NhAA+jgVAO03HADjNCUB2jMxAtM0PgXNNEs
HyTVWCcU1YArCNmkLwDZxC743eAu8N38MujeGDLk4jAy3OJMMtjiaDbQ5og2yOaoNsTm0Da8
6wQ6tOtI0qjrlD6c68xGkOv4Sojv/E6E6/xOgOv8ToDr/E6A6/xP/lwcD/y4FAf84CAD/PAoA8j0NAOQ
7EwDaOh0A0D0rAsk9OAXDPUUJvj1QC7s9Wg24PWMPtT1rD7M9cxCyPXkQsD2AEa4+hxGtPo
4Rqz6VEao+nRKoP6YSpj+wEqU/vBOjQMwToEDgFJ1A8BWAQPwWmT//F5g//xeXP/8YlZ//GJc//xj
/JwEC/zQCAP89AwDtQAMA3kEEANk/CQDQqHcAx0YmAcBGMwW6RT8KtURLDbJDVRCuQ14
SrENmE6pCbRSoQ3QVpkN7FaVDghajQ4kWokORFqBEmReeRKIXnESsF5pFuBiYRMcYIkTcGZ
RE7RqRRPobkEP/G49D/xyPQ/8cj0P/HI9D/xz/LAAB/zkAAPFBAADeRgAA1UkCANBJBQDIShMA
wE4iAbIOLwWzTTsJrktGDqpKUBKmSVkUpEhhFqFlaBifSG8ZnUh2GptlfRqaSIQbmEiMG5dlIByV
SJ4ck0ioHZFltB2PSMMejUjXHotl6h+JSPgfh0f/H4ZH/x+GR/8fhkf/H4ZH/x//MQAA/z4AAONFAAD
YTQAAzIEAAMhRAwDBUQ4AuVUfALJVLASsVDcJp1JCDqNQSxKfT1QWnE5cGZINZBuXTGscI
UxxHpNLeB6RS4AfkeUlll5LkCGMS5ohikullohLsCKGS78ihEvTI4JL6COAS/Yjf0v/l35L/yN+S/8jfkv
/l35L/yP/NgAA9UEAAN5LAADSUwAAyVcAAMJYAAC7VwoAtFobAK1bKAOnWjQlolg+DZxWSBK
YVFAXIVNYGpJRYB2PUGYfjU9tIYtPdCKJT3sjh06DJIVOjCWDTPYlgU+hJn9PrSZ9Trwne07PJ3l
O5Sd4TvUnd07/J3ZO/yZ2Tv8mdk7/JnZO/yb/OQAA6EQAANIQAADNWAAAxFOAALxeAAC2XQc
Ar18YAKhhJgKiYDEGnF48DjdbRRKTWU0Xj1dVG4tWXB6IVGMhhlNpl4NScCWBUnghm1KAKH
1SiSl7UpMpeVKeKndSqyp1Ubkr1HMK3FR4itwUfMqb1H+Km9R/yIvUf8pb1H/KW9R/yn/PAAA5E
kAANVVAADJXQAAwGIAALhJAACxYQMAqmQVAKRllwGeZS8FmGM5C5NgQhGOXkoWiVxSG4
ZaWR+CWGAif1dmJXxWbSd6VXQpeFV8K3VVhSxzVZAteVWbLW9VqC5tVbYua1TJLmlU4C5pV
PEtaFT9LGhU/yxoVP8raFT/K2hU/yv+PwAA4E0AANFZAADFYQAAvGYAALVnAACtZgAApmsA
KBqlAGaaiwElGg2Co5lQBCJY0gVhGBPGGoBeVh98XV0ieVtjJnZaaihZWXErcFh5LW5Ygi9sWl0wa
liZMGhYpjFmWLQxZFjGMWJY3TFiWPAwYlj8LmJY/y5iWP8tYlj/LWJY/y32QgAA3VEAAM5dAAD
CZQAAuWoAALFsAACpagAAomwOAJxuHgCWbioDkGw0ClpqPQ6FZ0UUGVNGXtjUx53YVoic
19gJnBeZylsXW4salx2L2dcdFIXloyY1yWM2FcozNfXLlzXVvEM1xb2zNbW+8xW1v7MFxb/y9cW
/8vXFv/L1xb/y/sRQAA2IUAAMphAAC/aQAAtm4AAK5wAAClbgAAnnALAJhyGwCTcigCjXEyB4dv
Ow2BbEMTe2pKGHZoUR1yZlcibmReJmpjZSpmYmwtY2F0MGFgfTJeYlgzXGCUNFpgoTRZYLA
0V1/CNFzf2TRVX+0yVI/6MVZe/zBWxv8wVI7/MFZe/zDnSQAA1VKAAMdIAAC8bQAAs3IAAKpzA
AChcgAAmnQIAJR1GQCODiUBiHUwBoJzOQt9ckERd29IF3JtTxxtbFUhaWpbJmRoYiphZ2ktXWZ
xMFplejNYZYU0VmWRNVRInzVTZa01UWS/NFBk1jRQZOwzUGP5MIBj/zFQYv8wUGL/MFBi/zDj
TgAA0l0AAMRpAAC5cQAAsXYAAKd3AACddgAAIXgFAI95FQCKeiMBhHouBH55Nwl4dz8Pc3VG
FW1zTBpoclMgZHBZJF9vXylcbWYsWGxuMFVrdzJSa4l0UGuONE9rnDRNaqs0TGq9M0tq1DNK
aesYSmj4MUto/zFLaP8wS2j/MEto/zDfUwAAzWIAAMBuAAC2dQAARnsAAKN7AACyewAAkHwBA
lp+EgCFfyAAf38rA3p+NAh0fTwNbntEE2l6ShhkeFAdX3dXlIt1XSdXdGQqU3NrLiBydDBNcn8yS3
GMMkpxmjJlcaoyR3G7MUZx0jBFcOkwRW/3MEVU/y9Fbv8vRW7/L0Vu/y/aWAAAYWcAAL1yAAC
zegAAqn8AAJ+AAACUGAAaioEAAISDDQB/hRwAeoUoAnSFMQZvgzoLaYJBEGSBSBVff04bWn
5UH1Z9WYNSfGEnTntpKkt6ci1JeX0uR3mKL0V5mC5EeaguQ3m6LUJ50SxBeOgsQHb3LEB1/y1
Adf8tQHX/LUB1/y3UXwAAxG0AALI4AACvgAAApYMAAJqEAACPhQAAhIcAAH2JCAB4ihcAc4sk
AW6LLgRpjclZlk+DV+IRRJah0sXVYZSG1GFWB9NhF8jSoNnJkeDcChEgnspQ4KIKUGClYlAgqc
oP4K4Jz6Czyc9gecmPH/2KDx+/yg7ff8oO33/KDt9/yjOZgAAv3QAALV+AACshQAAoYgAAJWJAA
CKigAAfl0AAHWPAGbvkRIaA5lFAGeSKgJikjMGXpE7ClmQQg5Uj0kTUI5PFkyOVhpJv0dRo1IIE
OMbiJBjHkJP4yGlz6MISl9jKUiPly3ITuMzSA6iuYgOlN1ITeH/il3h/8jN4f/lzeH/yPHbgAAunsAALGEA

ACojAAAm40AAJCOAACFkAAAd5MAAG2WAABmmAsAYpkZAF+aJQFbmi4DVpk3BIKZPgpOm
EUOS5hMEUeXUxREI1oXQZZiGT+WbBs9IncbO5aEGzqWkxs5lqQaOJa1GTeWzBg2leUYNJPO
GjOR/hsykf8cMpH/HDKR/xzBdgAAtYIAAKyMAACikQAAIpIAAluUAAB/lwAAc5oAAGWdAABcoAM
AV6ESAFSiHgBRoygBTqMxA0qiOQZHokAJRKJHDEGhtG4+oVYQPKFfEjghaBM4oXQUngGBF
DWhkBM0oaESM6GyEjGhyBExoOMQL57zEi6c/RMunP8ULpz/FC6c/xS6fwAAsoAAKiTAACdlwA
AkZgAAIWbAAB5ngAAbKEAAGCkAABTqAAAS6oIAEirFQBqyAARKwpAUGsMgl/rDkEPKxBBjq
sSag3rFAJNaxZCzOsZAsyrG8MMKx9DC+tjAsurZ0LLKyuCiusxAkqq98JKanxCiio+wsnqP8MJ6j/D
Ceo/wy0iQAAq5MAAKObAACXnQAAiqAAAH6jAABypgAAZaoAAFmsAABNrwAAQrIAADu0CQA4
tRUAN7YfADW3KAA0tzABMrc4AjC4QAMuuEkELLhSBCu4XQUpuWkFKLI2BSe5hgUmuZcEJLm
pBCO5vgMhuNkDILbtBB+1+QUftP0FH7T9BR+0/QWukgAAppsAAJ2iAACRpQAAhKkAAHetAAB
prgAAXLEAAFCzAABFtgAAPLoAADK9AAAqwQYAJsMQACXEGgAkxCIAI8UrACLGMwAhxjwAI
MdGAB/HUQEeyF4BHMhrARvJewAayY0AGMmhABbJtQAUYm8AE8boABPE9gETw/oBE8P6AR
PD+gGpnAAAoKMAAJWpAACIrQAAerAAAGyyAABetQAAUbgAAEa7AAA8vgAAM8IAACrGAAAj
ygAAG80DABXRCAAS0xAAEdQYABHVIQAQ1ioADtc0AA7YPwAN2EwADNIbAAvZawAJ2n0AC
NqRAAbZpQAE2boAA9jRAAHW5QAB1uwAAdbsAAHW7ACjpAAAmKoAAIuvAAB9sgAAbrUAAG
C4AABSuwAARr8AADzDAAyXwAAKcsAACHPAAAa0gAAFNYAAA3ZAAAH3QMABeAKAAThE
wAD4hsAAuMkAADkLgAA5TkAAOZHAADnVwAA52kAAOh8AADokAAA6KQAAOI1AADoxgAA6
M4AAOjOAAADozgCagwAAjrAAAIc0AABxuAAAYrsAAFO/AABGwwAAO8gAADHMAAAa0QAAH9
UAABjZAAAR3AAAC98AAAXhAAAA5QAAAO0BAADtBwAA7g4AAO4WAADvHgAA8CcAAPEzA
ADyQQAA9FEAAPVkaAD1eAAA9osAAPacAAD2qQAA9q8AAPavAAD2rwCRsQAAg7YAAHS6A
ABkvgAAVclAAEfHAAA7zQAAL9IAACbWAAAd2wAAft8AAA7iAAAI5AAAAuYAAADpAAAA8AAA
APcAAAD4AAAA+AEAAPkHAAD5DgAA+hcAAPshAAD9LAAA/jkAAP9LAAD/XQAA/3AAAP+AAA
D/jQAA/5MAAP+TAAD/kwD/ACMG/wAgBf8AIAx/ACIE/wAsBP8AOAP/AEUC/wBSAv8AXgH/AGk
B/wBzAP8AfAD9AIQA+wCLAPkAkQD3AJcA9QCcAPMAoQDyAKYA8QCsAPAAsgDvALoA7gDE
AOwA0ADrAN8A6QDtAOcA9wDmAP8A5AD/AOEA/wDfAP8A3wD/AN8A/wD/ACAG/wAdBf8AHA
X/ACAE/wAqBP8ANQP/AEID/wBPav8AWwH/AGYB/gBwAPsAeQD4AIEA9gCIAPQAjgDyAJQA8
ACZAO4AnwDtAKQA7ACqAOoAsADpALgA6ADBAOYAzQDkAN0A4gDrAOAA9gDfAP8A2gD/AN
kA/wDYAP8A2AD/ANgA/wD/ABwF/wAZBf8AGAT/AB8E/wAnA/8AMgP/AD4D/wBLAv8AVwH7AGI
B+ABsAPUAdQDzAH0A8ACEAO4AiWdsAJEA6gCWAOGAnADmAKEA5QCnAOMArQDiALUA4A
C/AN4AywDcANsA2gHqANgD9wDSB/8A0Aj/AM8J/wDOCF8Azgn/AM4J/wD/BRkF/wUVBP8FFQT
/Ch0D/wokA/8JLgP/BzoC/AZGAvCFUwHyBV4B7wVoAOwGcQDqBnkA6AaAAOYghwDIBo0A4wa
TAOEGmQDeB58A3QelANwlqwDaCLMA2Am9ANUJygDTCTsA0A3sAMsR+ADIEv8BxhP/AcUT/
wHEE/8BxBP/AcQT/wH/ChUF/woQA/8QEgP/FRkC/xYhAv8UKgLG6EjUC8xFBAu0QTQHpefKB5R
BjAOIQbQDfEXUA3RF8ANsRgwDaEYkA2BGPANYSIQDVE5sA0xOiANEUqQDPFbEAzRa7AMs
XxwDJF9gAxBroAcEb9gG+HP8CvB3/A7od/wO5Hf8DuR3/A7kd/wP/DxAE/xAJA/8ZDgL/HhUB/x8
dAfoeJQHwGy8B6Ro7AeMZRWHeGVMB2hpeANYaZwDTG3AA0Rx3Ac8dfgHNHoUBzB6LAcofk
QHJH5cBxyCeAcUhpQHDIawBwSK2Ab8jwgG9I9IBuiTKArYI8wOzJv4EsSb/BbAl/wavJf8GryX/Bq8
l/wb/FakD/xkGA8iCgH/JhAA/CYXAPAIHwDmlygA3iE0AdghQAHSI00BziRYAsolYgLIJmoCxSdy
A8MoeQPBKH8DwCmGA74qjAO9KpIDuyqZA7oroAO4K6gEtiyyBLQsvgSyLc0Ery3gBawu8AaqLv
wlqC7/CKYu/wmLV8JpS7/CaUu/wn/GAQC/yIDAf8qBgD/LAoA9CwOAOcFgDcJx8A0yotAMwsO
gLHLkcDwy9SBL8vXAW8MGUFujBsBrgxcwa3MXoGtTGABrQyhweyMo0HsTOUB68znAetM6QHr
DSuB6o0uQioNMglpjXbCKM17AqfNfklNjX/DJw1/w2cNP8NnDT/DZw0/w3/HQAB/ykAAP8wAQDt

MglA4DAEANssCQDSMBkAyTUnAMI3NQO9N0EFuTdMB7U3VgiyN18JsDhnCa44bgqtOHUKqzh
7Cqo5gguoOYgLPzmQC6U6mAukOqAMojqqDKA7tQyeO8MNNdvWDZk76Q6WO/cPITr/EJM6/xC
TOv8Rkzr/EZM6/xH/IgAB/y4AAO41AADeOQAA1TsBANA5BgDJOhQAwD4jALo/MAO0PzwGsD5
HCKw+UQqpPlomPz5hDaU+aQ6jPm8Ooj52D6A+fQ+fP4QPnT+LEJs/kxCaP5wQmECmEZZAsR
GUQL8RkkDREpBA5hONP/UUiz//Flo//xSKPv8Uij7/Flo+/xT/KAAA+zMAAOI7AADWQgAAzUQAA
MdDAgDBQg4AuUYeALJHKwKtRjcGqEVCcARFTAyhRFUOnkRdEJxDZBGaQ2sSmUNxE5dDeB
OVQ38UIEOHFJJDjxSRQ5gVj0OiFY1ErRaLRLsWiUPNF4ZD4xeEQ/MYg0P+GIJD/xiBQ/8YgUP/
GIFD/xj/LQAA7TcAANxCAADQSQAAX0wAAMBLAAC6SgoAskwaAKxOKAKnTTMGokw+Cp1LR
w2aSIAQI0IYEpRIXxSSR2YVvKEdtFo9HcxeNR3oYi0eCGIIHixmlR5QZKhkefGoRHqhqrCR7gbgEfJG
31H3xx7R/Ecekf9HHIG/xx5Rv8beUb/G3IG/xv/MQAA5T0AANDIAADLTwAAwVMAALpSAAC0UAY
ArVIXAKdUJAGhUzAFnFI6CZdQRA2TT0wRKE1UF11MWxaLTGIYiEtpGYZLbxqESnYbg0p+HIFK
hx1/SpEefUqbHntKpx95SrUfd0rGIHVK3CBzSu8gckr7H3JK/x9xSv8ecUr/HnFK/x78NQAA4UIAAN
JNAADGVQAAvVKAALVYAACvVQIAqFcTAKJZIQGcWS0EI1c3CJJVQA2OVEkRilJQFYZRVxeEU
F4agU9IHH9ObB19TnMee056IHIngyF3TY0hdU6YInNOpSjXTrlj03DI21N2SNrTe0ja036ImpN/yJ
qTf8hak3/IWpN/yHyOAAA3UcAAM5SAADCWgAAuV0AALFdAACqWgAAo1wQAJ1eHwCYXioDkl
w0B41aPgyIWEYRhFdNFYBVVBh9VFsbelNhHXhSaB91UW8hc1F3lnFRgCRvUYolbVGVJWtRoi
ZpUa8mZ1HAJmVQ1iZkUOsmZFD5JWNQ/yRjUP8jY1D/I2NQ/yPqOwAA2ksAAMtWAAC/XgAAtm
IAAK1hAACmXgAAn2ANAJliHACUYigCjmEyBolfOwuEXUMQf1tLFHtZURh3WfGbdfDeHnFWZS
BuVWwjbFR0JWpUfSZoVlcnZISTKGRUoChiVK0pYFS+KV5U0yldVOooXVT4J11T/yZdU/8IXVP/J
V1T/yXnQAAA1k8AAMdaAAC8YgAAs2YAAKpIAACiYgAAm2QKAJVmGQCQZiUBimUwBYRkOQ
p/YkEPemBIE3ZeTxdyXFUbbItcHmtaYiFoWWokZVhxJmNYeihhWlUpX1iQKl1YnSpbWKsrWli8Kl
hX0SpXV+gqV1f3KFdX/ydXV/8nV1f/J1dX/yfkRAAA0IMAAMRfAAC5ZgAAsGkAAKZpAACeZgAAI
mgHAJFpFgCLaiMBhmotBIBoNgl7Zz4OdmVGEEnFjTBdtYVMbaWBaHmVfYCYjXmclX11vJ11ceCl
bXIlrWVvYOK1dcmyxVXXksVFy6K1JbzytSW+cqUlv2KVJa/yhSWv8oUlr/KFJa/yjgSAAAz1cAAMFiA
AC3aQAARW0AAKNsAACZagAAkmwDAIxtEwCHbiAAgW4rA3xtNAd2bDwMcWpDEW1oShZoZ1
EaZGVXHMbKXiJdY2UIWmJsKFdhSpVYYArU2GMLFFhmSxQYagsTmC4LE1gzStMYOUqTF/1
KUxf/yIMXv8oTF7/KExe/yjdTQAAY1wAAL5nAAC0bgAAqnAAAJ9wAACVbgAAjXAAAI dxEACCcx0
AfXMoAndyMgZycToKbXBBd2huSBRjbU4ZX2tVHVtqWyBYaWlkVGhqJ1JncilPZ30rTWaJK0xmly
tKZqYrSWa2KkhmyypHZeQpR2T0KUdk/ihHY/8oR2P/KEdj/yjYUgAAx2EAALtrAACxcgAApnQAAJ
t0AACQcwAAiHQAAlF2DAB8eBoAeHgmAXN4LwRtdzcJaHY/DWN1RhJfc0wXW3JSG1dxWR5Tc
GAiUG9nJU1ucCdKbXopSG2HKUdtlSIFbaQpRG20KENtySdCbOlnQmvzJ0Jq/SdCaf8mQmn/JkJ
p/ybTWAAAw2YAALdwAACudwAAoncAAJd4AACMeAAAgnkAAHt7BwB2fRYAcn4iAG1+LANofjU
HY308C157QxBaekoUVnlQGFJ4VhtOd10fS3Zlkh2biRGdXglRHWEJk1kyZBdallQHWzJD91yC
M+dOEjPXLyJD1x/CQ9cP8kPXD/JD1w/yTNXgAAvmwAALR1AACpewAAAnXwAAJJ8AACHfQAAe
38AAHSBAgBugxEAa4QeAGeFKQFihTIFXoQ5CFmDQA1VgkcQUYFNFE2AVBhKf1sbR39jHUR
+bB9BfnYgQH6DIT5+kSE9fqEgPH6xHzt9xh46feAeOXvyHzh6/B84ef8gOHn/IDh5/yDHZQAUnIA
ALB7AACkfwAAmIAAAI2BAACCggAAdlUAAG2IAABmigsAYosZAF+MJABbjC4DV4w2BIOLPQIPi
kQNTIpKEEiJURNFiFgVQohgGEClaRk+h3QaPleBGzuHjxo5h58aOlewGTeHxRg2ht4XNYTxGTS
D+xozgv8aM4L/GjOC/xrBbQAAtnkAAKyCAACfhAAAK4UAAliGAAB9iAAAclwAAGSPAABekQUA
WZMTAFaTHwBTICKBUJQxA0yTOQZJk0AIRpNHC0OSTg5AkIUQPZJdEjuRZxM5kXIUOJF+FDa
RjRM1kZ0TNJGuEjORwhEykn0RMI7wEi+N+hMujP8ULoz/FC6M/xS8dQAAsYAAAKelAACaiQAA
jooAAIOMAAB3jwAAa5IAAF6WAABVmQAATpsLAEucGABJnCIAR50rAUSdMwJBnDoEP5xCBjy

cSQg6nFEKOJxZCzWcYww0nG4NMpx7DTGcig0wnJoMLpyrCy2cvwssm9oKK5nuCyqX+Qwplv8
NKZb/DSmW/w22fgAArIgAAKGNAACVjgAAiZAAAH2TAABxlgAAZZoAAAFqdAABOoQAARKMBA
D+IDgA9pRoAO6YjADqmLAA4pjMBNqY7AjSmQwMypksFMKZUBS6nXgYsp2kGK6d2BiqnhQYo
p5YGGJ6enBSanuwUkptQEI6TrBSKj9wYhov8HlaL/ByGi/wewhwAAqJEAAJuTAACPIQAAg5cAAHa
bAABrnwAAX6IAAFOIAABlqAAAPqsAADWtAgAvrw0ALa8YACywlQArSckAKrExACixOQAnsUIBJ
rFLASSyVgEjSmECIbJuAiCyfgEfs48BHbKhARyytAEZss0AGbDnARiv9QEYrv0CGK79Ahiu/QKrk
QAAopkAAJWaAACJnQAAfKEAAHCkAABkqAAAV6wAAEyuAABBSAAAN7MAAC21AAAluAEAH
7oJABY7EwAbvBwAGrwkABm9LAAyVtQAF74+ABa+SQAVv1UAFL9jABLAcgARwlQADsCXAAY/
qwAKvsEACb3aAAm87AAJu/cACbv3AAm79wCmmgAAAnKAAAI+iAACCPgAAdasAAGiuAABasAA
ATbMAAEK1AAA4uAAALrsAACa9AAAewAAAF8QAABDGBAAKyQoACMoTAAfLGwAHyyQABsw
tAAXMOAAEzUQAA81SAALOYgAAznMAAM6GAADOmGAAzq0AAMzCAADM1gAAy+YAAMvmA
ADL5gCgogAAlagAAlisAAB5rwAAa7EAAFy0AABPtwAAQroAADi9AAAuwAAAjCMAAB3HAAAWy
gAAD80AAAjPAAAC0gAAANQHAADVdWAA1hcAANgfAADaKAAA2zMAAN0/AADfTgAA4F8AAO
ByAADhHgAA4ZoAAOGrAADgugAA4MkAAODJAADgyQCXqQAAi60AAH2xAABttAAAXrcAAE+7
AABDvgAAN8IAAC3GAAAKygAAG84AABTRAAAM1AAABtYAAADaAAAA3AAAAAN4AAADgAgAA
4QkAAOMRAADIGQAA6CIAAOotAADtOgAA70oAAPBcAADxbwAA8YMAAPKVAADyogAA8q0AA
PKtAADyrQCOAwAAgLMAAHC3AABhugAAUub4AAETCAAA3xwAALMsAACLQAAAZ1QAAEdkAA
ArcAAD3gAAA0AAAADjAAAA5QAAAOcAAADpAAAA6gAAAO0BAADvCAAA8REAAPQbAAD3
JwAA+jQAAP1EAAD+VwAA/moAAP97AAD/iAAA/5MAAP+TAAD/kwD/AB8E/wAcBP8AGwP/ACA
D/wAqAv8ANgH/AEMB/wBPAP8AWwD/AGcA+wBwAPgAeQD2AIAA9QCGAPQAjADzAJEA8gCX
APEAnADwAKEA7wCmA04ArADtALMA6wC8AOoAyADnANYA5ADkAOEA8QDfAPsA3QD/ANw
A/wDcAP8A3AD/ANwA/wD/ABwE/wAZBP8AFwP/AB4C/wAoAv8AMwH/AD8B/wBMAP8AWAD5A
GMA9QBtAPIAdQDwAHwA7wCDAO4AiQDsAI4A6wCUAOoAmQDpAJ4A6ACjAOcAqgDIALAA5A
C5AOIAxADfANIA3ADhANgA7wDWAPkA1QD/ANQA/wDTAP8A0wD/ANMA/wD/ABgE/wAVA/8A
FAP/ABsC/wAkAv8ALwH/ADsB/wBIAPgAVADzAF8A7gBpAOsAcQDpAHkA6AB/AOYAhQDIAIsA
5ACQAOMAIQDhAJSA4ACHAN4ApwDdAK4A2wC2ANkAwQDVAM4A0gDfAM8A7QDNAPcAywD/
AMoA/wDJAP8AyQD/AMkA/wD/ABUE/wARA/8AEwL/ABkC/wAhAf8AKwH/ADcA+ABDAPEATwDr
AFoA5gBkAOMAbQDhAHQA3wB7AN4AgQDcAlcA2wCNANKAkGdYAJgA1gCeANQApADSAKsA
0AC0AM4AvgDLAMwAyQLeAMYE7ADDBvoAwQf/AL8l/wC+Cf8Avgn/AL4J/wD/AxAD/wEJAv8ID
wH/DBYB/wseAf4IJwD1BJEA7gQ9AOgCSQDhAIUA3QNfANoEaADYBXAA1QV3ANMGfQDSB4M
A0AeJAM8HjgDNCJQAzAibAMoJoQDlCqkAxguyAMQMvQDCDcwAvw7fALwR7wC5E/sAthT/ALU
U/wC0FP8BtBT/AbQU/wH/CAkC/woFAf8SCwH/FREA/xUYAPQSIQDrDisA4w02AN0MQwDXDU4
A0w5ZANAQYgDNEmoAyxNyAMkTeADHFH8AxRWEAMQWigDCFpAAwReXAL8YngC9GKYAux
mvALkaugC3GsgAtRvbALEd7ACvHvkBrR7/Aqse/wKpH/8CqR//Aqkf/wL/DAMC/xQCAP8aBgD/HA
sA9hsRAOoYGGDgFSMA2BQvANEYPADLG0gAxxxTAMQeXQDBH2UAvyBtAL0gcwC8IXoAuiG
AALkihG3lowBtiOTAbQjmgGzJKIBsSSrAa8ltgGtJcQBqybVAagm6AKIJ/YDoij/BKEo/wSgJ/8FoCf
/BaAn/wX/EQAB/xwAAP8hAADxlgIA5x8HAN8bDgDUHBsAzCEpAMYkNgDAJkIBvCdOAbkoVwK
3KWACtSlInArMqbgKxKnUCsCt7Aq4rgQOtLlgDrCyOA6oslgOoLZ4Dpy2nA6UusgOjLr8EOs7RBJ4
v5QWbL/QGmS//B5gu/weXLv8Hly7/CJcu/wj/GAAA/yIAAO4nAADfKgAA2CsBANInBwDKKBUAwi
wkALwvMQG3MD0CsZFIA7AxUgStMVoEqzJiBakyaQWoMnAFpjN2BqUzfAajM4MGojSKBqA0kga
fNJoGnTWkB5s1rgeZNbsHlzXMCJU14QiSNfEKkDX9Co81/wuONP8LjTT/C400/wv/HgAA9ygAAO
lvAADWNQAazjcAAMg0AwDBMhAAuTYfALM4LAGuOTgDqjIDBac5TAakOVUHoJldCKA5ZAieO

WsJnTlxCZs5dwmaOn4KmDqFCpc6jQqVOpYKkzqgC5E6qwuQO7cMjTvIDIs73QyJOu8NhZr8Do
U6/w6FOv8OhDr/DoQ6/w7/JAAA6S0AANs4AADPPgAAxkAAAL8+AAC5OwoAsj4aAKxAJwGnQD
MDo0A+Bp8/RwicP1AJmT9YC5c+XwuVPmYMID5sDZI+cw2RPnoNjz6BDo4+iQ6MPpIPij+cD4g/p
w+GP7QQhD/EEII/2hF/P+0Rfj/6En0+/xJ8Pv8SfD7/Enw+/xL9KAAA5DUAANU/AADJRgAAwEgAA
LhFAACyQgYArEQWAKZGJAGhRy8DnEY5BphFQwmVREwLkkNUDY9DWw6NQmIPi0JoElpCb
xGIQnURhkJ9EoVChRKDQo4TgUOZE39DpBR9Q7EUe0PBFXID1hV3QuoVdUL5FXRC/xV0Qv8
VdEL/FXRC/xXyLAAA3zsAANBFAADETAAAU08AALNMAACtSQIApkoSAKBMIAcBTswDlkW2Bp
JKPwmOSUGmi0hPDohHVxCGR10Sg0ZkE4JGaxSARnEVfkZ5FXxGgRZ6RosXeEaWF3ZGoRh0
Rq4Ycka+GXBG0hlvRugZbUb3GW1F/xhsRf8YbEX/GGxF/xjrMQAA2kAAAMtLAADAUGaAtI MA
K5SAACoTgAAoVAOAjTSHQCWUikCkVEzBYxPPAmITkQMhE1MD4FMUx+F+S1oTfEpgFXpKZxZ
4SW4Xdkl1GHRJfhlySYgacEmSG25JnhtsSascakm7HGhJzxxnSeYcZkn2G2VJ/xtlSf8aZUn/GmVJ
/xrnNgAA1kUAAMdQAAC8VgAAslgAAKIWAACjUwAAAnFQLAJZWGgCRVyYBjFYwBYdUOQiDU0
EMf1FJD3tQUBJ4T1YUdU5dFnNNYxhwTWoabk1yG2xMexxqTIQdaU2QHmdNnB5ITakfY024H2
FMzR9fTOQfX0z0HI9M/x1fTP8cX0z/HF9M/xzkOwAA0koAAMRUAAAC5WgAAr1sAAKVaaACeVw
AA1glAJJaFwCNWyMBh1otBIJZNgd+Vz4LeVZGD3VUTRjyU1MVb1JaF2xRYBlqUWcbZ1BvHW
VQeB9jUllgYVCNIGBQmSFeUKchXFC2IVpQyiFZUOIhWU/zIFIP/h9ZT/8eWU//HIIP/x7hPwAAz04
AAMFYAAC3XgAAq18AAKJeAACaWgAAk1wFAI1dFACIXyEAgl4rA35dNAZ5XDwKdFpDDnBZS
hJsV1AVaVZXF2ZVXhpjVWUcYVRsH19UdSBdVH8hW1SKIIIUlyNXVKUjVIS0I1RTyCNTU+AiU1
PyIVNT/SBTUv8fU1L/H1NS/x/dRAAAy1IAAL5cAAC0YgAAqGIAAJ5hAACWXgAAj2ACAlhEQCE
Yx4AfmMoAnliMQV0YDoJcF9BDWtdSBFnXE4UZftVF2FaWxpeWWldW1lqH1IYcyFXWHwjVvill1
NYISRSWKmkUFizJE5YxiNNV98jTVfxlk1W/CFOVv8gTlb/IE5W/yDZSAAAYFYAALtgAACwZQAA
pGUAAJpIAACSYgAAimQAAIRIDQB/ZxsAemcmAXVnLwRwZTcla2Q/DGdjRhBjYkwUX2FTF1xf
WRpZXmAdVl5oIFNdcCJRXXojT12GJE1dkyRMXalkS1yxJElcxSNIXN0jSFvwlkhb/CFIWv8gSFr/I
Eha/yDVTQAAXFsAALhkaACtaAAAoWgAAJdoAACNZgAAhWgAAH9qCgB5axgAdWwjAHBsLQN
razUH22o8C2JpQw9eaEoSWMZQFIdVxIUZF4cUWRIH05jbiFMYngiSmKEI0hikSNHYqAjRmKvIk
RiwyJDYdwhQ2DvIUNg+yBDX/8gQ1//IENf/yDRUGAAwV8AALVoAACpawAAAnWwAAJNsAAClAw
AAgG0AAHluBgBzcBQAb3EgAGtxKgJmcTIFYnA6CV5vQQ1abkcQVm1OFFJsVBdPa1saTGpjHU
lqax9HaXUGRWmBIUNpjyFCaZ4hQWmuleBpWR8/aNofPmfuHz5m+h8+Zf8fPmX/Hz5l/x/MVwAAv
WQAALJtAAClbwAAmXAAAI5wAAECeAAAenIAAHN0AgBtdRAAAaXcdAGV3JwFhdzADXXc3B1I2
PgpVdUUOUXRLEU5zUhRKc1kXR3JgGkRxaRxCcXMDQHf/Hj9xjR4+cZwdPXGsHDxwwBw7cN
kbOm7tGzlt+Rw5bP8cOWz/HDIs/xzGXgAAuWoAAK5yAACgcwAAIXQAAIp0AAB/dQAAC3cAAGx
6AABmfAsAYX0YAF5+lwBbfwCV340BVN+OwhQfUILTHxJDkI7TxFGelYTQ3peFkB6Zxc+eXEY
PHI9GTt5ixk6eZsYOXmrGDh5vhc2eNcWNnfsFzV1+Rg0dP8YNHT/GDR0/xjBZQAAAtXAAAKI2AA
CbdwAAKHgAAIV5AAB6ewAAbH4AAGSAAABegwUAWYQTAFaFHgBUhigBUiYwA02GOAVKhT
8IRoVFCkSETA1BhFMPPoNbETyDZBI6g28TOIN7EzeDiRM2g5kTNIOPeJOcVBEygtUQMYDrET
B++Blvff8TL33/Ey99/xO8bAAAsXcAAKR7AACWfAAAi30AAIB+AAB1gAAAaIQAAF6HAABVigAA
UlwMAE2NGQBLjiMASI4rAUaOMwJDjjoEQI1BBj6NSAg7jVAKOY1YCzeNYQw1jWwNM414DTK
Nhw0xjZYML4ynDC6MugstjNILLlrqCyul9wwqh/8NKof/DSqH/w22dAAARH4AAJ6AAACRgAAAhO
AAHuEAABvhwAAZloAAFiOAABPkQAAR5QFAEOVEgBAIhwAP5YIAD2XLQA7IzUBOZc8AzaXR
AQ0I0sFmPZUBjCWXgcvlmgHLZd1ByyXgwcrI5MHKZekBiiWtwYmls8FJpToBSST9gYjkf4HI5H/C
COR/wixfQAApoQAAJiFAACMhgAAgYgAAHWLAABpjgAAXpIAAFOWAABImQAAP5wAADieCAA
0nxMAMqAcADGgJQAwoC0ALqE0ACyHPAEroUQBKaFNaiihVwlmoWIDJaFvAyShfQlioY4Clagf

Ah+hsgEdoMkBHZ/kARyd8wlbnP0DG5z/Axuc/wOshgAAn4oAAJOLAACHjAAAEo8AAG+TAABJlw
AAWJoAAE2eAABCogAAOaQAADCmAAAOqAcAJKkRACOqGgAiqlAlaoqACCrMgAfqzoAHatDA
ByrTgAbrFkAGqxmABisdQAXrIYAFayYABOsqwARq8IAD6reABCp8AAQp/sAEKf9ABCn/QCnjwA
AmZAAAI2RAACAIAAAAdJgAAGicAABcoAAAUaQAAEanAAA8qgAAMq0AACmvAAAhSQAAGrICA
BS0CwAStRQAEbUdABC2JAAPti0ADrY2AA23QAALt0wACrdaAAi3aQAHt3oABbeMAAO3oAAAttr
MAALXJAAC04AAAs+4AALPwAACz8AChlgAAIJcAAleaAAB6ngAAbalAAGGnAABVqwAASa4AA
D6wAAA0sgAAKrQAACG3AAAZuQAAE7sAAAU9AQAfVwgAAcARAADAGAAAwSAAAMlpAADc
MgAAwz4AAMRLAADEWgAAxGsAAMR9AADEkQAAXKQAAMO3AADCyQAawtwAAMLgAADc4
ACbngAAjaAAAICkAAB0qQAAZ60AAFiwAABLsgAAP7UAADS4AAAqugAAIb0AABm/AAARwgAA
CsQAAAPHAAAAyQAAMMoFAADLDAAAZRQAAM4bAADQJAAA0S0AANM5AADVRwAA1VgAA
NZrAADWfwAA1ZMAANWIAADVtAAA1cMAANXHAADVxwCUpgAAh6oAAHmuAABqsQAAWrQA
AEy3AAA/ugAAM70AACnAAAAgwwAAGMYAABDJAAAIZAAAAc8AAADRAAAA0wAAANUAAAD
YAAAA2gYAANwNAADeFQAA4R4AAOMoAADnNAAA6kMAAOpVAADqaQAA6n0AAOqRAADpo
QAA6a0AAOmVADprwCKrAAAFLaAG2zAABdtwAATroAAEC+AAA0wgAAKcYAAB/KAAAWzg
AADdEAAAXVAAAA2AAAAANoAADcAAAA3gAAAOEAAADjAAAA5QAAAOcAAADqBQAA7Q0A
APAXAADzlgAA9i4AAPk+AAD6UQAA+2QAAPt3AAD7hwAA+5QAAPuXAAD7lwD/ABsD/wAYAv8
AFwL/AB0B/wAoAP8AMwD/AEAA/wBNAP0AWQD6AGMA+ABsAPYAdAD0AHsA8wCBAPIAhwD
xAlwA8ACRAO8AlgDuAJsA7AChAOoApgDoAK0A5QC1AOMAvwDiAMwA4ADdAN4A7ADdAPcA
2wD/ANkA/wDXAP8A1gD/ANYA/wD/ABcD/wAUAv8AEwL/ABsB/wAIAP8AMAD/ADwA/ABJAPcA
VQD0AF8A8gBoAO8AcQDuAHgA7AB+AOsAhADqAlkA6ACOAOCakwDmAJgA5ACeAOIAowDfA
KoA3QCxANsAuWdZAMgA1gDZANQA6ADSAPUA0AD/AM4A/wDNAP8AzAD/AMwA/wD/ABQD/
wAPAv8AEAH/ABkB/wAiAP8ALAD8ADgA9ABFAPAAUADtAFsA6gBkAOcAbADIAHQA5AB6AOIA
gADhAIUA3wCKAN4AjwDcAJUA2gCaANgAoADUAKcA0gCuANAAuADOAMQAzADUAMoA5QD
HAPMAxgD8AMQA/wDDAP8AwgD/AMIA/wD/ABAC/wAKAf8ADgH/ABYA/wAeAP8AKAD0ADMA6
wA/AOgASwDkAFYA4QBfAN4AaADcAG8A2gB2ANgAfADWAIEA1ACGANIAjADQAJEAzgcXAM
wAnQDKAKQAYACsAMYAtQDEAMEAwgDRAMAA4wC+APEAvAD6ALoA/wC5AP8AuAD/ALgA/
wD/AAoC/wAFaf8ACwD/ARIA/wAaAPgAlgDrAC0A4wA5AN4ARQDaAFaa1gBaANMAYwDQAGo
AzgBxAMwAdwDKAH0AyQCCAMcAiADFAI0AxACTAMIBmgDAaAEvwGpAL0CswC7A78AuAT
PALYG4gC0CPIAsQn+ALAK/wCvCv8Argv/AK4L/wD/AAMB/wMCAP8JBgD/CgwA+gcUAO4CHA
DfACYA2QEzANQDPwDPBEoAywVUAMgHXQDGB2UAwwhsAMEKcgDACngAvgt+AL0MhAC8D
loAug2QALkNIwC3Dp8AtQ+oALQQsgCyEb8ArxHPAKwT4wCqFfMApxX+AKUX/wCkF/8Aoxf/AK
MX/wD/BAAA/wwAAP8RAAD3EQQA7w0KAOMIEwDXBx4A0AsRAMoOoADFEkQAwRNOAL0VV
wC7FmAAuRZnALcXbgC2GHQAtBh6ALMZfwCxGYyAsBqMAK4akwCtG5sAqxykAKkcrgCnHbsA
pR3LAKMe3wCglPABnSD8AZwg/wGbIP8Bmh//Apof/wL/CwAA/xQAAPExAADiGAAA2hcCANUR
CQDNExcAxRglAL8cMgC6Hj4Ath9JALMgUgCxIVsAryJiAK0iaQCsl28AqiN1AKkkewCoJIEApiWIA
KUIkAGjJpgBoSahAZ8nqwGeJ7cBmyjHAZko2wGWKO0CICj7A5Mo/wOSJ/8DkSf/BJEn/wT/EwAA
9hoAAOMhAADZJgAAzyYAAMkiBADDIBEAuyQgALUnLQCxKDgArSIDAaoqTQGoK1YBpitdAqQr
ZAKiLGoCoSxxAp8sdwKeLX0CnC2EapsujAOZLpQDmC6dA5YvqAOUL7QDki/DBJAv2ASNL+sFi
y/5Bokv/walLv8GiC7/Bogu/wb/GQAA6SIAANwsAADPMgAAxjIAAL8vAAC6KwsAsy4bAK0wKACp
MTMBpTi+AqlySAOfM1EDnTNYBJszXwSZM2YFmDNsBZYzcgWVM3kFkzSABZIOiAaQNJAGjjS
aBow1pQaLNbAHIDxAB4Y10weENeglgjX3CYA1/wl/NP8JfzT/CX80/wn3HgAA4ywAANQ2AADIO
wAAvjwAALc4AACyNAYArDYWAKY4lwChOS8BnTo5A5o5QwSXOUwFIDIUBpl5WweQOWEHjzl

nCl05bgiMOXQlilj8CYk5gwmHOYwJhTqWCoM6oQqBOq0Lfzq8C306zwt7OuUMeTr1DHc5/wx3O
f8Mdjn/DHY5/wztJAAA3TMAAM49AADQCgAAuEIAALA/AACrPAEApT0SAKA/HwCbQCsBlkA1A5
M/PwWPPkcHjT5PClo+VgmlPV0Khj1jC4U9aguDPXAMgT13DH89gA1+PokNfD6TDno+ng54Pqo
Odj65D3Q+zA9yPuMQcD30EG89/g9vPf8Pbj3/D249/w/pKwAA2DoAAMIDAAC9SAAAskGAAKpFA
ACIQgAAAn0MNAJpFHACVRicBkEUxA4xEOWalQ0MIhUNLCoNCUguAQIkMfkFfDXxBZg56QWwP
eUF0D3dBfBB1QYURc0GQEXFCmxJwQqgSbK2EmtByRNqQeATaEHyE2dB/RJnQf8SZ0D/Em
dA/xLIMQAA0z8AAMRIAAC5TQAArU0AAKZLAACgRwAAmUgKAJRJGACPSiQBi0ouA4ZJNwaC
SEAlf0dHCnxHTgx5RIUOd0VbD3VFYhBzRWkRcUVwEm9FeBNtRYIUa0WNFGIFmBVORaUVZk
W0FmRFxhZiRd4WYUXwFWBE/BVgRP8UYET/FGBE/xThNgAAz0QAAMFNAAAC0UAAAqVEAAK
FPAACbTAAAI0GA19OFQCKTyEAhU8rAoFONAV9TTwleUxEC3ZLSw1zSIIpCEIYEG5JXxJrSG
YTaUhtFWdldRZmSH8XZEiKF2JIIhhgSaMYXkixGFxlxBIBSNwZWkjvGFPl+xdaR/8WWkf/FlpH/xb
dOwAAy0gAAL5RAACxVAAApIQA AJ1TAACXTwAAkFEDAlpSEgCFUx4AgFMoAnxTMQR3UTkH
c1BBCnBPSA1sTk4Pak1VEWdNXBNITGMVYkxqFmBMcxhfTHwZXUyHGltMkxpZTKEaWEyvGIZ
MwhtUS9obVEvuGVRL+hhUS/8YVEv/F1RL/xfAPwAAyEwAALtVAACtVwAAolcAAJIXAACSUwAAi
1QAAIVWDgCAVxsAfFgmAXdXLwRyVjcHblU+CmpTRQ1nUkwPY1FSEWFRWRRfUGAWXFBnG
FpQcBIYT3kaVICFG1VQkRxTUJ8cUICuHFBPwBxPT9gcTk/tG05O+hpOTv8ZTk7/GE5O/xjVQwA
AxFAAALhZAACqWgAAAn1sAAJZaAACOVgAAh1gAAIFaCwB7WxkAd1wjAXJbLANuWjQGaV8C
WVYQwxiV0KPIXQE VxVhRZVV0WV1RIGFRUbRpSVHcbUVSCHE9Ujx1NVJ0dTFSShUtUvhx
JU9UcSVPrG0IS+RpJUv8ZSVL/GUIS/xnSRwAAwVQAALVcAACnXQAAAnF4AAJJdAACKWgAAgl
wAAHxeCAB2XxYAcMAhAG1gKgJpXzIFZV46CGFeQQtdXUcOWIxOEvdBVRUWIsWUVIjGE9Z
axpNWXUcS1iAHEIzJR1IWZsdR1mrHEVYvRxEWNQcRFfqG0RX+BpEVv8aRFb/GURW/xnOTAA
AvlgAALFfAACkYAAAmGEAAI5hAACFXgAAfWAAAHZiQBxYxIAbWUeAGhIjWfKZTAEYQG3B1
xjPgpZYkUNVWFLEFJhUhNPYFkVTF9hGEpfaRIIXnMbRI5+HEReixDXpocQI6pG0FeuxtAXdlaP
1zpGj9c9xk/W/8ZP1r/GT9a/xnJUQAAu10AAK1iAACgZAAAIQAAIpkAACAYwAAAd2UAAHBnAQ
BraQ4AZ2oaAGNrJABfay0CW2o1BVdqPAhUaUILUWhJDk1nTxFKZiYUSGZeFkVIZxhDZXAZQW
V8GkBliRo/ZZgZPWWoGTxkuRg7ZNEYO2PoGDph9xg6Yf8XOmD/Fzpg/xfFVwAAuGIAAKImAAC
cZwAAkGgAAIZoAAB7aQAACWoAAGpsAABkbgoAYHAXAF1xIQBZcSoBVnEyBFJxOQZPcEAJT
G9GDElvTQ5GblQRQ21cE0FtZBU+bG4WPWx6FjtshxY6bJYWOWymFThsuBU3bM8UNmrnFDV
p9hU1aP4VNWf/FTVn/xXAXQAAtGcAAKVqAACXawAAjGwAAIJtAAB2bgAAanAAAGNzAABddA
UAWHYSaFV3HQBTeCcAUHgVak14NgRKeD0HR3dECUR2SgtBdlENPnZZDzx1YhE6dWwSOH
V4Ejd1hRI2dZQRNXWIETR0thAydM0QMnPMEDFx9REwcP4RMG//ETBv/xG7YwAAr20AAKBuA
ACTbwAAh3AAAH1xAABycwAAZXYAAfX5AABVewAAUH0MAE1+GABLfYIASYArAUaAMgJDgD
kEQX9ABj5/Rwg8f04JOn5WCzd+Xww1fmkNNH51DTJ+gw0xfpIMMH6jDC9+tAstfcsLLXzICit69A
wreP0MKnj/DSp4/w23agAAqXEAAJpzAACOdAAAAG3UAAHh2AABteQAAYHwAAFeAAABOgwAA
SIUGAESGEgBChxwAQIglAD6ILQA8iDQBOog8AzilQwQ2iEoFM4hTBjGIXAcwiGYILohyCC2lgAg
riJAHKoegBymHsgYnh8kGJ4XjBSaD8wclgVwIJIH/CCSB/wiycgAAo3YAAJv3AACJeAAAfnoAAHN
8AABofwAAXIMAAFGHAABJigAAQI0AADqPCgA3kBUANpAfADSRJwAzkS4AMZE2AC+RPQEuk
UUCLJFOAiQRVwMpkWIDJ5FuAyaRfAMkkYwDI5GdAiKRwlgkMUCH4/gAR6N8QMejPsDHYz/B
B2M/wSsewAAAnHwAAI99AACEfgAAelAAAG2DAABihwAAV4oAAEyoAABcKQAAOpQAADGXA
QArmQwAKZkWACiaHgAnmiYAJZotACSbNQAJmz0AlptGACCBUAafm1sAHptoABycdgAbnIYAG
ZuXABebqgAVmr8AFJnaABSY7wAUlvkAFJb/ARSW/wGlgQAAIoEAAIqCAAB+hAAAcocAAGeLA
ABbjwAAUZMAAEaWAAA8mgAAM50AACufAAAjoQAAHaMKABqKEwAZpBsAGKQjABeIKgAWp

TIAFaU7ABSIRgASpIEAEaZeAA+mbAANpn0AC6aPAAmloQAHpLUABaPMAAWi4wAFovAABaH
3AAWh9wCdhwAAkYgAAIWJAAB4jAAAbJAAAGCUAABVmAASpWAAAD+gAAA2owAALKYAAAC
SoAAAcqgAAFawAAA6tBgAJrg4AB64XAAavHwAFrycABK8vAAOvOQACsEUAALBSAACwYAAA
sHAAALCCAACvIQAAr6gAAK68AACT0gAAreMAAKzsAACs7ACXjQAAi44AAH6RAABylQAAZZo
AAFmeAABOowAAQ6cAADmqAAAvrQAAJa8AAB2xAAAVsgAADrQAAAe2AAAAtwYAALcNAAC
4FQAAuRwAALkAAC6LQAAuzcAALxDAAC8UgAAvGIAAL1zAAC9hwAAvJoAALytAAC7vwAAu9
AAALrbAAC62wCSIAAAhZcAAHibAABroAAAXqUA AFKqAABHrgAAOrAAADCzAAAmTQAAHbcA
ABS5AAAMuwAABr0AAAC/AAAAwAAAAMEDAADCCgAAwxEAAMUZAADGIAAAxykAAMk0AAD
MQQAAzFEAM1jAADOdGAAzooAAM6dAADOrQAAzroAAM3DAADNwwCMnQAAf6IAAHKnAAB
lrAAAV7AAAEiyAAA8tQAAL7gAACW7AAAcvgAAFMAAAAvCAAExQAAAMcAAADJAAAAyWAA
AM0AAADPAAAA0QMAANMKAADVEgAA1xsAANoIAADdMAAA4D8AAOFRAADiZAAA43cAAO
OLAADkmwAA5KcAAOSuAADkrgCGqAAAEa0AAGqwAABaswAASrcAAD26AAAwvgAAJcEAABz
EAAASyAAACcoAAAHNAAAA0AAAANMAAADVAAAA2AAAANoAAADcAAAA3wAAAOEAAADkA
gAA5woAAOoTAADuHgAA8SsAAPQ7AAD0TgAA9GMAAPV2AAD1hgAA9plAAPaaAAD2mgD/A
BcC/wAUaf8AEwH/ABsA/wAIAP8AMQD/AD4A/wBKAPwAVQD5AF8A9gBoAPQAcADzAHcA8QB
9APAAggDtAlcA6wCMAOoAkADoAJUA5gCbAOUAoQDkAKcA4gCvAOEAuQDfAMUA3QDVANs
A5wDYAPQA1QD+ANUA/wDUAP8A0wD/ANMA/wD/ABMC/wAPaf8AEAD/ABkA/wAiAP8ALgD7A
DoA+ABGAPUAUQDyAFsA7wBkAO0AbADrAHMA6QB5AOcAfgDkAIMA4gCIAOAAjQDfAJIA3gC
YANwAngDbAKQA2QCsANYAtQDUAMEA0gDQAM8A4gDNAPAAyWd7AMoA/wDIAP8AxwD/AM
cA/wD/AA4C/wAJaf8ADQD/ABYA/wAfAPsAKQD0ADUA8ABBAO0ATADpAFcA5gBgAOQAaADh
AG8A3gB1ANwAegDaAH8A2ACEANYAiQDUAI8A0wCUANEAmgDPAKEAzgCpAMsAsQDJALw
AxwDLAMQA3QDCAO0AwAD5AL4A/wC+AP8AvQD/AL0A/wD/AAkB/wAEAP8ACgD/ABIA/wAaA
PEAJQDrADAA5wA8AOMARwDfAFIA3ABbANKAYwDVAGoA0gBwANAAAgDOAHsAzACAAMsA
hQDJAlsAyACQAMYAlwDFAJ4AwwCIAMEArgC+ALkAvADHALoA2QC3AOoAtgD2ALQA/gCzAP
8AswD/ALMA/wD/AAMB/wABAP8ABwD/AA0A9wAVAOgAHwDiACoA3QA2ANgAQgDTAEwAzWb
WAMwAXgDJAGUAxgBrAMUAcQDDAHcAwgB8AMAAgQC/AlcAvQCNALwAkwC6AJAoAuACiALY
AqwC0ALYAsgDEALAA1gCuAOgArAD0AKsA/gCpAf8APwL/AKcC/wD/AAAA/wAAAP8AAQD+AA
cA7QANAN8AGADYACQA0QAwAMwAOwDHAeyAxABQAMEAWAC+AGAAvABnALsAbQC5AXI
AuAF4ALYcfQC1AoMASwOJALIEkACwBZgArwWgAK0GqQCrB7QAqQfDAKcJ1QCICukAogz3A
KAN/wCfDf8Ang3/AJ4N/wD/AAAA/wQAAPYGAADmAwAA3wAEANYAEQDOAR0AyAMoAMIGNA
C9CD8AugpKALcLUgC1DFoAsw1iALENaACwDm4Arg90AK0QeQCsEH8AqhGGAKkRjQCNEpU
ApROeAKQTpwCiFLMAoBXBAJ0V1ACaF+gAmBj3AJYY/wCVGP8AIRf/AJUX/wD/BQAA+AsAAO
URAADbFQAA0hMAAMwMBgDFChQAvhAhALgTLgC0FTkAsBdEAK0YTQCrGVUAqRIIdAKcaYw
CmGmkApRtvAKMbdQCihHsAoByCAJ8diQCdHpEAmx6aAJofpACYH7AAIH++AJQg0QCRIOYAj
yH1AY0h/wGMIP8BiyD/AYsg/wH/DAAA6hUAAN0eAADRIgAAxiEAAL8dAQC8GA0AtRwcAK8fKA
CrIDQApyl/AKUjSACil1AAoCRYAJ4kXwCdJWUAMYrAJolcQCYJncAlyZ+AJUmhQGUJ44BkieX
AZAnoQGPKK0BjSi7AYoozQGIKOMChijzAoQo/gODKP8Dgij/A4Io/wPzEgAA4yEAANQpAADHLg
AAvCwAALYpAACylwgArCUXAKcoJACjKi8Anys6AJwsQwGZLEwBlyxTAZUsWgGULGEckixnAp
EtbQKPLXMCji16AowtgQKLLooDiS6UA4cungOFL6oDgy+4BIEvygR/L+AEfC/xBXsv/QV6Lv8FeS
7/BXku/wXsHAAA3CoAAMwzAAC/NgAAtTUAAK4yAACqLgIAPs4SAJ8wHwCbMisAlzM1AZQzPg
KRM0cCjzNPA40zVgOLM1wEiTNIblgzaQSGM28FhTN2BYMzfgWBNiYFgDSQBn40mwZ8NacGe
jW1B3g1xgd2NN0HczTwCHI0/AhxM/8lcTP/CHEz/wjnJAAA1TEAAMY6AAC5PAAArjwAAK5AAC

jNgAAanjUNAJg3GwCUOSYAkDkxAY05OgKKOUMEhzhKBYU4UQWDOFgGgTheBn84ZAd9OGs
HfDhyCHo4egh4OIMldzmNCXU5mAlzOaQKcTmyCm85wwptOdoLaznuC2o4+wtpOP8KaTj/Cmk4
/wriKgAA0DgAAMJAAAC0QQAaqUEAAKI/AACcPQAAlzwJAJI9FwCOPiMAij4tAYY+NgODPj4Eg
D1GBn09TQd7PVQleTxaCXc8YQI1PGcKczxuCnl8dgtwPH8Mbj2KDGw9lQxqPalNaD2vDWY9w
Q5kPdcOYzzsDml8+Q1iPP8NYTz/DGE8/wzeMAAAyz0AAL1EAACvRgAApUYAAJ1EAACXQQA
AkkEFAI1CEwCIQx8AhEMpAYBDMgN8QzsFeUJCB3ZBSQh0QVAJcUBWCm9AXQtQGQMa0Br
DWpAcw5oQHwOZkCHD2RAkg9iQZ8QYUGtEF9AvhBdQNQRXEDqEFtA+A9bP/8PWz//DIs//w7Z
NQAAx0IAALIIAACrSgAAoUoAAJIIAACTRQAAjUUBAIdGEACCRxwAfkGmAxpILwN2RzcFc0Y/B
3BFRgltRU0KakRTC2hEWg1mRGEOZERoD2JEcBBgRHkRX0SEEV1EkBJbRJ0SWkSrE1hEvB
NWRNITVUPpEIVD9xFVQ/8RVEP/EFRD/xDVOgAAxEYAALVLAACoTQAAnU0AAJVMAACPSQ
AAiEkAAIJKDAB9TBkAeUwkAHVMLQJxSzQEbUo8B2pKQwlnSUoKZEhQDGFIVw5fR14PXUdlE
VxHbRJaR3cTWEeBE1ZljhRVSJsUU0ipFFJlUhrQR9AUT0fnFE9H9hNPRv8ST0b/EU9G/xHRPg
AAwUoAALJPAACkUAAAmIEAAJFPAACKTAAAg00AAH1OCQB4TxYAdFAhAHBQKgJsUDIEaE
85BmROQAhhTUcKXkxNDFxMVA5ZTFsQV0tiEIVLaxNUS3QUUkt/FVBLixVPS5kWTUuoFkxLuB
ZKS84VSUvmFUIK9hRJSv8TSkn/EkpJ/xLNQgAAvk4AAK5SAACHUwAAIIMAAI5TAACGTwAAf1E
AAHhSBgBzUxMAb1QeAGtVJwFnVC8DY1Q3BV9TPQhcUkQKWVFLDFZRUC5UUFgQUIBgEIB
QaBROT3IVTE99FktQiRZJUJcWSFCmFkZQtZFT8wWRE/IFURO9RRETv4URE3/E0RN/xPKRw
AAu1IAAKtVAACeVgAAk1YAAIpWAACBUwAAelUAAHNWAwBuWBAAalkbAGZZJQBiWS0CXIk0
BVpYOwdXV0IJVFdIDFFWTw5PVVYQTVVeEkpVZhRIVG8VR1R6FkVUhxZEVJUWQ1SIFkFUtR
VAVMsVP1PkFT9S9BQ/Uv0TQFH/E0BR/xPGSwAAuFUAACKhXAACaWQAakFkAAIZZAAB8VwA
AdVkAAG5bAABoXAwAZF0YAGBelgBdXisBWV4yBFZeOQZTXUAIUFxGC01cTQ1KW1QPSFtb
EUVaZBNDWm0UQlp4FUBahRU/WpQVPlqjFT1atBQ8WckUO1niEztX8xM7V/0TO1b/EjtW/xLCU
AAAtFkAAKRbAACXAAAajF0AAIJdAAB4XAAAb14AAGhgAABiYQkAXmMVAfPkHwBXZCgBVG
QvAlFkNgVOYz0HS2NECUhiSwxFYIIQO2FZEEFhYhE/YGsSPWB2EzxxgM7YJITOWCiEjhgshl
3YMgRNl/hETZd8xE2XPwRNVz/ETVc/xG+VQAAR10AAKBeAACTXwAAiGAAAH5hAABzYQAAa
GMAAGFIAABcZwQAV2kRAFRqGwBRaiQATmssAUtrNANJajoFRmpBB0NpSAIBaU8LPmlXDTx
oXw46aGkPOWh0EDdogRA2aJAQNWigDzRnsQ8zZ8YOMmbgDjFk8g4wY/wPMGL/DzBi/w+6W
wAAqmEAAJtiAACPYwAAg2QAAHIIAABuZgAAYmkAAfprAABVbQAAUG8MAE1wFwBKcSAASH
lpAEZyMAJDcjcDQXI+BT5xRQc8cUwlOnFUCjdwXQs1cGcMNHByDDJwfwwxcl4MMHCeCy9wrw
sub8QKLW7fCixs8Qsra/sMK2r/DCtq/wy2YgAApWUAAJZmAACKZwAAf2gAAHRqAABpbAAAXm
8AAFZxAABOdAAASHYGAER4EgBCeRwAQHkkAD96LAA9ejMBO6Ajh6QQQ2ekkFNHIRBjJ5
WgcweWQIL3lvCC15fQgseYwHK3mcByp5rQYoeMIGJ3fdBSZ18AcldPoIJXL/CCVy/wixaAAAn2o
AAJFrAACFbAAAem0AAHBvAABlcQAAXWUAAFB4AABlewAAQH4AADyACwA5gRYAN4EfADa
CJwA0gi4AM4I1ATGDPQEvGkQCLYJNAyuCVgMqgmAEKIJsBCeCeQQmgokDJIKZAYOCqwMhg
b8CIIDaAiB/7gMffkEH3z/BR98/wWqbQAAMW8AAIxxAACAcQAAdnlAAGt0AABgeAAAVXsAAEt/
AABDggAAO4UAADKIBAAuiQ4ALloYACuKIAApiycAKIsvACeLNgAmjD4AJYxHACOMUAEijFsBII
xnAR+MdAEdlQBHlyVABqLpwAZi7sAF4rVABel7AAXh/gBFob/ARaG/wGicwAAk3QAAlD1AAB8d
gAAcHgAAGV7AABafwAAUIMAAEWHA9igAANI0AACyPAAAKkgUAIJMPAB+TGAAelB8AHZQ
mABuVLgAalTYAGZU+ABiVSAAXIVMAFZVgABSVbQASIX0AEZWPA6VoQAMILQACpPMAAq
S5AAKkfMACpD8AAqQ/ACbeQAAjXkAAIJ6AAB2fAAAan8AAF+DAABUhwAASosAAECPAAA2kg
AALpUAACaXAAAemgAAF5wEABKdDAAQnhUADp4cAA2fJAAMnywAC580AAqfPgAJn0kAB59
WAAafZAAEn3MAAp+FAACelwAAAnqoAAJ2/AACc1gAAm+gAAJryAACa8gCUfwAAiH8AAHyBAA

BwhAAAZIlgAAFiMAABNkQAAQ5QAADmYAAAwmwAAJp4AAB+hAAAXowAAEaQAAAmmbAAD
pwwAAKcUAACoGwAAqCIAAKkqAACpMwAAqT4AAKIKAAACpWAAAqWgAAKI5AACpjAAaqZ8A
AKiyAACnxgAAp9kAAKbnAACm5wCOhQAAg4YAAHaJAABpjQAAXZIAAFKWAABGmwAAPJ8A
ADKiAAAppQAAIKgAABiqAAAQrAAACa0AAAKvAAAAAsAQAALELAACxEgAAshgAALMgAACzKA
AAtDEAALU8AAC1SgAAtloAALZrAAC2fgAAtZIAALWIAAC0twAAtMcAALTXAAC01wCJjAAAfI8A
AHCTAABjmAAAVp0AAEuiAAA/pgAANKoAACutAAAhSAAAGLEAABCzAAAIItQAAAbYAAAC4AA
AAuQAAALoBAAC7BwAAAvA0AAL4UAAC/HAAAwCQAAMluAADFOgAAxUoAAMZbAADGbgAAx
4IAAMaWAADGqAAAxrUAAMbCAADGwgCDIQAAdpkAAGmeAABdpAAAUkKAAESuAAA4sQAA
LLMAACG2AAAYuAAAD7oAAAa8AAAAvgAAAMAAAADCAAAAwAAAMUAAADGAAAAyAEAA
MoIAADMdWAAzhcAANEgAADVKwAA2jgAANtJAADcXAAA3XAAAN6EAADelgAA3qMAAN6uAA
DergB9nwAAcaUAAGSqAABWwAAAR7MAADm2AAAsuQAAIbwAABe/AAAOwgAABcQAAADHA
AAAYAAAAMsAAADNAAAAzwAAANIAAADUAAAA1gAAANkAAADcAQAA3ggAAOESAADIHAAA
6igAAO42AADvSQAA8F0AAPFwAADyggAA8o8AAPKaAADymgD/ABMB/wAQAP8ADwD/ABgA/
wAjAP8ALwD/ADsA/gBGAPoAUQD3AFsA9QBkAPEAawDuAHIA7AB3AOoAfQDpAIIA5wCHAOY
AiWDIAJAA5ACWAOMAnADhAKIA4ACqAN4AsgDbAL4A2ADOANYA4QDUAPEA0gD9ANEA/wD
RAP8A0AD/ANAA/wD/AA4B/wAKAP8ADAD/ABUA/wAgAP0AKwD5ADcA9gBCAPMATQDvAFcA
7ABgAOgAZwDIAG4A4gB0AOEAeQDfAH4A3gCDAN0AiADbAl0A2gCSANgAmADWAJ8A1ACm
ANIArWDPALoAzADIAMsA2wDJAOWAxwD5AMYA/wDEAP8AwwD/AMIA/wD/AAgB/wADAP8ACQ
D/ABIA+wAcAPUAJwDxDIA7QA+AOoASADmAFMA4QBbAN0AYwDaAGoA2ABwANYAdQDUA
HoA0gB/ANEAhADQAlkAzgCPAMwAIQDLAJsAyACjAMYAqwDDALYAwQDDAL8A1AC9AOgAvA
D1ALkA/gC3AP8AtgD/ALUA/wD/AAMA/wAAP8ABgD/AA0A8wAXAOwAlgDnAC0A4gA4AN4AQ
wDZAE0A1ABWANAAXgDOAGUAzABrAMoAcQDIAHYAxwB7AMYAgADEAIUAWwCLAMEAkQC
/AJgAvQCfALsAqAC5ALiAtwC+ALQAZwCzAOMAsADwAK0A/ACsAP8AqwD/AKsA/wD/AAAA/wA
AAP8AAgD4AAgA6gASAOIAHQDcACgA1QAzAM8APQDLAEgAyABRAMUAWQDCAGAAwABm
AL8AbAC9AHEAvAB3ALoAfAC5AIEAtwCHALYAjQC0AJQASgCcALAApQCuAK4ArAC7AKoAyw
CoAN4ApQDvAKQA+QCjAP8AogD/AKIA/wD/AAAA/wAAPwAAADsAAEA4AALANYAFgDPACIA
yAAsAMQANwDAAEIAvQBLALoAUwC4AFsAtgBhALQAZwCyAG0AsQByALAAdwCvAH0ArQCD
AKwAiQCqAJEAqACZAKcAogCIAKwAogC4AKABYACeAt0AnAPtAJsD+QCZA/8AmQT/AJgE/wD/
AAAA+gAAAOgAAADeAAAA1QADAMwADwDFABoAvwAIALsAMQC2ADsAswFFALACTgCuA1U
ArARcAKsFYgCpBWgAqAZtAKcGcwCIB3kApAh/AKIIhgChCY4AnwqWAJ0KoACbC6sAmQy4AJc
MyQCVDD4AkW7wAJEO/ACQD/8Ajw//AI8P/wD/AAAA6wUAAN8NAADREAAAxw4AAMEHBQC9
AxIAtwcfALIKKgCuDTUAqg4/AKgQSACIEFAAoxFXAKISXgCgEmMANxNpAJ0TbwCcFHUAmhR7
AJkVggCXFYsAlhaUAJQWngCSF6kAkBe2AI4XxwCMGN0AiRnvAlcZ/ACGGf8AhRn/AIUZ/wDyB
QAA4xMAANUbAADFHgAAvBwAALYYAACzEQoArhMYAKkWJACIGC8AoRo6AJ4bQwCcG0sA
mhxSAJgcWQCXHV8AIR1IAJQdawCSHXEAkR53AJAefwCOH4cAjB+QAlogmwCIIKYAhyGzAIQh
xACCIdoAgCLtAX4i+gF9lf8BfCH/AXwh/wHrEQAA2x4AAMsmAAC9JwAAsyYAAK0jAACpHgMAp
R0TAKAgHwCclioAmSM1AJYkPgCTJEYAkSROAJAIVQCOJVsAjCVhAIsIZwCJJm0AiCZ0AIYme
wCFJ4QBgyeNAYEomAF/KKQBfSmwAXspwQF5KdYCYdjRAnUo+QN0KP8Dcyf/A3Mn/wPIGWAA
1CgAAMMuAAC1MAAArC8AAKYsAACHKQAAnSYNAJgoGwCUKiYAkSswAI4rOQCLLEIBiSxKA
YcsUAGFLFcBhCxdAYIsYwKALWkCfy1wAn0teAJ7LoACei6KA3gulQN2LqEDdC+uA3lvvgRwLtM
Ebi7pBGwu+AVrLf8Fay3/BWst/wXglwAAzTAAAL01AACvNgAApjYAAJ80AACaMQAALI4JAJEvFg
CNMSIAijIsAIYyNQGEMj4BgTJFAn8yTAJ9MIMDezJZA3kyXwR4MmYEJtBHQydARzM30FcTO

HBW8zkgVtNJ4GazSrBmk0uwZnM9AHZTPnB2Qz9gdjMv8HYzL/B2My/wfaKgAAyDYAALc6AAC
qOwAAoDsAAJo5AACUNwAAkDQEAlS1EgCGNh4AgzcoAH84MQF8ODkCejhBA3c3SAR1N08Ec
zdVBXE3WwVvN2IGbjdpBmw3cQdqN3oHaTeECGc4jwhlOJwJYzipCWE4uQlfOM4JXjilCV039Qlc
N/8JXDb/CVs2/wnVLwAAxDsAALM+AAcMpwAAAnEAAAJU+AAcPOwAAIjoAAIU6DgCAPBoAfD0
IAHk9LgF2PDYCCzw9BHA8RAVuO0sGaztRBmk7WAdoO18IZjtmCWQ7bgliO3cKYTuBCI88jAtdP
JkLWzynC1o8twxYPMsMVjvjDFY79AtVO/4LVTr/C1U6/wrRNQAAwD8AAK9CAACiQwAAmEMAA
JFCAACKPwAAhT4AAH8/CwB7QBcAd0EhAHNBKgfVqTICbEA6BGIAQQVnP0gGZD9OB2I/VQh
gP1sJXz9jClO/awtbP3QMWj9+DVg/ig1WQJcNVUCIDINAtQ5RP8kOUD/hDk8/8w1PPv0MTz7/DE8
+/wzNOQAAvEIAAKtFAACfRgAAIUcAAI1FAACGQwAAgEIAAHpDBwB1RBQAcUUfAG1FJwFqR
TACZkU3BGNEPgVhREQHXkNLCFxDUQlaQ1gKWENgDFZDaA1VQ3EOU0N8DIFDiA9QQ5UPT
kOjD01Dsw9LQ8cPSkPgD0pC8g5KQvwOSUH/DUIB/w3JPgAAuEUAACKhIAACbSQAakuoAAIIJA
ACCRgAAe0YAAHVHBABwSBEAbEkcAGhJJQBkSS0CYUk0A15IOwVbSEEGWEdICFZHTgpUR
1YLUkddDVBHZQ5PR28PTUd5D0tHhRBKR5MQSUeiEEdHsRBGR8UQRUbeEERG8Q9ERfwO
REX/DkRF/w7GQgAAtUgAAKVLAACTAAAJkwAAIVMAAB+SQAAdkoAAHBLAQBrTA4AZ00ZAG
NOIgbfTioBXE4xA1INOARWTT8GU0xFCFFMTApPS1MLTUtbDUtLYw5JS2wPR0t3EEZLgxBFS5
ERQ0ugEEJLsBBBS8QQQERdED9K8A8/SfsPP0j/Dj9I/w7DRgAAsUsAAKJOAACVTwAAi08AAIF
PAAB5TAAAcU4AAGtPAABIUQoAYVIWAF5SHwBaUygBV1MvAIRSNgRRUjwGTIFDCEXRSglKU
VELR1BYDUZQYQ5EUGoPQIB1EEFQgRBAUI8QPICeED1QrxA8UMIPO0/cDztO7w87TfsPO03/
DjtM/w7ASgAArU4AAJ5RAACSUgAAh1IAAH5TAAB0UAAAbFIAAGVUAABgVQcAW1YTAfHxH
QBVWCUAUIgtAU9YMwNMWDoFSldBB0dXRwIFVvK4KQ1ZWDEFWxw0/VWgOPVVzDzxVfw87V
o0POLadDzhVrQ43VcEONIXaDjZT7g42UvoONVH/DjVR/w68TwAAqIIAAJtUAACOVQAAg1YAAHI
WAABvVQAAZlCAAGBZAABaWgMAVVwPAFJdGQBXPXIATV4qAUpeMQJHXjgERV0+BUNdRQ
dAXUwJPIxUCjxcXAw6XGYNOVxxDTdcfQ02XlwNNVycDTRcrAwzW8AMMVvZCzFZ7QwwWPK
MMFf/DTBX/w23UgAApVUAAJdXAACKWAAAF1kAAHVaaABqWgAAYFwAAfLeAABUYAAAT2IL
AEtjFQBjZB8AR2QnAERkLgFCZDUCQGQ7BD5kQgU7ZEoHOWRRCDdjWgk1Y2QKNGNvCjJje
woxY4oKMGOaCi9jqwkuYr4JLGLYCCxg7QkrXvkKK13/Citd/wuyVwAAoVkaAJJbAACGXAAAE10
AAHFeAABIXwAAW2IAAFNkaABNZgAASGgGAERqEQBBahoAQGsJAD5sKgA8bDEBOmw4Ajhs
PwM2bEYENGxOBTJrVwYwa2EHLmtsBy1reQcsa4gHK2uYBiprqQYoarwGJ2rVBSdo6wYmZvgH
JWX/CCVw/wisWwAAm10AAI1fAACBYAAAdmEAAGxiAABhZAAAV2cAAE9qAABHbQAAQg8AA
DxxCwA5chYAN3MeADZzJgA0dC0AM3Q0ADF0OwEvdEMCLnRLAyx0VAMqdF4EKXRpBCd0dg
QmdIUDJXSVAYRzpwMic7oCIXLTAiBw6gMgb/cEH23/BR9t/wWmYAAAImlAAIhjAAB8ZAAAcM
UAGhnAABdagAAUm0AAEpwAABCcwAAO3YAADR5BgAwehAALnsZACx7IAARfCcAKnwuACh8
NgAnfT4AJn1GACV9TwEjfVoBlN1IASB9cgEffYEBHn2SARx8pAAbfLcAGXvPABI56AAyEpyBGHf
+Ahh2/wKfZQAakGcAAINoAAB3aQAAbWoAAGNtAABYcAAATnQAAEV3AAA9egAANX0AAC2A
AAAmgwgAI4QRACGEGQAghCEAH4UoAB6FLwAdhTYAHlY/ABuGSQAahIMAGIZfABeGbQAVh
nwAFIaNABKFnwAQhblADYTJAA2D4wAOgfMADoD8AA6A/wCYawAAimwAAH5tAABzbGAAaHA
AAF10AABTDwAASXsAAD9/AAA3ggAAL4UAACelAAAFigAAGlwlABWNEQAUjhGAE44gABKOJw
ARjy4AEI82AA+PQAANj0sADI9XAAqPZQAJj3QAB4+FAAWOlwADjqoAAY2+AAGM1gABi+kAAY
rzAACK9gCRcAAAhHEAAHlyAABudAAAYngAAFd7AABNgAAAQ4QAADqHAAAwiwAAKI4AACC
QAAAZkgAAEpQAAAUWCAAGlxAABZcXAASyHwADmCYAApGuAAGYNwAAmUIAAJIOAACZXA
AAmWsAAJh8AACyJgAAI6EAAJe0AACWYQAAlD4AAJTrAACU7wCLdgAAf3cAAHR5AABofAAAX
IAAAFgFAABGiQAAPi0AADORAAApIAAAIZYAABqZAAASmWAAAC50AAASfAwAAAnwkAAKARAA

ChFwAAoR4AAKIIAACjLQAAozcAAKRDAACkUAAApGAAAKRwAACjgwAAo5cAAKKqAAChvQA
AodEAAKDgAACg5QCGfQAAen4AAG2BAABhhQAAVYoAAEqPAAA/kwAANpcAACubAAAingAA
GqEAABKjAAALpAAABKYAAACoAAAAqQIAAKolAACqDgAAqxQAAKwbAAActlwAArisAAK82AA
CvQwAAsFIAALBjAACwdgAAsloAAK+dAACvrwAArsEAAK3QAAAct1gCBhAAAdlcAAGeLAABakA
AAT5UAAEOaAAA4ngAALqIAACSmAAAbqQAAE6sAAAqtAAADrwAAALAAAAACxAAAAswAAAL
QAAAC1AwAAAtgkAALcQAAC4FwAAuh8AALsoAAC9NAAAvkMAAL9UAAC/ZwAAwHsAAMCPAA
DAogAAwK8AAMC8AAC/wQB6jAAAbpEAAGGWAABUnAAASKEAADymAAAxqgAAJ64AAB2xA
AAUswAACrUAAAK3AAAAuAAAAALoAAAC7AAAAvQAAAL4AAAC/AAAAwQAAAMMDAADFCgA
AxxIAAMobAADQJQAA0jIAANNCAADUVQAA1WkAANV+AADVkQAA1qAAANarAADWrgB1lwAA
aJwAAFuiAABOqAAAQq0AADayAAAotQAAHrgAABO6AAAKvAAAAAL8AAADBAAAAwGAAAMQA
AADGAAAAxwAAAMkAAADMAAAAazgAAANAAAADTAAAA1wQAANsNAADfFwAA5CMAAOkwA
ADqQgAA7FYAAO1qAADtfQAA7Y0AAO6YAADunAD/AA0A/wAKAP8ACwD/ABYA/wAhAP8ALA
D/ADcA/QBDAPkATgD0AFcA8ABfAOwAZgDrAG0A6QBzAOcAeADmAH0A5QCCAOQAhwDjAlw
A4QCRAOAAIwDeAJ0A2wCIANkArQDWALgA1ADHANIA2wDQAO0AzgD7AMoA/wDHAP8Axd/
AMYA/wD/AAgA/wADAP8ACAD/ABMA/wAdAPsAKAD3ADMA9AA/AO8ASQDqAFMA5gBbAOMA
YwDhAGkA3wBvAN0AdADcAHkA2gB+ANkAgwDYAlgA1QCNANMAkwDRAJoAzwChAM0AqgDL
ALQAYQDBAMcA1ADEAOcAwAD2AL0A/wC9AP8AvAD/ALwA/wD/AAIA/wAAAP8ABQD+AA4A9
wAZAPIAJADuAC8A6AA6AOMARADfAE4A2wBWANgAXgDVAGUA0wBrANEAcADPAHUAzgB6
AMwAfwDKAIQAYQCJAMcAjwDFAJYAwwCdAMEApGc/AK8AvQC8ALoAzQC3AOEAtADxALMA/
QCyAP8AsgD/ALIA/wD/AAAA/wAAAP8AAQD1AAoA7gAVAOcAHwDgACoA2gA0ANYAPwDSAE
kAzgBSAMsAWQDIAGAAxgBmAMQAawDCAHEAwQB1AL8AegC+AH8AvQCFALsAiwC6AJIAuA
CaALYAogC0AKwAsQC3AK4AxwCsANsAqgDtAKkA+QCoAP8AqAD/AKcA/wD/AAAA/wAAAPsA
AADsAAUA4wAQANoAGgDRACQAzAAvAMkAOQDFAEMAwwgBMAL8AVAC8AFsAugBhALgAZw
C3AGwAtQBxALQAdgCzAHsAsgCBALAAhwCvAl4ArQCWAKsAngCoAKgApgCzAKQAwgCiANU
AoQDoAJ8A9QCdAP8AnQD/AJwA/wD/AAAA/wAAAO0AAADhAAAA1gAJAMwAFADFAB8AwAA
pAL0ANAC5AD0AtgBGALMATwCxAFYArwBcAK4AYgCsAGcAqwBsAKoAcQCpAHcApwB8AKYA
gwCkAlOaogCSAKAAmwCeAKUAnACwAJOAwwCYANEAlgDmAJUA8wCUAPwAkwD/AJMA/wD/
AAAA7QAAAOEAAADUAAAAygACAMEADQC7ABgAtwAjALMALQCwADcArQBBAKoASQCoAF
AApgBXAKQAXQCjAGIAoQBoAKAAbQCeAHIAAnQB4AJwAfwCaAYcAmQGPAJcCmQCVAqMAk
wOvAJEDvQCPBNAAJQXIAIsG8wCKBv0AiQb/AlgG/wDzAAAA5QMAANUJAADGCwAAvQgAALg
CBQCzABEArgAcAKoCJgCnBDEApAY7AKEHQwCfCEsAnQISAJsJWACZCI4AmApjAJcKaQCW
C28AIAAt1AJMLfACRDIQAKA2NAI4NIwCMDqIAig6uAlgOvgCGD9IAgxDoAIER9wCAEf8AfxH/AH4
R/wDsBAAA3BEAAMkXAAC8FwAAshYAAK0TAACrCwgApgoVAKINIACeECsAmxI1AJgTPgCWE
0YAIRNAJIUuWCRFFkAjxVfAI4VZQCMFWsAixZxAIkWeACIF4AAhheKAIQYIACCGKAAgBmsA
H4ZuwB8Gc8AehrmAHga9QB3Gv8Adhr/AHUZ/wDIEAAA0xwAAMAgAACzlgAAqiEAAKQeAACh
GgAAanhUOAJkYgWCVGiYakhswAJAcOQCNEEAix1IAIkdtWCHHVUAhh5bAIQeYQCDHmcAg
R9uAIAfdQB+IH0AfCCHAHshkQB5IZ0AdyGqAHUiuQBzIsWAcSLjAG8h9AFul8BbSH/AW0g/wHe
GgAAyiUAALkoAACsKQAAoykAAJ0nAACylwAAIh8JAJEgFgCNliEAiiMrAlckNACFJD0AgyVEAIEI
SwB/JVEAfSVXAHwmXQB6JmQAeSZqAHcmcgB1J3oBdCeEAXlojgFwKJoBbiinAWwotgFqKMoC
aCjhAmYo8wJIJ/4DZSf/A2Qm/wPYlgAAwysAALlvAACmMAAAnTAAAJcuAACSKwAAjicEAlonEg
CGKR0AgionAH8rMAB9KzgAeytAAHgrRwF3K00BdStTAXMrWgFxlGABcCxnAm4sbgJtLHcCay2
BAmktjANnLpgDZS6IA2MutANhLscEYC3fBF4t8QRdLf0EXSz/BF0s/wTSKQAAvjAAAK00AAChNg

AAmDUAJJEzAACMMQAAiC4AAIMuDQB/LxkAezAjAHgxLAB1MTQAczE8AXExQwFvMUkCbTF
QAmsxVgNpMV0DaDFkA2YxawRkMnQEYzJ+BGEyiQVfM5UFXT0jBVwzsgZaM8UGWDLdBicy8
AZWMfwGVjH/BIYw/wbNLwAAuTUAACKk4AACdOgAAIDkAAI03AACGNgAAgjQAAH00CQB5NBU
AdTUgAHE2KQBuNjEBbDY4AWo2PwJnNkYDZTVMA2M1UwRhNVkEYDZgBV42aAZdNnEGWz
Z7Blk3hgdYN5MHVjehB1Q3sAhTN8MIUTbbCFA27ghQNfsITzX/B081/wfJNAAAtTgAAKY7AACZ
PQAAkD0AAIg7AACCOgAAfDgAAHc4BgBzORIAbzocAGs6JQBoOy0BZTs1AmM6PANgOkIDXjp
JBFw6TwVaOIYGWTpdBlc6ZQdWOm4lVDp4CFI7hAIRO5EJTzufCU47rglMO8EJSzrZCUo67QIK
OfOJSTn/CUK5/wnFOAAAsTAAKI/AACWQAAAJEAAAIQ/AAB+PQAAdwAAHI9AgBtPQ4AaT4Z
AGU/lgBiPyoBXz8yAV0+OANaPj8EWD5FBVY+TAZUPIMHUj5aCFE+YwhPPmwJTj52Ckw+ggpL
P48KST+dC0g/rAtGP78LRT7XC0Q+7ApEPfkKRDz/CkQ8/wrBOWAArj8AAJ9CAACTQwAAiUMAA
IBCAAB5QAAAcz8AAG1BAABoQQsAZEIWAGBDIABdQygAWkMvAVdDNQJUQjwDUkJCBVBCS
QZOQIAHTEJYCEtCYAIJQmkKSEJzC0ZCfwFQ40LREObC0JDqwtBQ70LQELVCz9B6ws/QfgKP
0D/Cj9A/wq9PgAAq0IAAJxEAACQRQAaHUYAAH1FAAB1QwAAbkMAAGhEAABiRQgAXkYUAFt
HHQBRYUAVUcsAVJHMwJPRzkDTUdABUtHRwZJR04HR0ZVCUVGXgpERmCLQkZxC0FHfQt
AR4sMPkeaCz1Hqgs8R7wLO0bUCzpF6gs6RPgLOkT/CzpD/wq6QQAAP0UAAJIHAACMSAAAgk
kAAHIIAABxRQAAaUcAAGNJAABdSgUAWUsQAFVLGgBSTCIAUEwqAU1MMAJKTdCdSEw9B
EZMRAZETEsHQktTCEBLWwk/S2UKPUtvCzxLews7TIkLOkyYCzhMqAs3S7sKNkvSCjVK6Qo1S
fcKNUj/CjVI/wq2RAAApEgAAJVKAACJSwAAfkwAAHVMAABsSQAAZEesAAF1NAABYTglAU08N
AFBQFwBNUR8ASIIAEhSLgFGUjUCQ1I7A0FSQgU/UUkGPVFRBzxRWQg6UWMJOFFtCjdRe
Qo2UYcKNVGXCjRRpwkyUbkJMVDRCTFP6AkWtvcJME3+CjBM/wqxSAAAOEsAAJJNAACFTgA
Aek8AAHFPAABmTgAAXIAAAfDsaABSUwAATVUJAEIWEwBHvXwARVckAENYKwBAWDIBPlg
5AjxYPwQ6WEcFOVhOBjdXVwc1V2AIM1drCDJXdwxgV4YIMFeVCC9XpgguV7gHLfBPByxV5wc
rU/YIK1L+CCtS/wmtSwAAAnE4AAI5QAACBUQAAAdIIAAG1TAABiUwAAWFUAAFFXAABMWQAA
R1sFAENcEABAXRkAPI4hADxeKAA7Xi8AOV82ATdfPQI1X0QDM15MBDFeVAUwXI4GLI5pBi1e
dQYsXoQGKI6TBSlepAUoXrYFJ13OBCZb5gUmWvUGJVn9BiVY/weoTwAAI1IAAIIUAAB9VQAAc
IYAAGhXAABdWAAAVFoAAE1dAABGXwAAQGEAADxjCwA5ZBUAN2UdADVIJAA0ZisAMmYyA
DBmOQEvZkABLWZIAixmUQMqZIsDKWZmAydmcwMmZoEDJWaRAyRmogMjZbQCIWTMAiFj5
QlgYfQDIGD9BCBf/wWiVAAAKIYAAIRXAAB4WQAAbloAAGRbAABZXQAAUGAAAEhiAABBZQA
AO2gAADRqBgAwBBAALmwYAC1tHwArbSYAKm4tACluNAAnbjwAJm5EACVutQEjblcBlm9jASF
ubwEgbn4BHm6OAR1uoAEcbbIAGmzJABlr4wAZafMBGWj8Ahln/wKcWQAAjVoAAH9cAABzXQA
AaV4AAGBgAABVYgAAS2YAAERpAAA9awAANW4AAC5xAAAocwkAJXUSACN1GgAidSEAIXY
oACB2LwAedjYAHXY+ABx3SAAbd1IAGndeABI3awAXd3oAFneKABR2nAASdq4AEXXFABB04A
AQcvEAEXH7ABFw/wCVXgAAh18AAHpgAABvYgAAZWMAAFtIABBRaAAARmwAAD9wAAA3cw
AAL3YAACh4AAAhewlAG30LABI+EwAXfhoAFn8hABV/KAAUfy8AE383ABKAQQARgEsAD4BXA
A2AZQAMgHQACoCEAAI/lgAHfqqABX28AAR81AAEe+gABHrzAAV6+QCPYwAAgWQAAHVIAA
BrZwAAYGkAAfZsAABMcAAAQnQAADI3AAAXegAAKX0AACKAAAAbgwAAFIUCAAA2HCgAKhx
MACYgaAAiIIQAHICgABogwAAWIOQAEiUMAAoIPAAAGJXQAAiWwAAII8AAClJgAAh6EAAIe0AA
CGyQAAheAAAITsAACD8wClAQAAe2oAAHBrAABmbQAAWnAAAFB0AABGeAAAPXwAAD0AA
AAqgwAAI4YAABulAAAUiwAADY0AAAaPBgABjw0AAJAUAACQGgAAkSEAAJEoAACSMQAAkj
oAAJJGAACTUwAAk2IAAJNzAACShQAAkpgAAJGrAACQwAAAj9QAAI/kaACO7ACBbwAAadm8
AAGxxAABfdAAAVHgAAEp9AAA/ggAANoUAACyJAAAJjQAAHI8AABSSAAANkwAABpUAAACXA
QAAMAcAAJkNAACZFAAAmhoAAJsgAACcKAAAnDEAAJ08AACdSQAAnVgAAJ1pAACdewAAn

Y8AAJ2iAACctQAAM8gAAJvZAACa4wB9dQAACnYAAGV5AABZfgAAToIAAEOHAAA4jAAAL5AA
ACWTAAAcIlgAAAFJkAAAYbAAAFnQAAAJ8AAACHAAAAogAAAKMFAACkCgAaPRAAAKYXAAcN
HgAAqCYAAKkvAACqOwAAqkoAAKpbAACqbgAAqolAAKmXAACpqqgAAqbsAAKjJAACo1AB4fA
AAa38AAF+DAABTiAAAR40AADyTAAAXlwAAJ5sAAB2eAAAVoQAADKQAAASmAAAApWAAAK
kAAACrAAAArAAAAK0AAACuAAAArwYAALAMAACyEwAAsxsAALUjAAC3LgAAuDwAALhMAAC
5XwAAuXMAALqHAAC6mwAAuawAALi5AAC4wwByhAAAZyKAaFiOAABMIAAAQJkAADWeAAA
qowAAIKcAABeqAAANrQAABa8AAACxAAAAsgAAALQAAAC1AAAAtwAAALgAAAC5AAAAuwAA
ALwAAAC+BgAAwA0AAMMWAADGIAAAyiWAAMs8AADMTwAAzWMAAM53AADPiwAAz5wAAM
+nAADPrwBsJwAAX5QAAFKaAABGoAAAOqYAAC6rAAAKrwAAGrMAABC2AAAFuAAAAALoAAA
C8AAAAvAAAAAL4AAADAAAAAwQAAAMMAAADFAAAAxwAAAMoAAADMAAAA0AAAAANMIAAD
ZEGAA3h0AAOMqAADkPAAA5VEAAOZIAADneQAA6loAAOiVAADonQD/AAgA/wADAP8ACAD/A
BMA/wAeAP8AKQD9ADQA+AA+APQASQDwAFIA7QBBAOsAYgDpAGkA5wBuAOUAdADkAHgA
4gB9AOAAggDfAlcA3QCMANsAkgDZAJgA1wCgANUAqQDTALMA0QDAAM4A0wDJAOkAxgD4
AMUA/wDEAP8AxAD/AMMA/wD/AAIA/wAAAP8ABQD/ABAA/QAbAPgAJQDyAC8A7gA6AOsARA
DnAE4A4wBXAOEAXgDeAGQA2wBqANkAbwDWAHQ1QB5ANMAfgDRAIMA0ACIAM4AjgDMA
JUAYwCcAMkApQDHAK4AxAC7AMAAzAC8AOIAuwDzALoA/wC5AP8AuQD/ALgA/wD/AAAA/wA
AAP8AAgD7AAsA9AAWAOWAIQDmACsA4gA1AN8AQADbAEkA1gBSANIAWQDPAGAAzQBmA
MsAawDJAHAAyAB0AMYAeQDFAH4AxACEAMIAigDAAJAavwCYAL0AoQC6AKoAtgC1ALMAxg
CyANsAsADuAK8A/ACuAP8ArgD/AK4A/wD/AAAA/wAAAPoAAADxAAcA5wASAN4AHADYACYA
0wAwAM8AOgDMAEQAyABNAMUAVADCAFsAwABhAL4AZgC9AGsAvABwALoAdQC5AHoAuA
B/ALYAhQC0AlwAsgCUALAAnACTAKYAqwCxAkKAwACoANMApgDoAKUA9wCkAP8AowD/AKM
A/wD/AAAA/wAAAPAAAADIAAMA2AAMANAfWdJACEAxAArAMEANQC+AD8AuwBIALkATwC
2AFYAtABcALMAYQCxAGYArwBrAK4AcACsAHUAqwB6AKkAgQCoAlgApgCPAKUAmACjAKIAo
QCtAJ8AuwCdAM0AmwDjAJ0A8gCZAPwAmQD/AJgA/wD/AAAA8gAAAOUAAADXAAAAyWHA
MMAEgC8ABwAuAAmALUAMACyADkAsABCak4ASgCrAFEAqQBxAKcAXACIAGEApABmAKM
AawCiAHAAoQB2AJ8AfACeAIMAnACMAJsAlQCZAJ8AlwCqAJUAtwCTAMkAkQDeAJAA7gCOA
PoAjQD/AlwA/wD1AAAA6AAAAANYAAADIAAAAvwABALgACwCyABYArgAgAKsAKQCoADMApQ
A8AKMARACHAEsAnwBSAJ0AVwCcAF0AmwBiAJkAZwCYAGwAlwByAJYAeACUAIAAkGCI AJE
AkgCPAJwAjQCoAlsAtQCJAMUAhwDbAIUA7QCDApCaggD9AIIA/wDtAAAA3QAAAMkFAAC8B
QAAswMAAK8ABQCqABAAPgAZAKIALwCfAC0AnAA2AJ0APgCYAEYAlgBNAJQBUwCTAVgAkQ
JdAJACYwCPAmgAjQNuAlwDdQCKBH0AiQSFAlcFjwCFBpoAgwamAlEHswB/B8UAFajbAHsI7A
B5CfGAEaj/AHgl/wDIAwAAzw4AAL8RAACyEgAAqhEAAKQNAACiBwcAngMSAJ0FHQCXBycAIA
kwAJEKOQCPC0EAjQtIAlwMTgCKDFQAIQxZAlcNXwCGDWUAhA1rAIMOcgCBDnoAfxCDAH0Q
jQB7EZkAeRGIAHcSswB1EcYAcxLdAHES8ABwEvwAbxL/AG8S/wDeEAAAxhcAALYbAACqHAA
AoRwAAJsZAACYFQAAlw4KAJIQFwCPEiEAixMrAlkUMwCGFTsAhRVDAIMWSQCBfk8AfxZVA
H4XWwB8F2EAexdnAHkYbwB3GHcAdhiAAHQZigByGpYAcBqjAG4asQBsGsMAahraAGga7gBn
GvsAZhr/AGYa/wDVGQAAvx8AAK8iAACjJAAAmiMAAJQhAACQHgAAjRoFAIoYEgCGGh0Agxs
mAlAcLwB9HTcAfB0+AHoeRQB4HksAdh5RAHUeVwBzHI0Ach9kAHafawBvIHQAAbSB9AGshiAB
pIZQAZyKhAGUirwBjIsEAYiHYAGAh7QFflfoBXiH/AV4g/wHNIAAAuCUAAKkpAACdKgAAISkAAI4
oAACJJgAAhilAAlIgDQB+IRgAeyliAHgjKwB1IzMAcyQ6AHIkQQBwJEgAbiVOAGwIVABrJV0AaS
VhAGgmaABmJnEBZCd6AWMnhQFhKJEBXyifAV0orQFbKL8BWijVAIgn6wJXJ/kCVyb/AlYm/wL
HJgAAsysAAKQuAACYLwAAjy8AAIktAACDKwAAfykAAHsnCAB3JxQAdCgeAHApJwBuKS8AbC

o2AGoqPQBoKkQAZipKAWQqUAFjK1cBYSteAV8rZQFeLG4CXCx4AlssgwJZLY8CVy2dA1Ytqw
NULb0DUi3TA1Es6gNQLPgEUCv/BE8r/wTCKgAAry8AAKAYAACUMwAAizMAAIQxAAB+MAAAe
S4AAHUtBABxLREAbS4bAGoulwBnLysAZS8zAGlvOgFhMEABXy9GAV0vTQJbL1MCWTBbAlgw
YgNWMGsDVTF1BFMxgARSMo0EUDKbBE8yqgRNMrsFSzHRBUox6AVKMPcFSTD/BUkv/wW+
LgAAqzMAAJw2AACQNwAAhzCAAH81AAB5NAAAdDIAAG8yAQBmMg0AZzMXAGMzIABgNCgA
XjQvAFw0NgFaND0BWDRAIY0SQJUNFADUzRYBFE1YARQNWgFTjVyBU02fgVLNOSGSjaZBk
g2qAZHNrkGRTbPBkQ15wZENPYGRDT/BkQz/wa6MQAAqDYAAJk5AACNOgAAgzoAAHs5AAB
1NwAAbzYAAGo2AABINwkAYTcUAF04HQBbOCUAWDksAFY5MwFTODkBUThAAk84RgNOOE
0ETDIVBEs5XQVJOWYGSdlwBkY6ewZFOokHRDqXB0l6pgdBOrCHQDrOBz855gc+OPUHPjj+B
z43/we2NQAAPdKAAJY7AACKPAAAgD0AAHc8AABxOgAAajKAAGQ6AABfOwYAWzsRAfG8Gg
BVPSIAUj0pAFA9MAFNPTYBSzw8Akk9QwNIPUoERz1SBUU9WgZEPWQGQj1uB0E+eQdAPoc
HPj6VBz0+pQc8PrYHOj7MBzo95Qc5PPQHOTv+Bzk7/weyOAAAOtWAAJM+AACHPwAAfEAAA
HQ/AABtPQAAZj0AAF8+AABaPwMAVKAOFNAFwBQQR8ATUEnAEtBLQFIQTQBRkE6AkVBQ
QNDQUgEQUFQBUBCWAY+QmEHPUJsBzxCdwc6QoUHOUBzhDpAc3QrUHNkLLBzVB5Ac
0QPQHND/9BzQ//wevOwAAanj4AAJBAAACDQgAAeUIAAHBCAABOQQAAYEEAAFPCAABVQwA
AUUQLAE1FFABKRh0ASEYkAEZGKwBDRzEBQkC4AkBHPwM+R0YEPEdNBTtHVgY5R18GOE
dqBzdHdQc1R4MHNEeSBzNHogcyR7MGmUBJBjBG4gYvRPMHL0P9By9D/werPgAAmkEAAIx
AACARQAAdUUAAGxFAABkRAAAW0UAAFVGAABQSAAAS0kHAEdKEQBFSxoAQkshAEBMK
AA/TC8BPUw1ATtMPAI5TEMDN0xLBDZMUwU0TF0FM0xnBjFMcwYwTIEGL02QBi5MoQUtTLIF
LEzIBStL4QUqSflGKkj8BipH/wanQQAAl0QAAlIGAAB8SAAAckgAAGhIAABfSQAaVkkAAE9LAA
BKTQAARU4DAEFPDQA/UBYAPFEeADtSJQA5UiwAN1IzATZSQQE0UkECMIJIAzFSUQMvUlsE
LVJIBCxScQQRUn8EKIOPBCISnwQoUrEEJ1HHAyZQ4AMIT/IEJU77BSVN/wWjRQAakkcAAIRJA
AB4SwAAbksAAGRMAABbTQAAUE8AAEpQAABEUgAAP1QAADtVCQA4VxMANIcbADRYlgAy
WCkAMVkvAC9ZNgAuWT0BLFIFASTZTglpWVgCKFljAydZbwMmWX0DJVmNAiNZngliWa8CIVjF
AiBX3wEgVfEDIFT7AyBT/wSdSQAajksAAIBNAAB0TgAAaU8AAGBQAABXUQAATVMAAEZVAA
BAWAAAOLoAADRcBQAwxQ4ALi4XACxfHgArXyUAKV8sAChgMgAnYDoAJmBCACRgSwEjYFU
BlmFgASFhbAEfYXoBHmCKAR1gmwAcYK0AGl/CABpe3QAaXPABGVv6ARla/wKYTQAaiU8AA
HtRAABvUgAAZVMAAFxUAABTVgAASVkaAEJbAAA8XQAANWAAAC5jAAAoZQkAJWYSACRm
GQAiZyAAIwcnACBnLgAfaDUAHmg9AB1oRgAbaFAAGmhbABloaAAYaHYAF2iHABVomAAUZ6
oAEma/ABFI2gASZO4AEML5ABJi/wCSUQAAG1MAAHZVAABrVgAAYVcAAfHYAABPWwAARV4
AAD5hAAA3ZAAAMGcAAClpAAAibAMAHW4LABpvFAAZbxsAGG8hABdwKAAWcC8AFXA3ABN
wQAAScUoAEXFWABBXyWAOcXEADXGCAAtwkwAJb6YACG65AAdt0AAHbOYAB2vzAAhq+wC
MVgAAfVgAAHFZAABmWgAAXVwAAFNeAABKYQAAQWQAADloAAAxawAAKm4AACNxAAAacc
wAAFnYEABB3DAAOeBQADXgbAAx4lgALeSkACnkxAAh5OgAHeUQABnIPAAR5XAADeWsAAX
I7AAB5jQAAeJ8AAHeyAAB2xwAAAdd8AAHTtAABz9QCFWwAAAd10AAGxeAABiXwAAWGEAAE5k
AABFaAAAPGwAADNwAAArcwAAJHYAAB14AAAWewAAEH0BAAh/CAADgA4AAIWAACBHA
AAgSMAAllqAACCMgAAgJwAAIJHAACCVAAAgmMAAIJzAACChQAAGpgAAIGrAACAwAAaf9YA
AH7mAAB98AB+YQAacmIAAGhjAABeZQAaU2gAAEIsAAA/cQAANnQAAC14AAAlfAAAHX4AAB
aBAAAQgwAACIUAAAKHBAAAiAoAAIkRAACJFgAAih0AAIsjAACLKwAAjDQAAlw+AACMSwAAj
FoAAIxqAACMfAAAJAAAlujAACltwAAissAAIneAACl6gB4ZwAAbmgaAGNpAABYbQAATXEAA
EN2AAA5egAAL34AACaCAAAehQAAFogAAA+KAAAljAAAAY4AAACQAAAAkQUAAJIKAACTEA
AAIBUAAJUcAACVlgAAlioAAJc0AACXQQAAMe8AAJhgAACYcgAAmIYAAJeaAACXrQAAlSAAA

JXSAACV4QB0bQAAaW4AAF1yAABRdgAARnsAADyAAAAyhAAAKIkAAB+MAAAWjwAADpIAA
AeUAAAAIgAAAJgAAACZAAAAmgAAAJsCAACcBwAAng0AAJ8TAACgGgAAoSEAAKlqAACkNQ
AApEMAAKRUAACkZgAApHoAAKSPAACkowAAo7MAAKPDAACj0QBvdAAAY3cAAFd7AABLgQ
AAQIYAADSLAAAqkAAAIJQAABeXAAAOmgAABp0AAACeAAAAoAAAAKIAAACkAAAApQAAAK
YAAACoAAAAqQIAAKoIAACsDgAArRYAAK8eAACxKAAAsjUAALNFAACzWAAAs2wAALKBAAC
ylgAAsqgAALK0AACyWABpfAAAXIEAAFCGAABEjAAAOZIAAC2XAAAJmwAAGZ8AABCjAAAGpg
AAAKgAAACqAAAAqwAAAK0AAACvAAAAsAAAALEAAACzAAAAtAAAAALYAAAC4AgAAugkAAL
wSAAC/HAAAWiYAAMQ1AADFRwAAxIsAAMdwAADlhAAAx5cAAMeIAADGsABjhwAAVowAAEq
SAAA+mQAAMp4AACakAAAcqAAAE6wAAAmvAAAAsgAAALQAAAC2AAAAtwAAALkAAAC6AA
AAvAAAAAL0AAAC/AAAAwQAAAMMAAADGAAAAyQAAAMwEAADQDQAA1RkAANsmAADdNgA
A3koAAOBfAADhcwAA4oUAAOKSAADjnAD/AEEA/wAAAP8ABgD/ABEA/wAbAPsAJQD4AC8A9
gA6APMARADuAE4A6gBWAOCAXQDIAGQA4wBpAOEAbgDgAHMA3gB4AN0AfQDbAIEA2gCH
ANgAjQDWAJQA1ACbANEApADPAK4AywC6AMYAzADEAOMAwwgD2AMEA/wDBAP8AwAD/AM
AA/wD/AAAA/wAAAP8AAgD/AA0A+AAAXAPIAQDuACsA6wA1AOcAQADiAEkA3wBRANsAWQD
ZAF8A1gBIANQAagDSAG8A0QB0AM8AeADOAH0AzACDAMsAiQDIAI8AxQCXAMMAoADAACK
AvAC1ALsAxQC5ANwAtwDxALYA/wC2AP8AtQD/ALUA/wD/AAAA/wAAAP8AAAD3AAkA7AATA
OYAHQDhACcA3AAxANkAOwDUAEEQA0ABNAM0AVADLAFsAyQBhAMcAZgDEAGsAwgBvAMA
AdAC+AHkAvAB+ALoAhAC4AIsAtwCSALUAmwCzAKUAsQCwALAAvwCuANQArADrAKsA+gCr
AP8AqgD/AKoA/wD/AAAA/wAAAPUAAADpAAQA3wAOANcAGQDQACMAzAAsAMkANgDGAD8
AwgBIAL4ATwC7AFYAUBcALcAYQC1AGYAtABqALMAbwCxAHQAsAB5AK8AfwCuAIYArACO
AKoAlwCpAKEApwCsAKUAuQCjAM0AogDkAKAA9QCfAP8AnwD/AJ4A/wD/AAAA9gAAAOoAAA
DbAAAA0AAKAMcAFADBAB4AvQAnALoAMQC2ADoAswBCALAAsgCuAFEARQBxAKwAXACqA
GEAqQBmAKgAagCnAG8ApgB1AKQAewCjAlIAoQCJAKAAkgCeAJ0AnACoAJ0AtQCYAMYAlwD
dAJUA8ACUAPwAkWd/AJMA/wD5AAAA6wAAANsAAADMAAAAawQAFALoADgC1ABkAsAAiAKw
ALACqADQAqAA9AKYARQCkAEsAowBSAKEAVwCgAFwAnwBhAJ0AZgCcAGsAmwBwAJ0Adg
CYAH0AlwCFAJUajgCTAJkAkQCkAI8AsQCNAMEAiWDWAlkA6gCIAPcAhwD+AlcA/wDvAAAA3
QAAAMsAAAC/AAAAtgAAAK8ACgCpABMApgAdAKMAJgCgAC8AngA3AJwAPwCaAEYAmQBM
AJcAUgCWAFCAlABcAJMAYQCSAGYAkQBsAI8AcgCOAHkAjACBAIoAiwCIAJUahwChAlQArgC
CAL0AgADRAH8A5gB9APQAFAD9AHsA/wDmAAAAzwAAAL8AAACyAAAAqwAAAKYAAwChAA
0AnQAXAJ0AIACXACKAIQAxAJMAOQCRAEEAjwBHAI4ATQCMAFMAiwBYAlkAXQCIAGIAhwB
oAlUAbgCDAHYAaggB+AlAAiAB+AJMAfACfAHoArAB4ALsAdgDOAHQA5QBzAPIAcgD6AHIA/wD
bAQAAxQgAALULAACpDQAAoQwAAJsJAACZAwcAlQARAJIAGgCPACMAjAAsAlcCNACIAzsAh
gNCAIQESACDBE4AgQVTAlAFWQB+BI4AfQZIAHsGawB6B3MAeAd8AHYIhgB0CJEAcmgAHA
JqwBuCbsAbAnPAGsK5QBpCvMAaAr8AGgK/wDQDAAAuxIAAK0VAACHFwAAmRYAAJMUAAC
PEAAAJgkJAIsHFACHCR4AhAsmAIEMLwB/DDYAfQ09AHwORAB6DkoAeQ5PAHcOVQB2D1sA
dBBhAHlQaQBxEHAAbxF6AG0ShABrEpAAaROdAGcTqgBIE7sAYxPQAGET5wBgE/YAYBP/AF8
T/wDIFAAAtBoAAKUeAACaHgAAkh4AAIsCAACHGgAAhBUBAIMRDQB/ExgAexQhAHkVKgB2FT
EAdBY4AHMWPwBxFkUAbxdLAG4XUQBsf1cAaxheAGkYZQBnGG0AZhI3AGQagQBiGo0AYBu
aAF4bqQBdG7kAWxvOAFkb5QBYG/UAWBr/AFca/wDBGgAAriAAAKAJAACVJAAAJCQAAIUJAAAC
AlQAAfB4AAH0aCAB3GhQAcxsDAHAcJgBuHC0AbB00AGodOwBoHUIAZx5IAGUeTgBkHIQAYh
5bAGAfYgBfH2sAXSB0AFwhfwBaLYsAWCKYAFcipwBVlrcAUyHMAFIh5AFRIfQBUCD+AVAg/wG
7IAAAqSUAAJsoAACQKQAAhikAAH8oAAB6JgAAdiQAAHMHbABwIBAAbCEZAGkilgBmlkAZCM

xAGljNwBhlz4AXyREAF0kSgBcJFEAWiRYAFklXwBXJWgAViZxAVQmfAFTJ4kBUSeWAU8opQF
OKLUBTCfKAU5n4gFKJvMCSSb9AkkI/wK2JAAApSkAAJcsAACLLQAAgi0AAHosAAB1KwAAcCk
AAGwmAABpJgsAZScWAGlnHgBfKCYAXSgtAFspNABZKToAWCIBAFYpRwBUKU0BUypVAVEq
XAFQKmbUTitvAU0segJMLIYCSiyUAKgtowJHLbMCRSziAkQr4QJEK/IDQyr9A0Mq/wOyKAAAS
0AAJMvAACIMAAAfjEAAHYwAABwLgAAay0AAGcrAABiKwgAXywSAFssGwBZLSMAVi0qAFQtM
ABTLjcAUS49AE8uRAFOLkoBTC5SAUsvWgJL2ICSDBsAkeweANFMYQDRDGSAlxogNBMBI
DPzHHAz4w3wM+L/EDPi78Az4u/wSvKwAAanjAAAJAyAACENAAejQAAHI0AABsMgAAZjAAAG
EwAABdMAQAWTAPAFYxGABTMiAAUDInAE4yLQBMMjQASjl6AUkyQAFHM0cBRjNPAkUzVwJ
DNGADQjRqA0E1dQQ/NYIEPjWRBD01oAQ7NbEEOjXFBdk03gQ5M/AEODP7BDgy/wSrLwAAm
zMAAI01AACBNwAAdzCAAG43AABnNgAAYjQAAFw0AABXNQEUAUzUMAF2FQBNNh0ASzYkA
Eg3KgBGNzAARDc3AUM3PQFCN0UCQDhMAj84VQM+OF4DPDloBDs5cwQ6OYAEOTqPBDc6
nwQ2Oq8ENTnEBDQ43QQzN/AEMzf7BDM2/wSoMgAAIzYAAIo4AAB9OQAAczoAAGs6AABjOQ
AAXTcAAFc4AABSOQAATjolaEo6EgBIOxoARTshAEM7KABBPC4APzw0AT48OwE9PEICOzxK
Ajo9UgM4PVsDNz1mBDY9cQQ1Pn8EMz6NBDI+nQQxPq4EMD3CBC893AMuPO8ELjv6BC46/w
SkNQAADgAAIY6AAB6PAAAcDwAAGc9AABfPAAAWDoAAFI8AABNPQAAST4FAEU/DgBCPxc
AQEAeAD5AJQA8QSwAO0EyADIBOQE4QUABNkFHAjVCUAMzQlkDMkJKAzFCbwQvQn0ELkO
MBC1DnAMsQq0DK0LBAypB2gMpQO4EKT/6BCk+/wShOAAAKTsAAIM9AAB3PgAAbD8AAGNA
AABbPwAAUz4AAE1AAABIQgAAQ0MCAEBECwA9RBQAOKUcADhGlgA3RikANUYvADRHNgAy
Rz0BMUdFAS9HTQluR1cCLUdiAytHbQMqSHsDKUiKAyhImglR6wCJkfAAiVG2QIIRe0DJET5Ay
RD/wSdOwAAjT4AAH9AAABzQQAaUIAAGBDAABYQwAATkMAAEdFAABCRgAAPkgAADpJC
AA3ShEANEsZADJLIAAxTCYAL0wtAC5MMwAsTTtoAK01CASpNSwEoTVQBJ01fAiZNawliTXkCJ
E2IAiNNmQEiTaoBIUy+AR9M2AEfSuwCH0n4Ah9I/wOZPgAAiUEAAHtDAABvRAAAZUUAFAxG
AABURgAASkgAAENKAAA+SwAAOE0AADRPBAAwUA0ALVEVACtSHAAqUiMAKFIPACdTMAA
mUzcAJVM/ACRTSAAiVFEAIVRcASBUaAEfVHYBHISGAB1UIwAcU6gAGIK8ABIS1QAZUOsAG
U/4ARIO/wGUQgAAhEUAHdGAABrRwAAYUkAAfHJAABQSgAARkwAAEBOAAA6UAAANFIAA
C5VAAApVwgAJVgRACRZGAAiWR8AIVkIACBaLAAfWjMAHIlo7ABxaRAAbWk0AGItZABlbZQAY
W3MAF1uDABValAAUWqYAE1m6ABFY0gASV+oAEIb3ABNV/gCPRgAAf0kAAHJKAABmSwAA
XUwAAFRNAABMTgAARFEAADxTAAA2VgAAL1gAACIbAAAIxGMAHI8LABtgEwAaYRoAGWEh
ABhhJwAWYS4AFWI2ABRiPwATYkkAEmJUABFiYQAPYm8ADWJ/AAxikaALYalACWC1AAhfzA
AIXuMACV3yAAIc+gCJSwAAek0AAG1OAABiTAAWVAAAFRAABIUwAAQFYAADhZAAAxxA
AAKI8AACNiAAAdZAAAF2cFABJpDQAQaRQAD2kbAA1qlgAMaikAC2owAApqOQAJakMACGpO
AAZqWgAFamgAA2p4AAJpigABaZwAAGivAABnxAAAZt0AAGXsAABk9ACCTwAAAdFEAAGhSAA
BeUwAAVVQAAE1WAABEWQAAO1wAADNgAAArYwAAJWYAAB5pAAAYbAAAEW4BAAPwCA
AGcRAAA3EWAAJyHQABciQAAHlrAABYmWAAAcjwAAHJHAABYVAAAcmlAAHJxABYgwAAcpY
AAHGpAABwvQAAb9QAAG7mAABt8AB7VQAAbiYAAGNXAABaWAAAUVoAAEddAAA/YAAAN
mQAAC1oAAAmawAAH24AABhxAAAScwAAC3UAAAR4BgAAeAwAAHkSAAB6GAAAEh4AAHsl
AAB7LAAAfDUAAHxAAAB8TAAAfFoAAHxqAAB8fAAAE48AAHujAAB6tgAAecwAAHjgAAB46wB1
WgAAaVsAAf9cAABWXQAATGEAAEJIAAA5aQAAMG0AACdxAAAFdAAAGHcAABF6AAAKfAAA
BH4AAAB/AgAAgAgAAIENAACCEwAAgXgAAIqfAACFJQAAhi0AAIY3AACGRAAAhiEAAIZhAAC
GcwAAhocAAIacAACFrwAAhMQAAIPYAACD5QBvXwAAZWAAAFtiAABQZQAARmkAADxuAAA
ycgAAKXcAACB6AAAYfgAAEYEAAAmDAAACHQAAAlcAAACIAAAAigIAAIsHAACMDAAAJREAA
I4XAACPHgAAKSUAJluAACSOQAakkcAAJJXAACSaQAaknwAAJKSAACRpgAAkbgAAJDLaA

CP2gBrZQAAYWcAAFVqAABKbgAAP3QAADV4AAArfQAAIoEAABmFAAAQIAAACIsAAAGNAAA
AjwAAAJEAAACSAAAAkwAAAJUAAACWBAAAlwkAAJkOAAcCaFQAAnBwAAJ0kAACfLgAAAnzsA
AJ9LAACfXQAAn3EAAJ+HAACfnAAAnq4AAJ6+AACeywBnbAAAWm8AAE90AABDeQAAOH8A
AC6EAAAJiQAAGo0AABGQAAAlkwAAAJYAAACXAAAAmQAAAJsAAACdAAAANGAAAJ8AAACH
AAAAogAAAKQFAACICwAApXlAAKkaAACrlwAArC8AAKw/AACsUQAARWUAAK16AACTjwAAraIA
AK2wAACtuwBgDQAAVHkAAEH/AAA9hQAAMYsAACaQAAAcIAAAEpKAAAmcAAAAnwAAAKEA
AACiAAAApQAAAKcAAACoAAAAqgAAAKsAAACtAAAArwAAALAAAACyAAAAAtUAALYNAAC5
FgAAvCEAAL4vAAC/QQAav1UAAL9qAAC/gAAAv5MAAL+iAAC/rQBafwAAToUAAEKLAAA2kQA
AKpgAACCCcAAAVoQAAC6UAAAGoAAAAqWAAAK0AAACuAAAAsAAAAALIAAAC0AAAAAtQAAALc
AAAC5AAAAUwAAAL0AAAC/AAAAwgAAAMUAAADJCQAAzhQAANMgAADVMAAA10QAANIZA
ADabQAA24AAANuQAADamwD/AAAA/wAAP8AAwD/AA0A/gAXAPsAIQD2ACsA8wA1APAAPw
DsAEkA6ABRAOYAWADiAF4A3gBkANsAaQDZAG4A1gByANQAdwDSAHwA0ACBAM4AhwDM
Al4AywCWAMkAnwDHAKkAxQC1AMIAxgDAAN4AvwDzAL4A/wC9AP8AvAD/ALwA/wD/AAAA/w
AAP8AAAD6AAkA9QATAO8AHgDqACcA5wAxAOQAOWdfAEQA2ABMANIAVADOAFoAzABgA
MoAZQDJAGoAxwBuAMYAcwDFAHgAwwB9AMIAgwDAAlOAvwCSAL0AmwC7AKUAuQCwALcA
wAC1ANYAtADtALIA/gCxAAP8AsQD/ALEA/wD/AAAA/wAAPwAAADwAAUA6AAQAOAAGgDbA
CMA1wAtANAANGDKAD8AxgBIAMMATwDBAFUAvwBbAL4AYAC8AGUAuwBqALoAbgC5AHMA
twB5ALYAfwC1AIUAswCNALEAlgCvAKAArgCsAKwAugCqAM4AqQDnAKcA+QCmAP8ApgD/AK
YA/wD/AAAA+wAAAO0AAADiAAEA1gALAM8AFQDJAB8AwgAoAL8AMQC9ADoAuQBDALcASg
C1AFEAswBWALEAXACwAGAArwBIAK4AagCtAG8ArAB0AKoAegCpAIEAqACIAKYAkQCkAJwA
owCnAKEAtACfAMYAnQDfAJwA8wCbAP8AmgD/AJoA/wD9AAAA8AAAAN8AADRAAAAxgAHA
L8AEQC3ABsAtAAkALEALQCwADUArQA9AKsARQCpAEsAqABRAKYAVgCIAFsApABgAKMAZ
QCIAGoAoQBvAJ8AdQCeAHwAnACEAJsAjQCZAjCAlwCiAJUARwCTAL8AkQDXAJAA7ACOAPo
AjQD/Al0A/wDzAAAA4QAAANAAAADCAAAAuAADALAADACsABYAqAAfAKYAKACKADAAogA4
AKAAPwCeAEYAnQBMAJsAUQCaAFYAmQBbAJgAYACXAGUAIQBqAJQAcACSAHcAkAB/Al8A
iACNAJMAIwCeAlkAqwCHALoAhQDPAIQA5gCDAPYAggD/AIEA/wDnAAAA0QAAAMAAAAC1A
AAArAAAAKYABwChABEAngAaAJwAlGZACsAlwAyAJUAOGCTAEEAkGBHAJAATACPAFEAjg
BWAlwAWwCLAGEAigBmAlgAbACHAHMAhQB7AIQAhaACCAI8AgACbAH4ApwB8ALYAewDJAH
kA4QB4APEAdwD6AHYA/wDaAAAAxQAAALUAAACpAAAAogAAAJ0AAgCYAAsAIQAVAJIAHQ
CQACUAjQAtAlSAnACJADsAhwBCAIYARwCEAE0AgwBSAIIAVwCBAFwAfwBiAH4AaAB8AG8A
ewB3AHkAgQB3AlwAdQCYAHQApQByALMAcADGAG4A3ABtAO0AawD5AGsA/gDOAAAAugM
AAKsHAACgCAAAmAcAAJMEAACQAAUAjAAOAkAFwCHACAAhAAoAIEALwCAADYAfgA9AH
wAQwB7AEgAegBNAHgAUwB3AFgAdgBeAHQAZQBzAGwAcQB0AG8AfgBtAYkAawGWAGkCo
wBoArIAZgLEAGQC2wBiAuWYgL2AGEC/ADEBwAAsgwAAKMRAACYEgAAkBEAAIoPAACGC
wAAhQUIAIICEQB/AhoAewMiAHkEKgB3BTEAdQU4AHMGPgByBkQAcQdJAG8HTwBuB1UAbA
hbAGsIYgBpCWkAaAlyAGYKfABkC4gAYguVAGAMowBeDLIAXAvFAFsL3ABZDO4AWQz5AFgL/
wC8DwAAqhUAAJwYAACRGQAAiRkAAIIYAAB+FQAAexEAAHoLCgB3CxAUdAwdAHMJQBvD
SwAbQ4zAGsOOQBqDkAAaA9GAGcQSwBIEFEAZBFYAGIRXwBgEWcAXxJwAF0TewBbE4cA
WRSUAFcUogBWFLEAVBTFAFIU3QBRFPAAURT8AFAT/wC2FQAAPBsAAJcdAACMHwAAgh8
AAHweAAB3HAAAcxkAAHEVBABvEw8AbBQZAGkUIQBmFSgAZBYvAGIWNQBhFjsAXxdCAF4
XSABcF04AWxhUAFkYXABYGWQAVhItAFUaeABTG4QAURuSAFAcoABOHLAATBvDAEsb2wB
KG+8ASRr7AEka/wCxGgAAoB8AAJliAACGJAAAfSQAAHYjAABxIQAAbR8AAGocAABnGgoAZB

oVAGEbHQBfGyQAXBwrAFscMgBZHTgAVx0+AFYdRABUHkoAUx5RAFIeWQBQH2EATyBrAE0
gdgBMIYIASiGQAEginwBHIq4ARSHCAEQh2gBDIe4AQyD6AUMf/wGsHwAAncMAAI4mAACCK
AAAEcGAAHEnAABrJgAAZyQAAGMiAABglAYAXSARAFohGQBYISEAVSEoAFMiLgBSIjQAUCI6
AE8jQQBNi0cATCNOAEokVgBJJV8ASCVpAEYmdABFJoAARCeOAUInnQFBJ60BPYfAAT4m2A
E9Je0BPSX5AT0k/wGolwAAmCcAAIopAAB+KwAAAdCsAAGwrAABmKgAAyigAAF4mAABaJQM
AVyUNAFQmFgBRJh4ATyYkAE0nKwBLJzEASSc3AEgoPQBHKEQARShLAEQpUwBDKVwBQS
pmAUArcQE/K34BPiuMATwsmwE7LKsBOSu+ATgr1gE3KuWBNyn5Ajcp/wKkJgAAISoAAIcsAAB7
LgAAcS4AAGkuAABiLQAAXSsAAfKqAABVKgAAUSoJAE4qEwBLKxoASSshAEYrJwBFLC4AQy
w0AEIsOgBALUEAPy1JAD4uUQE9LIoBPC9kATovbwE5MHwCODCKAjcwmgI1MKoCNDC9AjlV1
QEYlUSCMi74AJlt/wKhKQAAkSwAAIMvAAB3MQAAbTEAAGUxAABeMQAAWS8AAAFQuAABPLg
AASy8GAEGvEABFMBcAQzAeAEEwJQA/MCsAPTAXADwxNwA7Mj8AOjJGATkzTwE3M1gBNjNi
AjU0bQI0NHocMjWJAjE1mAlwNakCLzS8Ai0z1AetM+oCLTL3Ai0x/wKeLAAaji8AAIAyAAB0MwA
AajQAAGE0AABaNAAAVTMAAE8yAABKMwAARjMDAEM0DABANBUAPTUCADs1lgA5NSgAOD
YuADc2NQA1NzwANDdEATM3TAEyOFUBMThgAi84awluOXgCLTmHAiw5lwlrOagBKjm6ASg40
gEoN+kCKDb3Aig1/gKbLwAAizIAAH00AABxNgAAZzcAAF43AABXNwAAUDYAAEo2AABFNwAA
QTgAAD45CQA6ORIAODoZADY6HwA0OyYAMzssADE7MwAwPDALzxBAC48SgEsPVMbkz1e
ASo9aQEpPXyBKD6FASc+IQEmPqcBJT25ASM80QEjO+gBlzr2AiM6/gKXMGAAhzUAAHk3AABt
OQAAYzoAAFs6AABTOgAATDoAAEU6AABAPAAAPD0AADg9BgA1Pg4AMj8WADBAHQAUQC
MALUApACxBMAAqQTcAKUE/ACHCRwAnQIEBJkjbASVCZwEkQnQBI0ODASJDIAEHqQUAH0K
4AB5BzwAeQOcBHj/1AR4+/QGTNQAahDgAAHY6AABqOwAAYDwAAFc9AABQPQAAS4AAEA
/AAA7QAAANKIAADJDAgAvRAsALEUTACpFGgAoRiAAJ0YmACZGLQAKRzQAI0c8ACJHRAAh
R04AIEhYAB9IZQAeSHIAHUibABxIkGAbSKQAGke2ABhGzgAYReYAGUT1ABID/QGPOAAfzsA
AHI9AABmPgAAXD8AAFRAAABMQAAARUEAAD1DAAA3RQAAMkYAAAC1IAAAoSgcAJUsPAC
NMFgAhTB0AIEWjAB9NKgAeTTAAHU04ABxNQQAbTkoAGU5VABhOYQAXTm8AFk5+ABVOjw
AUTaEAE020ABFMzAARS+QAEkr0ABNJ/ACKPAAAEz4AAG1AAABiQQAaweIAAFBDAABJRA
AAQkUAADpHAAA0SQAALksAAChNAAAIUAIAHIIKABtTEgAaUxkAGVMfABhUJgAWVCwAFVQ
0ABRUPAATVEYAEIRRABFVXgAQVWsADIR7AA1UjAALVJ4ACIOwAAISxgAJUd4ACIDvAApP+
QCFQAAAdkIAAGIEAABdRQAAVEYAAE1GAABGRwAAPkAADZMAAAwTgAAKIEAACRTAAAE
VgAAGFgFABNaDQARWxQAEFSbAA5bIQANWygADFsvAAtbOAAKXEEACVxMAAdcWAAGW2
YABVt1AARbhgACW5gAAVqrAABZwAAAWNkAAfFqAAFW8wB/RAAAcEYAAGRHAABZSQAau
UkAAEpKAABCTAAAOk4AADJSAAARVQAAJVcAAB9aAAAZXQAAEI8CAAxhCQAHYhAABmMX
AARjHQADYyQAAMMrAAfJmWAAyZwAAGNHAABjUwAAY2AAAGNvAABjgAAAYpMAAGKMAA
BhugAAYNEAAF/IAABe8AB4SQAAa0oAAF9MAABWTQAATk0AAEZPAAA+UgAANVUAAC1YAA
AmXAAAH14AABihAAATZAAADGYAAAAPBwABagwAAGoTAABrGQAAax8AAGwIAABsLQAABD
YAAGxAAAABsTAAAbFkAAGxoAABseQAAa4wAAGugAABqtAAAacoAAGjfAABn6wByTgAAZU8A
AFtQAABTUQAASIIAAEFVAAA4WQAAMFwAACHgAAAGZAAAGWcAABNqAAAMbAAABm4AAA
BwBAAAcQkAAHIOAABzFAAAAdBoAAHUgAAB1JwAAAdi8AAHY5AAB2RAAAAdIIAAHZhAAB2cgAA
dYUAAHWaAAB0rQAAC8MAAHLZAABY5wBrUwAAYFQAADFVAABOVgAARVKAADxdAAAYYQA
AKWUAACFPAAAabQAAE3AAAAxyAAAFdAAAAHcAAAB4AAAAeQUAAHoKAAB8DwAAfRQAA
H4aAAB/IAAAgCgAAIExAACBPAAAGUkAAIFZAACBagAAgH0AAICSAAB/pwAAf7sAAH7QAAB9
4ABmWAAAXVKAFFNaAABJXgAAP2IAADVMAAAAsawAAI28AABtzAAATdwAAC3oAAAR7AAAAf
gAAAIAAAACBAAAAggAAAIQEAACFCQAAhg4AAIgTAACJGQAAiyAAAIwoAACNMwAAjUAAAI1

PAACNYAAAJHQAALyJAACMnwAAi7IAAlvGAACK1QBiXgAAWV8AAE1iAABCZwAAOGwAAC5x
AAAkdgAAG3oAABN+AAAKgQAAAoQAAACGAAAAiAAAAIoAAACMAAAAJQAAAI4AAACQAAA
AkQUAAJMKAAACVEAAAIhcAAJgeAACaKAAAmzQAAJtDAACaVAAAmmgAAJp9AACalAAAmag
AAJm4AACYxgBeZAAAUmgAAEdsAAA8cgAAMXcAACd9AAAdgQAAFIYAAaQKAAABjQAAAI8A
AACRAAAAkWAAAJQAAACWAAAAmAAAAJkAAACbAAAAAnQAAAJ4AAACgBgAAogwAAKQUA
ACmHQAAqSgAAKk2AACpSAAAqVwAAKhxAACoiAAaQJwAAKisAACouQBYbQAATHIAAEF4AA
A1fgAAKoMAAB+JAAAVjgAAC5IAAAKVA AAAAmAAAAJoAAACcAAAAAngAAAKAAAAChAAAAow
AAAKUAAACmAAAAqAAAAKoAAACsAAAAArgIAALAJAACzEgAAth0AALCqAAC4OwAAuE8AALh
kAAC5eQAAuY4AALmeAAC5qQBSdwAARn0AADuEAAuIwAAI5AAABmWAAAOmgAABJ4AAA
CiAAAApAAAAKYAAACoAAAAqgAAAKwAAACuAAAArwAAALEAAACzAAAAAtQAAALcAAAC6AA
AAvAAAAAL8AAADDBAAAxw8AAMwbAADOKwAA0D0AANBTAADQaQAA0H4AANCPAADQmgD
/AAAA/wAAAP8AAAD/AAkA/QATAPgAHgD1ACcA7wAxAOgAOwDjAEQA4ABMANwAUwDaAFk
A1wBfANUAZADTAGgA0gBtANAAcgDPAHcAzQB8AMsAggDKAlkAyACRAMYAmgDEAKUAWgC
xAMAAwgC+ANsAvADxALoA/wC5AP8AuQD/ALgA/wD/AAAA/wAAAP8AAAD5AAUA8QAQAOwA
GgDkACQA3gAtANsANgDVAD8A0QBIAM4ATwDLAFUAYqBaAMcAYADGAGQAxABpAMMAbQ
DBAHIAwAB4AL4AfgC9AIUAuwCNALoAlGc4AKAAtgCsALQAvACyANiAsAdqAK8A+wCuAP8Ar
QD/AK0A/wD/AAAA/wAAAPUAAADqAAIA4gAMANgAFgDQACAAzQApAMoAMgDGADsAwwBD
AMAASgC9AFAAuwBWALoAWwC4AGAAtwBkALUAaQC0AG4AswBzALIAeQCwAIAArwCIAK0A
kQCsAJsAqgCnAKgAtQCmAMkApQDjAKMA9gCiAP8AoQD/AKEA/wD/AAAA9AAAAOgAAADaA
AAAzwAIAMQAEgDAABwAvAAkALoALQC4ADUAtQA+ALIARQCwAEsArgBRAK0AVgCrAFsAqg
BfAKkAZACoAGkApwBuAKYAdACkAHsAowCDAKEAjACfAJYAnQCiAJsArwCaAMEAmADaAJcA
8ACVAP8AIAD/AJQA/wD4AAAA5gAAANUAAADIAAAUwAEALUADgCxABcArgAgAKwAKACqA
DAAqAA4AKYAQACkAEYAogBMAKAAUQCfAFYAnQBAAJwAXwCbAGQAmgBpAJkAbwCXAHY
AlgB+AJQAhwCTAJEAKQCdAI8AqgCNALoAjADQAIoA6QCJAPkAiAD/AlgA/wDqAAAA1wAAAM
UAAAC4AAAArwAAAKkACQCIABMAogAbAJ8AJACeACsAmwAzAJkAOgCXAEeAIQBGAJQATA
CTAFEAKgBVAJAAWgCPAF8AigBIAI0AagCMAHEAigB5AIkAggCHAI0AhQCZAIMApGCBALUAg
ADIAH4A4gB9APMAfAD+AHwA/wDcAAAAxgAAALcAAACrAAAApAAAAJ4ABQCZAA4AlgAXAJ
MAHwCRACYAKAAuAI4ANQCMADsAigBBAlkARwCIAEwAhwBQAIUAVQCEAFoAgwBgAIIAZgC
AAG0AfwB1AH0AfgB7AlkAegCUAHgAogB2ALAAAdADDAHMA2wByAO4AcQD6AHAA/wDOAAAA
ugAAAKoAAACHAAAAmQAAAJQAAACPAAKAjAASAIkAGgCHACEAhQApAIMAMACCADYAgAA
8AH4AQgB9AEcAfABMAHsAUQB5AFYAeABcAHcAYgB2AGkAdABxAHIAegBwAIUAbwCRAG0A
ngBrAKwAaQC+AGgA1ABnAOKAZgD1AGUA/ADDAAAArwAAAKICAACXAwAAjwIAAIoAAACGA
AQAgwAMAIAAFQB+ABwAfAAkAHkAKwB4ADEAdgA3AHQAPQBzAEIAcgBHAHEATQBvAFIAb
gBYAG0AXgBrAGYAaQBwAGgAdwBmAIIAZACOAGIANAbhAKoAXwC7AF4A0QBdAOUAWwD0
AFsA+wC6AQAApWgAAJoLAACPDAAAhwwAAIELAAB9BwAAewIHAHgADwB2ABcAcwAfAHEA
JgBvACwAbQAZAGsAOABqAD4AaABDAGcASQBmAE4AZAFUAGMBWwBiAmMAYAJrAF4DdQ
BdA4AAWwSNAFkFmwBxBaoAVgS6AFQE0ABTBOUAUgTyAFIE+QCyCgAAoQ8AAJQSAACIF
AAafxQAAHkTAAB1EQAAcg0AAHEHCQBUBBEAawUaAGkFIQBmBigAZAcuAGMHNABhBzoAY
Ag/AF8IRQBdCUsAXAIRAFoKWABZCmAAVwtpAFYLCwBUDH8AUg2NAFANmwBPDaoATQ28A
EsN0gBKDecASg30AEkM/ACrEQAAmxUAAI4YAACCGgAAeRoAAHIZAABuFwAAahUAAGgRAG
BnDQsAZAwUAGENHABfDiMAXQ4qAFsOMABZDzYAWBA8AFcQQgBVEUgAVBFOAFISVgBRE
I4ATxNnAE0TcgBMFH4AShWLAEGvmgBHFakARRW7AEQU0gBDFOGAQXT3AEIU/wCmFgAAIx

oAAlkdAAB9HgAAdB8AAGweAABnHQAAYxsAAGAYAABfQYAXRQQAFoUGABXFR8AVRUmA
FMWLABSFjIAUBY4AE8XPgBNF0QATBhLAEoYUwBJGVsARxIIAEYabwBEG3sAQxuJAEEmA
BAHKgAPxu5AD0b0AA8G+cAPBr2ADwa/wCiGgAAkx0AAIqHAAAB5lgAAbyMAAGcjAABiIQAAXh8
AAFodAABYGwEAVRoLAFMaFABQGxwAThsiAEwbKQBKHC4ASRw0AEccOgBGHUEARR1IAE
MeUABCHlgAQR9iAD8gbQA+IHkAPSGHADshlgA6IacAOCG4ADcgzwA2IOYANh/1ADYf/gCeHQ
AAjyEAAIEkAAB1JQAAayYAAGMmAABdJQAAWCQA AFUHAABSIAAATyAIAEwfEQBJlBgARyAf
AEUgJQBEISsAQiExAEehNwA/lj4APiJFAD0jTQA8JFYAOyRgADklawA4JXcANyaFADUmlQA0Jq
UAMya3ADEIzQA wJeUAMCT0ADaj/QGblAAaiyQAAH0mAABxKAAAZykAAf8pAABZKQAAVCc
AAFAIAABMJAAASSUEAEYkDQBDJRUAQSUcAD8llgA9JSgAPCYuADomNAA5JzsAOCdCADco
SwA2KFQANSleADMqaQAYKnYAMSqEADArkwAuK6QALsq2ACsqzAArKeQAKyj0ASso/QGYlw
AAiCcAAHopAABuKwAAZCwAAFwsAABVLAAAUcsAAEwpAABHKQAARckBAEApCgA+KRiAO
yoZADkqHwA3KiUANSorADQrMQAzLDgAMixAADEtSAAwLVEALy5cAC4uZwAtL3QAKy+CACov
kgApL6MAKC+0ACYuywAmLeMAJi3zASys/AGVJgAAhSkAAHcsAABrLQAAYS4AAfKvAABSLw
AATC4AAEctAABDLQAAPy0AADsuBwA4Lg8ANi8WADMvHQA xLyIAMD AoAC8wLwAuMTYALTE
9ACwyRgAqMk8AKTNZACgzZQAnM3IAJjSAACU0kAAkNKEAlzOzACEzyQA hMulAIthYASEw/A
GRKAAAgSwAAHMuAABnMAAAXTEAAFUyAABOMgAASTEAAEMxAAA+MQAAOjIAADYzBAAz
MwwAMDQTAC40GgAsNSAAKjUmACk1LQAoNjMAJzY7ACY3QwAIN00AJDdXACM4YwAiOHAA
ITh+ACA4jwAfOKAAHjiyABw3yAAcNuEAHDxYABw1+wGOKwAAfi8AAHAXAABkMwAAWjQAAFI
0AABLNAARTQAAD81AAA5NgAANDcAADA4AAAtOAgAKjkQACg6FwAmOh4AJTskACQ7KgAi
OzEAITw4ACA8QAafPEoAHjxUAB09YAAcPW0AGz18ABo9jQAZPZ4AGDywABc8xgAWO+AAFz
rxABc5+wCKLwAAejIAAGw0AABgNgAAVzcAAE83AABINwAAQjgAADs4AAA0OgAALzsAACs9A
AAnPgUAJD8NACJAFAAgQB sAH0AhAB5BJwAcQS4AG0E1ABpBPQAZQkcAGEJRABdCXQAW
QmsAFUJ6ABRCigATQpwAEkGvABFBxAAQQN4AET/wABE++gCFMgAAadjUAAGg3AABdOQAA
UzkAAEs6AABFOgAAPzsAADg8AAAxPgAALD8AACdBAAAiQwEAHkUJABtGEQAZRhAGEceA
BdHJAAWRyoAFUcxABRHOGASSEMAEUhOABBIWgAPSGgADUh3AAxIhwALR5kACkerAAIGvw
AIRdgACUTrAApD9wCANgAAcTgAAGQ6AABZPAAAUDwAAEg9AABCPQAAPD4AADVAAAAuQ
gAAKEQAACNGAAAeSAAAGESFABRNDAASTRMAEE4aAA9OIAAOTiYADE4uAAtONgAKTj8A
CU5JAAhOVQAHTmMABU5xAAROGgADTZQA Ak2nAAFMugABS9IAAUrmAAFK8QB7OgAABDw
AAF8+AABVPwAATEAAAEVAAAA/QQAAOEIAADFEAAAqRwAAJEkAAB9MAAAZTgAAFFEBAA
1TCAAJVA8AB1UWAAZVHA AFVSM AA1UpAAJVMQABVT oAAFVEAABVUAAAVV0AAfVsAABV
fQAAVI8AAFSiAABTtQA AUswAAFHiAABQ7wB1PgAAZ0AAAFtBAABRQgAASUMAAENDAAA8R
QAANEcAAC1KAAAmTQA AIFAAABpSAAAUVQAADlC AAAhaBwACWwwAAFwTAABcGAAAXR4
AAF0IAABdLAAAXTUAAF0/AABdSwAAXVgAAF1mAABcdwAAXIoAAFydAABbsQA AWsYAAFnd
AABY6gBvQgAAYUQA AFZFAABORgAAR0YAAD9IAAA3SgAAL00AAChRAAAhVAAAGlcAABRa
AAAOXAAACF8AAAJhBQAAYgoAAGMQAABkFQAAZBoAAGUgAABmJwAAZjAAAGY5AABmR
QAAZIEAAGZgAABlcQAAZYMAAGSYAABkqWAAy8EAAGLYAABh6ABoRwAAXEkAAfJJAABLS
gAAQ0sAADtOAAAyUQA AKIUACJYAAA bXAAAFV8AAA5iAAAIZAAAAWcAAABpAgAAagcAAG
sLAABsEAAA bRUAAAG4bAABvIQAAcCkAAHAY AABwPQAACeOAAHBZAABwaQAAb3wAAG+RA
ABupQAAbboAAGzQAABs4wBiTAAAWEOAAFB OAABHTwAAPIIAADVWAAA sWgAAJF4AABxiA
AAVZQAADWgAAAdrAAAAbQAAAHAAAABxAAAACwIAAHQGAAB1CwAAadhAAAHgVAAB5GwA
AeilAAHsqAAB8NQAAfEEAAHtQAAB7YQA Ae3MAAHqIAAB6ngAAebIAAHjHAAB42gBdUQA AVV
IAAExTAABCvgAAOFoAAC5fAAAIYwAAHWgAABVsAAANcAAABXMAAAB1AAAA dwAAAHkAA

AB7AAAAfAAAAH4BAAB/BQAAgQkAAIIIOAACEFAAAhRsAAIciAACILAAAIgAAIhHAACIVwAAI
GoAAId/AACHIQAahqoAAIW+AACFzwBaVwAAUUVgAAEZbAAA8XwAAMWQAACdQAAEbwAAF
XMAAAx3AAAEewAAAH0AAAB/AAAAGQAAAIMAAACFAAAAhwAAAIgAAACKAAAAiwlAAI0HAA
CPDAAAkRMAAJMaAACVlwAAIi4AAJY8AACWTQAAImAAAJV1AACVwAAIKEAAJO0AACTxAB
WXQAAS2AAAEBlAAA1agAAKnAAACB2AAAWewAADX8AAASDAAAahgAAAIgAAACKAAAAjA
AAAI4AAACQAAAAkgAAAJQAAACVAAAAlwAAAJkAAACbAgAAnggAAKAPAACiGAAApSIAAKU
vAACIQAAApVMAAKVoAACkfgAApJQAAKOnAACjtQBQZgAARWsAADlxAAAudgAAI30AABmCA
AAOhwAABIsAAACPAAAAkgAAAJQAAACWAAAAmAAAAJoAAACcAAAAAnQAAAJ8AAACHAAA
AowAAAKUAAACnAAAAqgAAAKwEAACvDAAAshcAALUjAAC1MwAAAtUYAALRbAAC0cQAAtlgA
ALSaAACzpwBKcAAAPnYAADN9AAAngwAAHlkAABGPAAAGIAAAAjgAAACbAAAAngAAAJ8A
AACiAAAApAAAAKUAAACnAAAAqQAAAKsAAACtAAAArwAAALEAAACzAAAAtgAAALkAAAC8
AQAAwAsAAMUXAADGJgAAxjkAAMdPAADIZAAAYXgAAMmKAADKmAAD/AAAA/wAAAP8AAD/
AAUA/AAQAPEAGgDuACQA6wAtAOgANgDjAD8A3wBHANsATgDZAFQA1QBaNMAXwDRAG
QA0ABoAM4AbQDMAHEAywB3AMkAfQDHAIQAxQCMAMMAIQDBAKAAvwCtAL0AvgC7ANcAu
QDwALcA/wC2AP8AtQD/ALQA/wD/AAAA/wAAAP4AAD1AAIA6wAMAOMAFgDfACAA3AApANk
AMgDTADsAzwBDAMsASgDJAFAAxgBVAMQAWgDDAF8AwQBkAMAAaAC+AG0AvABYALoAe
AC4AH8AtgCHALQAKACzAJsAsQCoAK8AtwCtAM0ArADpAKsA/ACqAP8AqQD/AKgA/wD/AAAA/
AAAAPAAAADmAAAA1wAIAAAEWDMABwAyQAIA MYALQDDADYAvwA+ALwARQC4AEsAtgB
RALQAVgCyAFoAsQBfAK8AYwCuAGgArQBtAKwAcwCqAHoAqQCCAKcAiwCmAJYApACjAKIAs
QChAMUAnwDgAJ4A9gCdAP8AnAD/AJsA/wD9AAAA7wAAAOAAAADOAAAAXQAFAL8ADgC6A
BgAtwAhALUAKQCyADEArgA5AKwAQACqAEYAqABMAKcAUQCIAFUApABaAKMAXgCiAGMA
oQBpAJ8AbgCeAHUAnQB9AJsAhgCaAJEAmACdAJYAqwCUAL0AkWdWAJE7wCRAP4AkAD/
AI8A/wDvAAAA3QAAAM0AAAC9AAAAAtQABAK8ACgCrABMAqAAcAKUAJACjACwAogAzAJ8AO
gCdAEEAnABGAJoASwCZAFAAmABVAJcAWQCVAf4AIABjAJMAaQCSAHAAkAB4AI8AgQCNA
lwAiwCYAlOApGCIALYAhwDMAIUUA5wCEAPkAgwD/AIMA/wDhAAAAyWAAALoAAACvAAAApWA
AAKIABwCdAA8AmgAXAJgAIAcWACcAIQAuAJMANQCRADsAjwBBAl4ARgCNAEsAiwBQAlOa
VACJAFkAiABfAlcAZQCFAgSAhABzAIIAfACBAIcAfwCTAH0AoQB8ALAAegDEAHkA3gB4APIAd
wD/AHYA/wDPAAAAvAAAAK0AAACjAAAAAnAAAAJUAAwCRAAsAjgATAlwAGwCKACIAiQApAlc
AMACFADYAhAA8AIIAQQCBAEYAqABLAH4ATwB9AFQAfABaAHsAYAB5AGcAeABvAHYAeAB
1AIMAcwCPAHEAnABvAKsAbgC9AG0A1QBrAO0AawD6AGoA/wDDAAAArwAAAKIAAACyAAA
AkAAAAIoAAACGAACAgwAPAIEAFwB/AB4AfQAIAHwAKwB6ADEAeAA3AHcAPAB1AEEAdABG
AHMASwBxAFAAcABWAG8AXABuAGMABABrAGsAdABpAH8AaACLAGYAmQBkAKcAYwC4AG
IAzgBgAOYAYAD1AF8A/QC3AAAApWAAAJkAAACOOAAAAhgAAAIIEAAAB9AAIAegAKAHcAEgB1
ABkAcwAgAHEAJgBvAC0AbgAyAGwANwBrAD0AagBCAGgARwBnAEwAZgBRAGUAWABjAF8
AYgBnAGAAcQBfAHsAXQCIAFwAIQBAAKQAWQC0AFcAyQBWAOEAVgDwAFUA+QCuAAAAAn
wIAAJEGAACGCAAAfggAAHgGAAB0AwAAcgAFAG8ADQBABQAagAbAGGAlgBmACgAZQAuA
GMAMwBhADgAYAA9AF8AQgBeAEgAXABOAFsAVABaAFwAWABkAFcAbgBVAHkAVACFAFIA
kwBRAKIATwCyAE4XgBNANwATADuAEsA9wCoBQAAMAoAAIoNAAB/DwAADg8AAHAOAABr
DAAAaAkAAGcEBwBIAQ8AYgAWAGAAHQBeACQAXAApAFoALwBZATQAWAE5AFYBPwBVAK
QAVAJKAFIDUQBRA1kAUARiAE4FbABNBncASwaEAEkHkgBIB6IARgeyAEUGxQBEBd0AQwbs
AEMG9QCICwAAkhAAAIQTAAB5FQAACBUAAGkVAABkEwAAYREAAf8NAQBeCQkAXAYRAfK
GGQBWBx8AVAcIAFMikwBRCDAAUAK1AE4JOWBNcUEATApIAEsLTwBJC1cASAXgAEYNawB

FDXcAQw6EAEEOkwBAD6MAPg60AD0OyAA8Dd8AOw3vADsN+ACdEAAAJRUAAH8YAAB0GQ
AAahoAAGMaAABeGQAAWhcAAFcUAABWEQMAVQ4LAFINFABQDhsATg4hAEwPJwBKEC0A
SRAyAEcQOABGET4ARBFFAEMSTABCEIUQBNeAD8UaQA9FHUAPBWCADoVkQA5FqEANx
WyADYVxwA1FN8ANRTxADUU+wCZFAAAiRkAAHsbAABvHQAAR4AAAF4eAABYHQAAVBsAA
FEZAABPFwAATRUHAESUEABIFBcARhUdAEQVIwBDFSkaQRYvAEAWNAA/FjsAPRdCADwYS
QA7GFIAORlcADgaZwA3GnMANRuAADQbkAAyG6AAMRuxAC8axgAuGt4ALhnwAC4Z+wCVFw
AAhRwAAHceAABrIAAAySEAAFOhAABUIQAATyAAAEweAABJGwAARxoDAEQaCwBCGHMAQ
BoaAD4alAA8GyYAOxsRADkbMQA4HDgANxw/ADUdRwA0HIAAMx5ZADIfZAAwH3EALyB/AC4gj
gAsIJ4AKyCvACkfxAAoH90AKR7vACke+gCSGgAAgR8AAHMhAABnlwAAXiQAAFYkAABQJAAA
SyMAAEciAABFHwAAQR8AAD4fCAA8HxAAOh8XADcfHQA2HyMANCAoADlglGAXiTUAMCE8A
C8iRAAUlk0ALSNXACwkYgAqJG8AKSV9ACgljAAnJZ0AJSWuACQkwwAjI9wAlyPuACQi+gCOH
QAAfiEAAHAKAABkJgAAWycAAFMnAABNJwAARyYAAEMIAABAlwAAPCMAADkjBAA2JAwANC
QUADEkGgAvJCAALSQIACwIKwArJTIAKiY5ACknQgAoJ0sAJyhVACYoYAAIKW0AJC17ACMpiw
AiKZwAICmtAB8owgAeJ9oAHifuAB8m+QCLIAAAeyQAAG0mAABhKAAAWCkAAFAqAABJKgAA
RCkAAD8oAAA7JwAANygAADMoAQAwKAKALikRACspFwApKR0AKCojACcqKQAmKjAAJSs3A
CQsPwAjLEgAli1TACetXgAgLWsAHy55AB4uiQAcLpoAGy2sABotwAAZLNkAGSvtABoq+ACHlw
AAdyYAAgopAABeKwAAVSwAAE0sAABGLAAaQSwAADwsAAA3KwAAMiwAAC4tAAArLQYAK
C4NACYuFAAkLxsAly8hACEvJwAgMC0AHZa0AB4xPQAdMUyAHDJQABsyXAAaMmkAGTJ3AB
gyhwAXMpkAFjKrABUxvwATMNgAFC/sABUv+ACEJgAAAdCkAAGYrAABbLQAAUS4AAEovAABD
LwAAPi8AADkvAAAZLwAALDEAAcKyAAAmMwMAIzMKACA0EgAeNRgAHTUeABw1JAAbNSsA
GjYyABk2OgAYNkMAFzdOABU3WQAUN2YAFDd1ABI3hQARN5cAEDepAA82vQANNdUADjTrA
A8z9wCAKQAAcCwAAGluAABXMAAATjEAAEcxAABAMQAAOzIAADUyAAAvMwAAKTUAACU2
AAAhOAAAHTKHABo6DgAYOxUAFzsbABY7IQAVOygAEzwvABI8NwARPEAAEDxLAA88VgAOP
GMADDxyAAs8ggAKPJMACTymAAg7uAAHOS8ABznIAAg58gB7LAAAbC8AAF8xAABUMwAASz
QAAEQ0AAA9NAAAODUAADI1AAAsNwAAJjkAACI6AAAdPAAAGD4DABRACgARQREAEEEYA
A5BHgANQSQADElrAAtCMwAKQjwACUJGAAdCUgAGQl8ABUJtAARCFQADQY8AAkGiAAFAAtA
ABP8oAAT7hAAE+7gB2MAAAZzMAAFo1AABQNgaASDcAAEE3AAA7NwAANjgAAC85AAApOw
AAIz0AAB4/AAAZQQAAFEMBAA5GBwAKRw4ACEcVAAZHGWAFSCeABEgoAANILwACSDgAA
EhCAABITQAASFoAAEhpAABleQAAR4sAAEeeAABGsAAARcYAAETcAABD7ABxNAAAYjYAAF
Y4AABMOQAARToAAD46AAA5OgAAMjsAACs9AAAIQAAAH0IAABpFAAAVRwAAEEkAAAIMBg
AETQwAAE4SAABOGAAATx4AAE8kAABPKwAATzMAAE89AABPSAAAT1UAAE9kAABOdAAA
ToYAAE6ZAABNrQAATMIAAEvZAABK6ABrOAAAXToAAFI8AABJPAAAQj0AADw9AAA1PgAALk
AAACdDAAAhRgAAG0gAABVLAAAQTQAACIAAAARSBQAAUwoAAFQPAABVFAAAVhoAAFcg
AABXJgAAVY4AAFc4AABXQwAAV1AAAFdeAABWbgAAVoAAAFWUAABVqAAAVL0AAFPUAAB
S5gBIPQAAWD4AAE4/AABGQAAAQEAAADIBAAAxQwAAKUyAACJJAAAcTQAAFIAAABBSAA
AKVQAABFCAAABZAwAAWwgAAFWMAABdEQAAxHYAAf8bAABglgAAYCkAAGAYaABgPQAA
YEOAAGBYAABfaAAAX3oAAF6PAABepAAAXbgAAfzPAABb4wBfQQAu0IAAEpDAABEQwAA
PEQAADRHAAsSgAAJE4AAB1RAAAWVQAAEFgAAAlaAAADXQAAAF8AAABhAAAAyWQAAG
QIAABIDAAAZhEAAGgXAABpHQAAaiMAAGosAABqNgAAakMAAGpRAABqYQAAaXMAAGmIA
ABonQAAZ7IAAGblAABm3QBZRgAAT0cAAEHHAABASAAAN0sAAC5OAAAmUgAAHIYAABdaA
AAPXgAACGEAAAJkAAAAZgAAAGkAAABqAAAAbAAAAG0EAABvCAAAcAwAAHERAABzFwA
AdR4AAHYIAAB2LwAAAdjsAAHZJAAB1WQAAdWsAAHSAAAB0lgAAc6sAAHLAAABx1ABUSwAA

TUsAAEVMAAA7TwAAMVMAAChYAAAfXAAAF2EAAA9IAAAHaQAAAGsAAABuAAAAcQAAAH
MAAAB1AAAAAdgAAAHgAAAB6AQAAewUAAH0JAAB/DwAAgRUAAIMcAACEJQAAhDEAAIQ+A
ACETgAAg2EAAIN1AACCiwAAgaEAAIC1AACAyABSUAAAASVEAAD9UAAA1WAAAKI0AACFjAA
AYZwAAD2wAAAzwAAAAdAAAAHYAAAB5AAAAewAAAH0AAAB/AAAAgQAAAIMAAACFAAAA
hgAAAIgCAACKBwAAjQ0AAI8UAACRHQAakycAAJM0AACSRAAAKIYAAJFrAACRgQAAkJcAAI
+rAACOVABOVQAAQ1kAADIdAAAuYwAAI2kAABlvAAAQdAAABnKAAAB9AAAAGAAAAIIAAACE
AAAAhgAAAIgAAACKAAAAjAAAAI4AAACQAAAAkgAAAJQAAACWAAAAmQQAAsLAAACeFAA
AoR0AAKlqAACiOQAAoUwAAKFgAACgdgAAn40AAJ+hAACesQBIXgAAPWMAADJpAAAncAAA
HHYAABJ8AAAHgQAAAIUAAACJAAAAiwAAAI0AAACQAAAAkgAAAJQAAACXAAAAmQQAAs
AAACcAAAAAnwAAAKEAAACjAAAApgAAAKkAAACsBwAAArxEALMdAACzLAAAsj0AALJSAACx
aAAAsH4AALCTAACvowBCaQAAN28AACT2AAAgfAAAFYMAAAqJAAAAjQAAAJIAAACWAAAAI
wAAAJ0AAACcAAAAngAAAKAAAACiAAAApAAAAKYAAACoAAAAqgAAAK0AAACvAAAAsgAAA
LYAAAC6AAAAvgUAAMMRAADGHwAAxTAAAMVFAADEWwAAw3EAAMOGAADDIQD/AAAA/w
AAAP8AAAD+AAIA9QAMAPEAFgDuACAA7AApAOgAMgDjADsA3QBCANcASQDTAFAA0ABVA
M4AWgDMAF8AygBjAMgAaADHAG0AxQByAMQAeADCAH8AwACHAL4AkQC8AJwAugCpALgA
ugC2ANMATADvALIA/wCxAP8AsAD/AK0A/wD/AAAA/wAAAPsAAADtAAAA5gAIAOEAEwDdABw
A2gAIANUALgDNADYAyQA+AMUARQDDAEsAwABQAL8AVQC9AFoAuWbFALoAYwC4AGgAtw
BtALYAcwC0AHoAsgCCALEAjACvAJcArQCkAKsAswCpAMoAqADoAKYA/QCIAP8ApAD/AKMA/
wD/AAAA9wAAAOoAAADbAAAA0gAFAMwADgDHABgAwQAhAMAAKQC9ADEAuQA5ALYAQA
C0AEYAsgBMALAAUQCvAFUArQBAAKwAXgCrAGMAqQB0AKgAbgCnAHUApQB9AKQAhgCiAJ
EAoACeAJ8ArQCdAMEAmwDfAJ0A9wCZAP8AmAD/AJcA/wD4AAAA6AAAAANMAAADHAAAAvw
ABALkACwCzABQAsAAdAK8AJQCuACwAqgA0AKgAOwCmAEEApABGAKIASwChAFAAoABVA
J8AWQCdAF4AnABjAJsAaQCaAHAAMAB4AJcAgQCVAIwAkwCZAJEApwCQALkAjgDTAI0A7w
CMAP8AiWd/AlsA/wDoAAAA1AAAAAMEAAAC2AAAAArgAAAKcABwCkABAAoQAYAKAAIACeACg
AnQAvAJ0ANQCYADwAlgBBAJUARgCUAEsAkGBPBJEAVACQAFkAjgBeAI0AZACLAGsAigByAI
kAfACHAlcAhQCTAIQAoQCCALIAGQDIAH8A5gB/APoAfgD/AH0A/wDWAAAAvwAAALEAAACoA
AAAnwAAAJ0AAwCWAAAsAkWUAJIAHACQACMAjwAqAI0AMACKADYAiAA8AIYAQQCFAEYAh
ABLAIMATwCCAFQAgABZAH8AXwB+AGYAfABtAHsAdwB6AIIAeACOAHYAnAB1AKwAcwDAA
HIA3ABxAPMAcAD/AHAA/wDEAAAAsQAAAKUAAACbAAAAkwAAAI0AAACJAAgAhwAQAIUAF
wCCAB8AgQAIAlAALAB+ADEFaA3AHoAPAB5AEEAeABFAHcASgB2AE8AdABUAHMAWgByA
GEAcABpAG8AcgBtAH0AbACJAGoAlwBpAKcAZwC5AGYA0gBIAOsAZQD7AGQA/wC2AAAApw
AAAJ0AAACPAAAAiAAAAIIAAAB+AAQAewAMAHgAEwB2AB0AdQAHAHQAJwByAC0AcQAYAG
8ANwBuADwAbABBAGsARQBqAEoAaQBQAGcAVgBmAF0AZQBIAGMAbgBiAHgAYACFAF8Ak
wBdAKIAXACzAFsAyQBAAOQAWQD1AFkA/wCtAAAAngAAAJAAACFAAAAAfgAAAHgAAAB0A
AAAcAAIAG4ADwBsABYAagAcAGkAlGBoACgAZgAtAGQAMgBjADcAYgA8AGEAQQBfAEYAXg
BLAF0AUgBcAFkAWgBhAFkAagBXAHUAVgCBAFQAJwBTaj4AUgCvAFEAwWbQANwATwDvA
E8A+gCIAAAAlgAAAIgCAAB9AwAAAdQQAAG4DAABqAAAAaAADAGUACgBjABEAYQAYAF8AH
gBeACMAXQApAFsALgBZADMAWAA3AFcAPABVAEIAVABHAFMATgBSAFUAUABdAE8AZwB
OAHITAB+AEsAjABKAJsASACrAEcAvgBGANUARgDqAEUA9QCfAAAAjgUAAIEIAAB2CgAAb
AsAAGYKAABiCAAAXwUAAF0BBgBbAAwAWQATAFcAGQBVBAB8AVAAIAFIKGBRAC8ATwAz
AE4AOABNAD4ATABEAEoASgBJAFIASABaAEYAZABFAG8ARAB8AEIAigBBAJkAQACpAD8Au
wA+ANEAPQDIAD0A8gCZBQAAiQsAAHsOAABvEAAAZhEAAF8RAABaEAAAVw0AAFUKAABU

BgcAUgMOAFABFQBOARsATAEhAEoCJgBJAisASAlwAEYDNQBFZAzoARANBAEMERwBBBU8
AQAVYAD8GYgA9B24APAd7ADsliQA5CJkAOAipADcHuW2B88ANQbIADQG8QCUCgAAhBAA
AHYSAABqFQAAYRUAAFoVAABUFQAAURMAAE4RAABMDQIATAoJAEoIEABHCBcARQgdAE
QllgBCCScAQKsAD8JMgA+CjcAPQo+ADwLRQA6C04AOQxXADcNYgA2Dm4ANA57ADMPigA
xD5sAMA+rAC4OvgAtDdQALA3nACwN8wCPEAAAFxMAAHEWAABIGAAAXBkAAFUZAABPGQ
AASxgAAEGWAABGEwAARBEEAEMOCgBBDHlAPw4YAD0OHgA8DyQAOhApADkQLwA3EDUA
NhE8ADURQwAzEkWAMhNVADATYAAvFGwALRV5ACwViQAqFZkAKRWqACgVvQAmFNQAJh
PpACYT9QCLEwAAexYAAG0ZAABiGwAAWBwAAFEaAABLHAAARxsAAEMaAABAGAAAPxYA
ADwVBgA7FA4AOBQVADYUGwA1FSEAMxUmADIVLAAwFjIALxY5AC4XQAAsGEkAKxhTACoZ
XgAoGWOAJxp4ACYahwAlGpcAlxqpAClauwAgGdMAIBjpACEY9gCIfgAAeBkAAGocAABeHgAA
VR8AAE0gAABIHwAAQx8AAD8eAAA8HAAAORoAADcaAgA0GQoAMhkRADAZFwAuGR0ALBoj
ACsaKQAqGy8AKBs2ACccPgAmHUcAJR1RACQeXAAjHmgAlh92ACafhQAfhH5YAHh+oAB0eug
AbHdIAGx3oABwc9QCEGAAAdBwAAGYfAABbIQAAUiIAAEoiAABElgAAPyEAADshAAA3IAAAN
B4AADEeAAAuHgcALB4OACkeFAAnHhoAJR4gACUfJgAkICwAlYAzACIhOwAhIUQAICJOAB8iW
gAdl2YAHcN0ABsjhAAal5UAGSOMABgiuQAWItAAFiHnABcg9QCBGwAAcR4AAGMhAABYlWwA
ATyQAAEcIAABBJAAAPCQAADgkAAAZlwAALyIAACwiAAAPlwQAJiMLACQjEQAiJBcAICQdAB8
klwAeJSoAHSUxABwmOQAbJkIAGidMABknVwAYJ2QAFyhyABYoggAVKJMAFCelABMnuAARJ
s8AESXmABIk9AB9HgAAbiEAAGAKAABVJQAATCYAAEQnAAA+JwAAOScAADQmAAAwJgAA
KyYAAcYnAAAJKAAAlSglAB4pDgAcKRUAAGykbABoqlQAZKicAGCouABcrNgAWKz8AFSxKABQ
sVQATLGLIAEixwABAtgAAPLJIADiykAA0rtQALKssACyriAAwp8QB6IAAAaiQAAF0mAABSKAAAS
SkAAElpAAA8KQAANikAADEpAAAsKQAAJyoAACIsAAAFLLQAAHC0FABkuCwAXLxIAFS8YABQ
wHwATMCUAEjAsABEWmWaqMTwADjFHAA0xUgAMMV8ACzFtAAoxfAAJMY4ACDGgAAcwsgA
GL8YABi7eAAYu7AB2lwAAZiYAAfKpAABPKgAARisAAD8sAAA5LAAAMyWAAc4sAAAqLAAAJC
0AAB8vAAAcMQAAGDIBABM0CAAQNQ8ADjUVAA02HAAMNiIACzYpAAo2MAAJNjkACDZDAA
c2TgAFN1sABDdpAAM3eAACNooAATacAAA1rgAANMMAADPaAAAY6gBxJgAAYikAAFUUsAABL
LQAAQy4AADwuAAA2LgAAMS4AACwuAAAnLwAAIjEAAB0zAAAYNQAAFDcBAA85BgAKOwwA
BzsTAAY7GQAFOx8ABDwlAAI8LQABPDUAADw/AAA8SgAAPFYAADxIAAA8dAAAPIYAADuZA
AA7qwAAOr8AADnWAAA45wBsKgAAxi0AAFIvAABIMAAQDEAADkxAAA0MQAALzEAACkyAA
AkNAAAHjYAABk4AAAVOGAAEDwAAAo+BgAFQAsAAUERAABBFgAAQhwAAELiAABCKQAAQj
EAAEI6AABCRQAAQIIAAEJgAABCCAAAQYIAAEGVAABAqAAAP7wAAD7TAAA+5gBnLgAAWT
EAAE0yAABEMwAAPTQAADc0AAAYNAAALDUAACY3AAAgOQAAGjsAABU9AAAQQAAC0IA
AAVEBQAARgoAAEcOAABHEwAASBkAAEkfAABJJQAASS0AAEk2AABJQQAASU0AAEIbAABJ
awAASH0AAEIraABHpQAARrkAAEXQAABE5ABhMgAAVDUAAEk2AABBNgAAOjcAADU3AAAv
OAAAKDoAACI8AAAcPwAAfKEAABFEAAALRgAABkgAAABLAWAATAgAAE0MAABOEQAATxU
AAFAbAABRIQAAUSgAAFExAABRPAAAUUgAAFFWAABQZgAAUHGAAE+MAABPoQAATrUAA
E3MAABM4gBbNwAATzkAAEY5AAA+OgAAOToAADi7AAARpQAAJD8AAB1CAAAXRQAAEUgA
AAtLAAAFtQAAAFAAAABSAQAAUwUAAFUJAABWDQAAVxIAAFgXAABaHQAAWyMAAFssAA
BaNgAAWkIAAFpQAABaYAAAWXIAAFiGAABYmwAAV7AAAFbHAABV3gBVOWAASz0AAEM9A
AA9PQAANj4AAC1AAAAmQwAAH0cAABhKAAARTgAACIAAAARTAAAAVgAAAFkAAABaAAAA
XAEAAf0FAABfCQAAYAOAAGISAABJfWAAZR4AAGYmAABILwAAZTsAAGVJAABIWQAAZGoA
AGN/AABiIQAAYaoAAGDAAABg2ABQQAAAR0AAAEFAAAA6QQAAMUQAACHAAAGSwAAGU
8AABFTAAAKVwAAA1oAAABdAAAAAXwAAAGIAAABkAAAAZQAAAGcAAABpBAAAaggAAGwM

AABuEgAAcBgAAHIfAABByKAAAcjQAAHFBAABxUQAAcGIAAHB3AABvjQAAbqMAAG24AABszg
BMRAAARUQAAD5FAAA0SAAAK0wAACJQAAAQVQAAElOAAAlEAAACYQAAAGQAAABnAAAA
agAAAGwAAABuAAAAcAAAAHIAAAB0AAAAAdQIAAHcGAAB5CwAAexEAAH4YAACAIQAAfysAA
H85AAB/SAAAfloAAH5uAAB9hAAAfJoAAHuuAAB6wwBKSQAAQkkaADhNAAAUUQAAJFYAABt
bAAASYQAACWUAAABqAAAAbQAAAHAAAAABzAAAAAdgAAAHgAAAB6AAAAfAAAAH4AAACAA
AAAggAAAIQAAACGAgAAiQcAAIsOAACOFgAAkCAAAJAtAACPPAAAJ00AAI5hAACNdWAAjI0A
AluiAACKtABHTgAAPVIAADJWAAAnXAAAHWIAABNoAAAJbQAAAHIAAAB3AAAAeQAAAH0AA
AB/AAAAgQAAAIQAAACGAAAAiAAAAIoAAACMAAAAJgAAAJEAAACTAAAAIQAAAJgFAACbDQ
AAncxAKAjAACgMQAAAn0MAAJ9WAACebAAAnIIAAJuXAACbqABBVwAANlwAACtiAAAgAAA
FW8AAAp1AAAAewAAAAIAAAACEAAAAhgAAAlgAAACLAAAAjQAAAI8AAACRAAAAIAAAAJYAA
ACYAAAAmgAAAJ0AAACfAAAAogAAAKUAAACoBAAArA0AALAYAACwJwAAsDcAAK9LAACuY
QAARXcAAKyMAACrngA7YQAAL2gAACRvAAAZdgAADX0AAAODAAAAiAAAAlwAAACQAAAAk
QAAAJQAAACXAAAAmQAAAJsAAACeAAAAoAAAAKIAAACIAAAApwAAAKoAAACsAAAArWAA
ALMAAAC3AAAAvAAAMELAADGGQAAXSkAAMQ9AADDUgAAwmngAAMF9AADAjgD/AAAA/w
AAP8AAD6AAAA9QAIAPeAEwDrABwA5wAIAOQALgDfADYA2gA+ANUARQDSAEsA0ABRA
M0AVgDLAFoAyQBfAMgAYwDGAGgAxABuAMIAcwDBAHoAvwCDAL0AJQC7AJkAuQCmALYAt
wC0ANEAsgDwALAA/wCvAP8AqgD/AKYA/wD/AAAA/wAAPMAAADrAAAA5AAFANsADgDVAB
gA0wAhANEAKQDMADIAXwA5AMQAQADBAEYAvwBMAL0AUQC7AFUAuQBaALgAXwC2AGM
AtQBpALMAbwCyAHUAsAB9AK4AhwCsAJMAqwCgAKkAsACnAMcApQDoAKQA/wCjAP8AoAD/
AJwA/wD+AAAA8QAAAOIAAADXAAAAzAABAMQACgDBABQAvwAdAL0AJQC6AC0AtgA0ALM
AOwCxAEERwBHAK0ATACrAFEAqQBVAkGAWgCmAF4ApABkAKMAagChAHEAoAB4AJ4Agg
CdAI4AmwCbAJAqgCYAL8AlwDeAJYA+ACVAP8AIAD/AJEA/wDyAAAA3AAAAM0AADDBAAAA
twAAALIABwCvABAArAAYAKsAIQCqACgApgAwAKIANgCfAD0AnQBCAJwARwCaAEwAmQBQ
AJgAVQCXAFoAlgBfAJQAZQCTAGsAkQBzAJAAfACOAlgAJQCVAIsApACKALYAIdASAIcA8QC
GAP8AhgD/AIUA/wDgAAAAyAAAAALoAAACtAAAApwAAAKIAAwCfAAwAnAAUAJkAHACYACQAI
gArAJMAMQCRADcAkAA9AI4AQgCNAEYAiWBLAIoATwCJAFQAIABZAIYAXwCFAGYAgwBuAll
AdwCAAIIfwCPAH0AngB8AK8AewDHAHkA5wB5AP0AeAD/AHgA/wDHAAAAtwAAAKoAAACfA
AAAmAAAAJMAAACPAAGAJAARAIoAGACJAB8AiAAmAIYALACEADIAggA3AIAAPAB/AEEAfQB
GAHwASgB7AE8AegBUAHkAWgB3AGEAdgBoAHQAacgBzAHwAcQCJAHAAMABuAKkAbQC+A
GwA3QBrAPUAawD/AGsA/wC5AAAAqQAAAJsAAACSAAAAIwAAAIYAAACBAAUAfgAMAHwAF
AB7ABsAegAhAHkAJwB3AC0AdQAYAHMANwByADwAcQBAAHAARQBUEoAbQBPAGwAVQB
rAFwAaQBjAGgAbQBMAHcAZQCEAGMAkwBiAKMAYQC2AGAA0ABfAO0AXgD+AF4A/wCtAAA
AngAAAJEAAACHAAAAfWAAAHkAAAB1AAEAacgAIAHAAEABuABYAbQAdAGwAlgBrACgAaQAt
AGcAMgBmAdcAZAA7AGMAQABiAEUAYQBKAF8AUABeAFcAXQBfAFsAaABaAHMAWAB/AFc
AjgBWAJ4AVQCwAFQAXwBUAOQAUwD3AFMA/wCkAAAAIAAAAIcAAAB9AAAAAdAAAAAG8AAA
BrAAAAZwAFAGUACwBjABIAYgAYAGAAHgBfACMAXgAoAFwALQBbADIAWgA2AFkAOwBXAE
AAVgBGAFUATABUAFMAUgBbAFEAZABQAG8ATgB7AE0AigBMAJkASwCrAEoAvwBJANsASQ
DwAEkA/QCcAAAAjAAAAH8AAABzAAAAawAAAGUAAABhAAAAAXgABAFwACABaAA4AWAAU
AFcAGgBWAB8AVAAkAFMAKQBRAC4AUAAyAE8ANwBOADwATABBAEsASABKAE8ASQBXA
EcAYABGAGsARQB4AEQAhgBDAJYAQgCmAAEAuQBAANIAQADpAD8A+ACVAAAAhQAAAHg
EAABsBgAAYwcAAF0GAABZBQAAVglAAAFQABABRAAoAUAAQAE4AFgBNABsATAAgAEoAJQ
BJACKARwAuAEYAMwBFADgARAA9AEMARABBAEsAQABTAD8AXQA+AGgAPQB1ADsAgwA

6AJIAOQCjADgAtQA4AMsANwDjADcA8gCPAQAAfwYAAHEJAABmCwAAXQwAAFYMAABRDA
AATgoAAEwHAABKAwYASQALAEcAEQBFABcARAAcAEIAIQBBACYAQAAqAD4ALwA9ADQAP
AA6ADsAQQA6AEgAOABRADcAWwA2AGYANQFyADMBgQAYAZAAMQGHADAASgAvAMcALw
DeAC4A7QCKBgAAegoAAGwNAABhEAAAWBEAAFAAABLEQAARxAAAEEUNAABDCgEAQgc
HAEAEDAA/AxMAPQIYADsCHQA6AyIAOAMnADcDLAA2BDEANQQ3ADMFPgAyBUYAMQZPA
DAHWWQAuB2UALQhyACwlgAArCZAAKQmhACglsQAnB8UAJwbbACYF7ACFCgAADg4AAGgRA
ABcFAAAUxUAAEwVAABHFQAAQhQAAD8SAAA9EAAAOw0CADoLCAA5CQ4ANwgUADUJGg
AzCR8AMgkKADAJKQAvCi4ALgo1AC0LPAArDEQAKgxOACkNWQAnDmUAJg5zACQPggAjD5I
Alg+jACAOTqAfDckAHgzeAB4M7ACBDQAAcRIAAGQVAABYFwAATxgAAEgYAABDGAAAPhcA
ADoWAAA3FQAANRMAADQRAwAyDwkAMQ4QAC8OFgAtDhsAKw8hACoPJgApECwAJxAzAC
YROgAIEkMAJBjNACITVwAhFGMAIBRxAB4VgAAfZEAHBSiABsUtAAZE8oAGRlhABkS7wB+
EAAAbRQAAGAXAABVGQAATBsAAEUbAAA/GwAAOhoAADYZAAAzGAAAMBcAAC4VAAAfA
UAKhQLACgUEgAmFBgAJRQeACMVIwAiFskAIRYwACAWOAAfF0AAHhdKAB0YVQAbGWEAG
hlvABkZfgAYGY8AFxmhABUYswAUGMgAExfgABQW8AB6EwAAahcAAF0aAABSHAAASR0AAE
IdAAA8HQAANx0AADMcAAAvGwAAKxsAACkaAAAmGQIAJBkIACIZDgAgGRUAHhkaAB0ZIAAc
GicAGxstABobNQAZHD4AGBxIABcdUwAWHWAaFR5tABQefQATHo4AER2gABAdsgAOHMcAD
hveAA4a7wB3FgAAZxkAAfocAABPHgAARh8AAD8gAAA5HwAANB8AAC8fAAArHgAAKB4AAC
QdAAAhHQAAHx4FABweCwAaHhIAGR8YABgfHgAXHyQAFiArABUgMwAUITSAEyFGABliUQAR
II4ADyJsAA4iewANlosACyKdAAohrgAJIMIACR/ZAAkf6gBzGAAAZBwAAfCfAABMIQAAQyEAAD
wiAAA2IgAAMSEAAcwhAAAolQAAJSEAAcEhAAAcIgAAGilCABcjCAAVJA8AEyQVABlkGwARJ
SIAECUoAA4IMAANJjkADCZDAAsmTgAKJ1oACSDnAAgndwAHJ4cABiaZAAUmqwAEJb8AAyTV
AAQj6ABvGwAAYB8AAfQhAABJlwAAQSQAADokAAA0JAAALiQAACojAAAmIwAAliMAAB4kAA
AZJgAAfICAAABMoBgAQKQwADSoTAAwqGQALKh8ACSoIAAgqLQAHKzUABis/AAUrSgAEK1YA
AyxkAAIscwABK4QAACuWAAAqqQAAKbwAACjRAAAAn5QBrHgAAXCEAAFAkAABGJQAAPiYAA
DcmAAAxJgAALCYAACcmAAAKJgAAICYAABsnAAAXKQAAEysAAA8tBQAKLwoABY8RAAUvFg
AEMBwAAZAJAAlwKgAAMDEAADA7AAAwRgAAMVIAADFgAAAXcAAAMIEAADCUAAAvpgAALr
oAAC3PAAAs4wBnIQAAWCQAAEwnAABDKAAA0ygAADQpAAAUKAAAKigAACYoAAAIKAAAH
SoAABGsAAAULgAAEC8AAAsyBQAGNAoAAjUOAAA1FAAANhoAADYgAAA2JgAANi4AADY3A
AA2QgAANK4AADZcAAA2bAAANn0AADWRAAA1pAAANLcAADLOAAAY4gBiJQAAVCgAAEkqA
ABAKgAAOCsAADlrAAAsKwAAKCoAACQrAAAEIQAAGi4AABUxAAAMQwAACzUAAAc3BAAC
OQkAADoNAAA7EgAAPBcAADwdAAA8lwAAPCsAADwzAAA8PgAAPEoAADxYAAA8aAAAPHk
AADuNAAA7oQAAOrUAADnLAAA44QBdKQAUCsAAEUtAAA8LQAANS4AAC8tAAArLQAAJi4A
ACAvAAAbMgAAfJQAABE2AAAMOQAABZsAAA19AwAAPwcAAEALAABBDwAAQhQAAEMZAA
BEHwAARCCAAEQvAABEOQAAREUAAENTAABDYwAAQ3UAAEKJAABBNgAAQLIAAD/JAAA/4
ABXLQAASy8AAEEwAAA5MAAAMzAAAC4wAAAoMQAAIjMAABw1AAAXOAAAEToAAAs9AAA
GPwAAAUEAAABEAQAARQUAAEcIAABIDAAASREAAEsWAABMGwAATCIAAEwrAABMNAAA
TEAAAEtOAABLXgAASnAAAEqEAABJmQAASK4AAEfGAABG3gBSMQAARzMAAD4zAAA3Mw
AAMjMAACs0AAAKNgAAHjkAABg8AAASPwAADEEAAAZEAAAAARwAAAEKAAABLAAAATQIAAE
4FAABQCQAAUQ0AAFMsaABVFwAAVh4AAFYmAABWLwAAVToAAFVIAABVWAAAVGoAAFN
+AABSIAAAUaoAAFDAAABP2gBMNgAAQzcAADs3AAA2NgAALzcAACc6AAAgPQAAGUAAABJ
DAAAMRgAABUKAAABMAAAATwAAAFIAAABUAAAAVQAAAFcCAABZBQAAWgkAAFWNAABe
EgAAYBgAAGEgAABhKQAAYDQAAGBCAABfUQAAX2MAAF53AABdjQAAW6QAAfQ6AABa0gB

HOgAAPzoAADo6AAAzOwAAKj0AACJBAAAaRAAAE0kAAAtMAAAFUAAAAFMAAABWAAAAW
AAAAFsAAABdAAAAAXwAAAGEAAABjAAAAZQQAAGcIAABpDAAAaxIAAG0ZAABulGAAbiwAAG
05AABtSQAAbFoAAGtuAABqhAAAaJsAAGewAABmxwBEPgAAPj4AADc+AAAtQQAAJEUABx
KAAAUTgAAC1MAAARXAAAAWgAAAF4AAABhAAAAZAAAAGYAAABpAAAAawAAAG0AAABu
AAAAcAAAAHICAAB1BgAAAdwsAAHoSAAB8GgAAfCUAAHwxAAB7QAAAE1EAAHplAAB5egAAAd
5IAAHanAAB1uwBDQgAAO0MAADFGAAAnSgAAHk8AABVVAAALWgAAAI8AAABjAAAAZgAAA
GoAAABtAAAAcAAAAHIAAAB1AAAAAdwAAAHkAAAB7AAAAfQAAAH8AAACCAAAAhAQAAIcKA
ACKEgAAjRwAAIwoAACMNgAAi0gAAIpbAACJcAAAh4cAAIacAACFrwBARwAANKsAACTPAAAg
VQAAFIaAAAxhAAADZwAAAGwAAABvAAAAcwAAAHcAAAB6AAAAfAAAAH8AAACCAAAAhAA
AAIYAAACIAAAAIwAAAI0AAACQAAAAkwAAAJYAAACZCAAAAnREAAJ8cAACeKgAAAnjoAAJ1NA
ACcYgAAmngAAJmNAACYoAA6UAAAL1UAACRbAAAZYgAADmgAAARvAAAAAdAAAAHoAAAB
9AAAAgQAAAIQAAACHAAAAiQAAAIwAAACOA AAAkQAAAJMAAACVAAAAIwAAAJJoAAACdAA
AAoAAAAKMAAACnAAAAqwcAAK8SAACwHwAAAsC8AAK9CAACTVwAArG0AAKUCAACqIAA0W
gAAKGEAAB1oAAASbwAABnYAAAB9AAAAgwAAAIgAAACKAAAAjQAAAJAAAACSAAAAIAAAA
JcAAACaAAAAAnAAAAJ4AAACHAAAAowAAAKYAAACpAAAArAAAAALAAAAC0AAAAuQAAAL8IA
ADFFQAAxCUAAMM3AADCTAAAwGIAAL52AAC9iAD/AAAA/wAAP4AAAD6AAAA8QAEAO0A
DgDqABgA6QAHAOUAKgDgADIA2wA6ANYAQQDRAEcAzQBNAMoAUgDHAFcAxABcAMIAAYAD
AAGUAvGqAL0AcAC7AHgAuQCAALcAigC1AJYAswCkALEAtgCwANIArWdyAK4A/wCqAP8ApA
D/AJ8A/wD+AAAA+QAAAPIAADpAAAA3QAAANkACgDVABQA0gAdANEAJQDIAC4AwgA2AL4
APQC7AEMAuABIALcATQC1AFIAswBXALIAWwCxAGAArWBIAK4AawCsAHIAqgB6AKkAhQCnA
JAApQCeAKMArWciAMgAoQDqAJ8A/wCdAP8AmAD/AJUA/wD2AAAA6wAAAN4AAADOAAAAx
gAAAMEABwC+ABAAugAZALcAIGC0ACkAsAAxAK0ANwCrAD4AqQBDACkASACmAE0ApABRA
KMAVgChAFsAoABgAJ4AZgCdAG0AmwB1AJkAfgCYAlOAlgCYAJUAqACTAL4AkgDgAJEA/ACQ
AP8AjAD/AlkA/wDnAAAA1gAAAMUAAAC5AAAAswAAAK4AAwCoAAwApgAVAKUAHQcKACQA
oAAsAJ0AMgCbADgAmQA+AJcAQwCWAEcAIABMAJMAUACSAFUAKABaAI8AYQCOAGcAjAB
vAlsaEACJAIQAhwCRAIUAOQCEALUAgwDTAIIA9ACBAP8AgAD/AH0A/wDSAAAAwAAAALEAA
ACoAAAAogAAAJsAAACXAAgAIQARAJMAGQCSACAAkQAnAI4ALQCMADMAigA4AIGAPQCHA
EIAhgBGAIQASwCDAFAAggBVAlAAWwB/AGEAfQBpAHwAcwB6AH0AeQCLAHcAmgB1AK0Ad
ADHAHMA6gBzAP8AcgD/AHEA/wC/AAAArwAAAKIAAACZAAAAkQAAAIaAAACIAAUhQAMAI
MAFACCABsAggAiAH8AKAB9AC4AegAzAHgAOAB3ADwAdgBBAHUARQB0AEoAcgBPAHEAV
QBwAFwAbgBkAG0AbQBrAHgAagCFAGgAIABnAKYAZgC9AGUA3wBIAPkAZQD/AGUA/wCwAA
AAoAAAAJQAAACLA AAAgwAAAH0AAAB5AAEAdgAJAHUAEABzABcAcgAdAHEAIwBvACgAbQ
AuAGwAMgBqADcAaQA7AGgAQABnAEUAZgBKAGQAUABjAFcAYQBfAGAAaABfAHMAXQB/A
FwAigBaAKAAWQC0AFgA0gBYAPEAWAD/AFgA/wCkAAAAIQAAAIkAAAB+AAAAdgAAAAHAAAA
BtAAAAagAFAGcADABmABMAZQAYAGQAHgBjACQAYQApAGAALQBeADIAXQA2AFwAOwBb
AEAAWgBFAFGASwBXAFIAVgBaAFQAYwBTAG4AUQB6AFAAiQBPAJsATgCuAE0AxwBNAOc
ATQD8AE0A/wCaAAAAjAAAAH4AAABzAAAAawAAAGYAAABhAAAAAXgACAFwACABbAA4AW
QAUAFkAGgBYAB8AVwAkAFUAKQBUAC0AUgAxAFEANgBQADsATwBAAE4ARgBMAE0ASwB
VAEOAXgBIAGkARwB1AEYAhABFAJYARACoAEMAvgBDAN0AQwD0AEMA/wCTAAAAgwAAAH
UAAABqAAAAAYgAAAFwAAABYAAAAVQAAAFIABABRAAoATwAQAE4AFQBOABsATQAgAEsA
JABKACkASAAAtAEcAMQBGADYARQA8AEMAQgBCAEkAQQBRAEAAWgA/AGUAPgBxAD0AgA
A8AJEAOWCjADoAtwA6ANEAOgDsADoA+wCMAAAAFAAAAAG4AAABjAgAAWgMAAFMDAABPA

gAATAAAAEoAAQBIAAcARwAMAEUAEQBEABcAQwAbAEIAIABBACQAQAApAD4ALQA9ADIA
PAA4ADsAPgA6AEUAOQBNADcAVwA2AGIANQBuaDQAFaAZAI0AMgCfADIA sQA xAMkAMQDk
ADEA9QCGAAAAAdQEAAgGFAABdBwAAVAKAAE0JAABICAAARQYAAEIEAABBAQMAPwAIAD
4ADQA9ABMAOWAXADoAHAA5ACEAOAAIADYAKgA1AC4ANAA0ADMAOgAyAEIAMQBKAC8A
VAAuAF8ALQBrACwAegArAlOAKwCbACoArQApAMIAKQDcACkA7gCBAQAACAUAAAGMJAA BX
DAAATg0AAEGNAABDDQAAPwsAADwKAAA6CAA AOAFADcCCgA2AA4ANAAUADMAGAAyA
B0AMAAiAC8AJgAuACsALAAxACsANwAqAD8AKQBHACgAUQAnAVwAJgJpACUCeAAkAogAI
wKZACMBqgAiAL0AIQDVACEA6QB8BQAAbAKAAAF4MAABTEAAAShEAAEMRAAA+EQAAOH A
AADYOOAAZDQAAMQsCADAIBgAvBgsALgQQACwEFQArAxoAKQMeACgElwAnBCgAJgUuAC
UFNAAkBjwAlwZFACIHUAAhCFsAHwhoAB4JdwAdCYcAHAmYABslqQAbB7sAGgXRABoE5AB4
CAAAaAwAAFoQAABPEgAARxQAAEAUAAA6FAAANRMAADESAAAuEQAALBAAACoOAwApD
AcAKAoLACYJEQAICRYAlwkbACIJIAAhCiYAlAosAB8LMwAeCzsAHQxFABsNUAAaDVwAGQ5q
ABcOeQAWDooAFQ6bABQOrAATDb4AEwvTABIK5QBzCwAAZBAAAFcTAABMFQAAQxYAAD0
WAAA3FgAAMhYAAAC0VAAAqFAAAJxMAACUSAQAJEQQAIG8HACAODAAfDhMAHQ4YABwOH
gAbECQAGhAqABkRMgAYEToAFhJEABUSTwAUE1sAExNpABIUdwAQFIgADxSaAA0TqwAMEr
4ACxHUAAsQ5wBwDgAAYRIA AFQVAABJFwAAQRgAADoZAAA0GQAALhgAACoYAAAmFwAAI
xYAACEWAAAeFQIAHBQEABoUCQAYFA4AFxQVABYUGwAVFSEAFBUoABMWLwASFJgAERd
CABAXTQAOGFKADRhmAAsZdAAKGYQACRiWAAgYqAAHF7oABhbPAAYV5ABsEQAAXRUA A
FEYAABGGQAAPhoAADcbAAAxGwAAKxoAACcaAAAjGQAAlBkAAB0YAAAbGAEAGBgCABUZ
BgATGQwAERoSABAaGAAOGh8ADhoIAA0bLAAMGzQACxw+AAocSAAIHFAQBx1iAAYdcAAF
HYEABB2TAAMcpQACG7cAARrMAAEZ4QBpFAAAWWhcAAE4aAABEGwAAOxwAADQdAAAuHQ
AAKR0AACUCAAAhHAAAHhsAABsbAAAYGwAAFBwBABEEBAAOHgoACx8QAAofFgAJHxwAC
B8iAAcgKQAGIDEABSA6AAQhRQADIVEAASFeAAAhbQAAIX4AACGQAAAgowAAH7UAAB7KA
AAD3wBIFgAAVxoAAEscAABBHgAAOR8AADEfAAArHwAAJx8AACleAAAfHgAAHB0AABkdAAA
WHgAAEH8BAA4hBAAKlwKABYQOAAUkFAADJBkAASQfAAAIJgAAJS4AACU3AAAIQQA AJk4A
ACZbAAAmagAAJnsAACWOOAAAlOQAAJLQAACLJAAAI3wBhGQAAUx0AAECfAAA+IAAANI EAA
C8hAAApIQAAJCAAACEgAAAEIAAGh8AABcgAAATlgAADiMBAAslBAAHJwgAAygMAAApEQA
AKhcAACodAAAqlwAAKisAACozAAArPgAAK0oAACtYAAArZwAAK3kAACqMAAApNWAAKbIAAC
fIAAAm3wBdHAAATyAAAEQhAAA7lwAAMyMAACWjAAAnlwAAlyIAACAiAAAcIlgAAGCMAABQk
AAQJgAACygAAAcqAwADLAcAAC0LAAAvDwAAMBQAADAaAAAwIAAAMCgAADEwAAAxOgA
AMUCaADBUA AAwZAAAMHUAAc+JAAAvnQAALrEAAC3HAAAs3gBYIAAASyMAAEAKAAA4JQ
AAMCUAAcOIAAAIJQAAlIQAAB4kAAAZJgAAFScAABApAAAMKwAABY0AAAMwAwAAMgYAAD
MJAAA1DQAANhIAADcXAAA3HQAANYQAADctAAA3NwAAN0MAADdQAAA3YAAANnIAADaGA
AA1mwAANK8AADPGAAAY3gBTJAAARyYAAD0nAAA0KAAALSGAACgoAAAKJwAAICcAABSpA
AAWKgAAES0AAAwvAAAHMQAAAzQAAAA2AQAAOAQAADkHAAA7CwAAPQ8AAD4UAAA/Gg
AAPyEAAD8pAAA/MgAAPz4AAD5MAAA+WwAAPW0AADyBAAA8lwAAOqwAADrEAAA53gBOK
AAQYkAADkqAAAXKwAAKyoAACcqAAAIKgAAHSwAABcuAAASMAAADDMAAAc2AAACOOAAA
ADsAAAA9AAAAPwIAAEAFABCCAAARAwAAEURAABHFgAASB0AAECIAABHLgAARzkAAEZ
HAABGVgAARWgAAER8AABEkgAAQqgAAEHAAABA2wBJLAAAPi0AADYtAAAvLQAAKy0AAC
UtAAAfLwAAGDIAABM1AAAMOAAABzsAAAE9AAAAQAAAAEIAAABEAAAARgAAAEgCAABKB
QAATAgAAE4MAABQEgAAUhgAAFIgAABSKAAAUTMAAFFAAABQUAAAT2IAAE52AABNjAAA
TKMAAEq6AABJ1QBEMAAAOZEAADMxAAAuMAAAKDAAACEzAAAaNgaAEzkAAA08AAAGQA

AAAEMAAABGAAAASQAAAEsAAABNAAAATwAAAFEAAABTAQAAVQQAAFcIAABZDQAAXBM
AAF0aABbdlwAAXS0AAFW6AABbSQAAWIsAAFlvAABYhQAAV5wAAFWzAABUzQA/NAAAODQ
AADIOAAAsNAAAJDcAABw6AAAUPgAADUIAAAZGAAAASQAAAE0AAABPAAAAUgAAAFUAAA
BXAAAAWQAAAFwAAABeAAAAAYAAAAGIDAABkCAAAZw0AAGkUAABrHAAaaiYAAAGoyAABpQ
QAAaFMAAGZmAABIfAAAA5QAAGKrAABGwgA8OAAANzgAADA4AAAnOwAAHj8AABZDAAAN
SAAABkwAAABQAAAAVAAAAFCAAABbAAAAXQAAAGAAAAABjAAAAZQAAAGcAAABpAAAAaw
AAAG0AAABwAgAAcwcAAHYNAAB5FQAAeR8AAHkrAAB4OQAAAd0oAAHZdAAB1cgAAAdlgAAH
KgAABwtAA7PAAANDwAACo/AAAhRAAAF0kAAA5OAAAFVAAAAFgAAABcAAAAAYAAAAGQAA
ABoAAAAagAAAG0AAABwAAAAcgAAAHUAAAB3AAAAeQAAAHwAAAB+AAAAgQAAAIQFAACI
DAAAIxUAAIshAACKLwAAiT8AAIdTAACGZwAAhH4AAIOVAACBqQA5QQAALkQAACRJAAAaT
gAAEFUAAAZaAAAAAYQAAAGUAAABqAAAAbgAAAHEAAAB1AAAAeAAAAHsAAAB+AAAAgAA
AAIIAAACFAAAAhwAAAIkAAACMAAAAJwAAAJMAAACWAwAAmgwAAJ0YAAACcJQAAMzUAAJ
pIAACYXQAAI3IAAJWIAACUnAAzSQAAKE4AAB1UAAATWwAAB2IAAABpAAAAbwAAAHMAAA
B4AAAAfAAAAH8AAACCAAAAhQAAAIgAAACLA AAAJgAAAJAAAACSAAAAIQAAAJgAAACbAA
AAngAAAKIAAACmAAAAqwEAAK8LAACyGAAAsCgAAK85AACTgAArGMAAKp4AACpiwAsUw
AAIVoAABZhAAAKaQAAAHAAAAAB3AAAAfQAAAIeAAACGAAAAigAAAI0AAACPAAAAkgAAJU
AAACYAAAAmwAAAJ0AAACgAAAAogAAAKUAAACoAAAArAAAAALAAAAC1AAAAugAAAMABA
ADIDQAAxx0AAMYvAADEQwAAwlgAAMBsAAC+fwD+AAAA+gAAAPgAAAD2AAAA8gAAAO4AC
gDsABQA5AAeAOEAJwDbAC8A1QA3ANEAPgDNAEQAyQBKAMYATwDEAFQAwwgBYAMAAXQ
C+AGIAvABoALoAbgC5AHUAtwB+ALQAIACyAJQAsACjAK4AtgCsANMAqwD2AKoA/wChAP8A
mwD/AJYA/wD3AAAA8gAAAO0AAADkAAAA3QAAANcABgDOABEAzAAaAMsAlwDFACsAvwAy
ALsAOQC4AD8AtgBFALQASgCyAE8AsABTAK8AWACTAF0ArABiAKoAaACpAG8ApwB4AKUAg
gCjAl4AogCcAKAAArgCeAMGAnADtAJsA/wCVAP8AkAD/AlwA/wDuAAAA6AAAAANUAAADLAAAA
xAAAAALsABAC4AA0AtQAWALMAHgCxACYArQAtAKoANACnADoApQA/AKMARACiAEkAoABO
AJ4AUgCcAFcAmwBcAJkAYgCXAGkAlQBxAJQAFACSAIgAkQCWAJAAPwCOAL4AjQDiAlwA/wC
JAP8AhAD/AIAA/wDiAAAAzAAAAAL8AAAC1AAAArAAAAKcAAACjAAkAoQASAKAAGQCgACEA
mwAoAJcALgCUADQAKgA6AJEAPwCPAEMAjgBIAIwATQCLAFEigBXAlgAXQCHAGQAqBrAl
QAdQCCAIEAgQCPAH8AnwB9ALQAFADVAHwA+AB7AP8AdwD/AHUA/wDLAAAAtwAAAKwAAA
CiAAAAmgAAAJUAAACRAAQAJwAMAlwAFQCLABwAigAjAlcAKQCFAC8AgwA0AIEAOQCAAD0
AfwBCAH0ARwB8AEwAewBRAHkAVwB4AF4AdgBmAHUABwBzAHoAcgCIAHAAMABvAKsAbQ
DHAGwA7gBsAP8AagD/AGgA/wC2AAAApWAAAjWAAACRAAAAigAAAIQAAACAAAEafQAJAHs
AEAB7ABcAegAeAHgAJAB2ACkAdAAuAHMAMwBxADgAcAA8AG8AQQBtAEYAbABLAGsAUQ
BpAFgAaABgAGYAaQBIAHQAYwCBAGIAkgBhAKQAXwC8AF8A4gBeAP8AXgD/AFwA/wCnAAA
AmQAAAlwAAACCAAAAewAAAHUAAABwAAAAbgAFAGwADABrABMAawAZAGsAHwBpACQA
ZwApAGUALgBkADIAYwA2AGEAOwBgAEAAxwBGAF0ATABcAFMAWgBaAFkAZABYAG8AVg
B7AFUIwBUAJ0AUwCzAFIA1QBSAPcAUgD/AFEA/wCcAAAAjgAAAIAAAAB1AAAAbgAAAGcA
AABjAAAAAYQABAF8ACABeAA4AXgAUAF0AGgBcAB8AWgAkAFgAKQBxAC0AVgAxAFUANgB
TADsAUgBAAFEARgBP AE0ATgBVAE0AXwBLAGoASgB2AEkAhgBIAJcARwCsAEcAyQBGAO0
ARwD/AEcA/wCTAAAAGgAAAHUAAABqAAAAAYgAAAFwAAABYAAAAVQAAAFQABABSAAoAU
QAQAFEFQBQABsATwAfAE0AJABMACgASwAtAEoAMQBIADYARwA7AEYAQQBF AEgAQwB
RAEIAWgBBAGUAQABxAD8AgAA+AJIAPQCmAD0AvwA9AOIAPQD7AD0A/wCKAAAAegAAAG
wAAABhAAAAAWAAAAFIAAABOAAAAASwAAAEkAAQBIAAYARwALAEYAEQBGABYARQAbAEQ

AIABCACQAQQAoAEAALAA/ADEAPQA2ADwAPAA7AEQAQOgBMADkAVgA3AGAANgBtADUafA
A1AI0ANACgADMAtwAzANYANADzADQA/wCCAAAAcgAAAGUAAABZAAAAUQAAAEoAAABG
AAAAQwAAAEAAAAA/AAMAPgAIAD0ADQA8ABIAPAAxADsAGwA5ACAAOAAkADcAKAA2AC0
ANAAyADMAOAAyAD8AMQBIADAAUQAvAFwALgBpAC0AdwAsAlgAKwCcACsAsAArAMsAKw
DqACsA+wB7AAAAbAAAAF4BAABTAwAASwQAAEQFAAA/BAAAOwMAADkBAAA3AAEANGAF
ADUACQA0AA4AMwATADIAFwAxABwALwAgAC4AJAAtACkALAAuACsANAAqADsAKQBEACg
ATgAnAFkAJgBIACUAdAAkAIQAJACXACQAqwAjAMIALwDgACQA9AB2AAAAZwIAAFkFAABOB
wAARgkAAD8JAAA6CQAANggAADIHAAAwbQAALgIDAC0ABgAsAAoAKwAPACoAEwApABgAK
AAcACcAIQAmACUAIQArACQAMQAJADgAlgBBACEASwAgAFYAHwBjAB4AcQAeAIEAHQCUA
B0ApwAdALsAHQDWAB0A7QByAAAAyGUAAFUJAABKcWAAQgwAADsNAAA1DAAAMAwAAC
0LAAAqCQAAKAkBACyGBAAIBAcAJAILACMBEAAjARQAIQEZACAAHQAFACIAHgAoAB0ALgA
dATUAHAE+ABsBSAAaAVMAGQJgABgCbWAXAn8AFwGRABYBowAWALYAFgDNABYA5QBuA
wAAXggAAFEMAABHDgAAPg8AADcQAAAxEAAALA8AACgOAAAIIDQAAIwwAACELAwAfCQUA
HggIAB4GDAACbHEAGwUVABoFGgAZBR8AGAUIBgFKwAXBjMAFgY8ABUHRgAUB1IAEwhfA
BllbgARCH4AEAiPABAHoQAPBrMADwTIAA8C3wBqBgAAWwsAAE4OAAABDEQAAOxIAADQSA
AAuEgAAKRIAACURAAAhEAAAHxAAABwOAgAaDQQAGQwHABgLCQAWCg0AFQoSABQKFw
ATCh0AEwojABILKQARCzEAEAw7AA4MRgANDVMACw5gAAoObgAJDn0ACQ6PAAgOoQAHD
bMABwzHAAcK2wBmCQAAVw0AAEsRAABBEwAAOBQAADEUAAARFAAAJhQAACITAAAEwA
AGxIAABkSAGAwEEQQAFBAGABMQCAARDwoADg4OAA4PFQANEBsADBAhAAsRKAakes8A
CRI4AAgSQwAHE08ABhNcAAUTagAEE3oAAxOMAAMTnwACErEAAhHFAAEQ2wBjDAAAVBA
AAEGTAAA+FQAANRYAAC4WAAAOFGAAIxYAAB8VAAAcFQAAGRQAABYUAgAUewQAEhMG
ABATCAAMEwkACHQMAAKUEgAIFRgABxUeAAYVJAFFFiwABBY1AAMXPwACF0sAARdZAAA
YaAAAGHgAABeKAAAXnQAAFrAAABXEAAAU2gBfDgAAURMAAEUVAAA7FwAAMxgAACwYA
AAmGAAAIRgAAB0XAAAFwAAFXYAABQVAwASFQUAEBUGAA0WBgAJFwgABxgLAAQZEAA
DGhUAAhobAAEalgAAGikAABsyAAAbPAAAG0gAABxWAAAcZQAAHHYAABuJAAAbnAAAGq8
AABnDAAAY2wBcEgAAThUAAEIXAAA4GQAAMBoAACkaAAAKGgAAHxoAABsZAAAYGAAAFh
gBABQXBAARFwQADRgEAAoaBQAHGwcAAx0KAAAEdgAAHxMAAB8ZAAAfHwAAHycAACAvA
AAgOQAAIEYAACBTAAAgYwAAIHQAACCHAAAFmwAAHq4AAB3DAAAc2wBYFQAAShgAAD8a
AAA1GwAALRwAACccAAAhHAAAHRSAABoaAAAXGgAAFRkCABlaAgAOGwIACxwCAAceBAA
EIAYAACEJAAAJDAAAJREAACUXAAAIHQAAJSQAACUsAAAINgAAJUMAACVQAAAIYAAAJXE
AACSFAAAjmQAAI60AACLEAAAh3ABUGAAARxsAADwcAAAYHgAAKh4AACQeAAAgHgAAHB0
AABkcAAAWGwAAExwAAA8dAAALHwAABYEBAAQjAgAAJQUAACcHAAAPcGAAKq4AACsUAA
ArGgAAKyEAACspAAArMwAAKz8AACtNAAArXAAAKm4AACqCAAAPlwAAKKwAACfDAAAm3Q
BPGwAAQx4AADgfAAAvIAAAKCAAACMGAAAEHwAAHB4AABgeAAAUHwAAECAAAASiAAAHJ
AAAAyYAAAAoAQAAKqMAACwFAAAuCAAAMQwAADMRAAAzGAAAMx4AADMmAAAYMAAAM
jsAADJJAAAXWAAAMWoAADB+AAAvlQAALqoAAC3CAAAAs3gBKHWAApYEADUiAAASlwAAJil
AACEiAAAEIQAAGiEAABUiAAARJAAADCYAAAcOAAADKqAAAC0AAAAvAAAAMQEAAADMDAA
A2BgAAOAkAADoOAAA7FAAAOxsAADsiAAA7KwAAOjcAADpEAAA5VAAAOGYAADD6AAA2kQ
AANagAADTAAAAz3QBFIgAAOIQAADeIAAAqJQAAJCQAACEjAAAcIwAAFyUAABlnAAAMKgAA
BywAAAlvAAAAMgAAADQAAAAA2AAAAOQAAADsBAAA9AwAAPwcAAEELAABDEAAARbCAAE
QeAABDJwAAQzIAAEI/AABCTwAAQWAAAD91AAA+jAAPaMAADy8AAA62gBAJgAANigAAC4
oAAAOJwAAJCYAAB8mAAAZKAAAEysAAA0uAAAHMQAAAJQAAAA3AAAAOQAAADwAAAA+A

AAAQAAAAEMAAABFAQAARwQAAEkHAABMDAAAThIAAE4aAABOIgAATS0AAE05AABMSQA
AS1oAAEluAABlhQAARp4AAEW2AABD0gA8KgAAMisAACwrAAAnKgAALioAABssAAAVLwAADjl
AAAc2AAABOQAAAD0AAABAAAAAQgAAAEUAAABHAAAAASQAAAEwAAABOAAAAUAAAAFIE
AABVCAAAWA0AAAFoVAABaHQAAWScAAFGzAABYQgAAVIMAAAFVmAABTfQAAUpUAAFCtAA
BPyAA3LgAAMC4AACstAAAmLQAAHjAAABY0AAAONwAACDsAAAA/AAAAQwAAAEYAAABJA
AAATAAAAE8AAABSAAAAVAAAAFcAAABZAAAAWwAAAF4AAABhAgAAYwcAAGcNAABoFgA
AaCAAAGcrAABmOQAAZUsAAGNeAABhdAAAX4wAAAF2kAABcvAA0MgAALzEAACkxAAAhNAA
AGDgAABA9AAAIQQAAAEYAAABKAAAATgAAAFIAABVAAAAWAAAAAFsAAABeAAAAAYAAAA
GMAAABIAAAAZwAAAGoAAABtAAAAcAIAAHMHAAB3DwAAeBkAAHckAAB2MQAAAdUIAAHNV
AABxagAAb4EAAG2ZAABrrwA0NQAALTYAACQ5AAAbPQAAEUMAAAhIAAAATQAAAFIAAB
WAAAAWwAAAF4AAABiAAAAZQAAAGgAAABrAAAAbgAAAHAAAAABzAAAAAdQAAAHgAAAB6A
AAafgAAAIeAAACFBwAAiRAAAIkCAACIKQAAhzkAAIVMAACDYAAAgXYAAICMAAB+oQAyOgA
AKD4AAB1CAAAUSAAACU4AAABVAAAAWgAAAF8AAABkAAAAaAAAAAgwAAABwAAAAAdAAA
AHcAAAB6AAAAfQAAAH8AAACCAAAAAhAAAAIcAAACKAAAAjQAAAJEAAACVAAAAmgYAAJ4
RAACdHgAAmy0AAJIAAACXVAAAIWkAAJN/AACRIAAAsQgAAIUgAABdOAAAMVQAAAVwAAABj
AAAAaAAAAAG4AAABzAAAAdwAAAHsAAAB+AAAAaggAAAIUAAACIAAAAIwAAAI4AAACQAAAA
kwAAAJYAAACZAAAAAnAAAAKAAAAACIAAAaqQAAAK4HAACxFAAAAsCMAAK41AACsSQAAql4A
AKhzAACmhwAITQAAGIMAAA9bAAADYwAAAGoAAABYAAAAAdwAAAHwAAACBAAAAhQAAAIk
AAACMAAAAKAAAAJMAAACWAAAAmQAAAJwAAACeAAAAoQAAAKQAAACoAAAArAAAAALAA
AAC1AAAAvAAAAAMQAAADNBgAAzRYAAMonAADIOgAAxU4AAMJjAAC/dgAAAAAAAAAAAAAA
AAAAAAAAAAAAQMEBQYICQoLDQ4PERITFBYXGBobHB0fICEiJCUMKCKqKy0uLzAyMzQ2Nzg5
Ozw9PkBBQkRFRkdJSktNTk9QUINUUVVdYVWVtcXV5gYWJjZWZnaWprbG5vcHFzdHV3eHl6fH1+
glGCg4WGh4iKi4yOj5CRk5SVlpiZmpydnP+hoqOkpqeoqqusra+wsbO0tba4ubq7vb6/wcLDxMbHy
MnLzM3P0NHS1NXW19na293e3+Di4+Tm5+jp6+zt7vDx8vT19vf5+vv8/v/////////
/////////AAAAAAAAAAAAAAAAAAAAAEDBAUGCAkKCw0ODxESExQWFxgaGxwdHyAhliQlJig
pKistLi8wMjM0Njc4OTs8PT5AQUJERUZHSupLTU5PUFJTfVFXWFlbXF1eYGFY2VmZ2lqa2xu
b3Bxc3R1d3h5enx9foCBgoOFhoeliouMjo+QkZOUIZaYmZqcnZ6foaKjpKanqKqrrK2vsLGztLW2uL
m6u72+v8HCw8TGx8jJy8zNz9DR0tTV1tfZ2tvd3t/g4uPk5ufo6evs7e7w8fL09fb3+fr7/P7/////////
/////////wAAAAAAAAAAAAAAAAAAAAAABAwQFBggJCgsNDg8REhMUFhcYGHsc
HR8glSlkJSYoKSorLS4vMDIzNDY3ODk7PD0+QEFCREVGR0IKS01OT1BSU1RVV1hZW1xdXm
BhYmNiZmdpamtsbm9wcXN0dXd4eXp8fX6AgYKDhYaHilqLjI6PkJGTIJWWmJmanJ2en6Gio6Sm
p6iqq6ytr7Cxs7S1tri5uru9vr/BwsPEXsflcyMzc/Q0dLU1dbX2drb3d7f4OLj5Obn6Onr7O3u8PHY9P
X29/n6+/z+/////////8AAQIDBAUGBwgJCgsMDQ4PEBESExQVFhcYGR
obHB0eHyAhliMkJSYnKCKqKywtLi8wMTIzNDU2Nzg5Ojs8PT4/QEFCQ0RFRkdISUpLTE1OT1BR
UINUUVVZXWFlaW1xdXl9gYWJjZGVmZ2hpamtsbW5vcHFyc3R1dnd4eXp7fH1+f4CBgoOEhYaHil
mKi4yNjo+QkZKTlJWWl5iZmpucnZ6foKGio6SlpqeoqaqrrK2ur7CxsR00tba3uLm6u7y9vr/AwcLDx
MXGx8jJysvMzc7P0NHS09TV1tfY2drb3N3e3+Dh4uPk5ebn6Onq6+zt7u/w8fLz9PX29/j5+vv8/f7/b
WZ0MQAAAAADBCEAAAEAAAAAAAAAAAAAAAAAAAAAABAAAAAAAAAAAAAAAAAAAAAAQAAA
AECAwQFBgcICQoLDA0ODxAREhMUFYXGBkaGxwdHh4fICEilyQlJicoKSorLC0uLzAxMjM0NT
Y3ODk6Ozw9Pj9AQUJDREVGR0hJSktMTU5PUFFSU1RVVldYWVWpbXf1eX2BhYmNkZWZnaGl
qa2xtbm9wcXJzdHV2d3h5ent8fX5/glGCg4SFhoeliYqLjI2Oj5CRkpOUIZaXmJmam5ydnP+goaKjp
KWmp6ipqqusra6vsLGys7S1tre4ubq7vL2+v8DBwsPExcbHyMnKy83Oz9DR0tPU1dbX2Nna29zd

[illegible]

P8v62j/L+to/y/raP8v62j/L+to/y/raP8v62j/L+to/y/raP848whAd7SLA7b2DUo2to9TtzYRXnf0kum071
h1MKrkPbCrZj3wa2e+MGuoPnBrqL5wK6j+sCuo/vAraP7v62j/L+to/y/raP8v62j/L+to/y/raP8v62j/L+
o/y/raP8v62j/L+to/y/raP8v62j/L+to/y/raP8v62j/L+to/y/raP8480gAd7TKw7a2TQo2ds7T9vZQ3re00i
n0rxg1cKrj/bCrZj3wa6d+MGuoPnBrqL5wK6j+sCuo/rAraP7v62j/L+to/y/raP8v62j/L+to/y/raP8v62j/L
+to/y/raP8v62j/L+to/y/raP8v62j/L+to/y/raP8v62j/L+to/y/raP84s4fAd3UKg3Z2jlp2Ns8T9nbQXvd1E
ap0rxf1sKrjvbCrZj3wa6d+MGuoPnBrqL5wK6j+sCuo/q/rqP7v62j/L+to/y/raP8v62j/L+to/y/raP8v62j/
L+to/y/raP8v62j/L+to/y/raP8v62j/L+to/y/raP8v62j/L+to/y/raP84s4eAdzVKA3Y2jEp2Nw8UNncQHzc
1kOq0bxe2MKrjvbCrZj3wa6d+MGuoPnBrqL5wK6j+sCuo/q/rqP7v62j/L+to/y/raP8v62j/L+to/y/raP8v
62j/L+to/y/raP8v62j/L+to/y/raP8v62j/L+to/y/raP8v62j/L+to/y/raP84c8dANzWJw3Y2zlq1908UdjCQH
7b2ECr0bxd2cKrjfbCrZf3wa6d+MGuoPnBrqL5wK6i+sCuo/q/rqP7v62j+7+to/y/raP8v62j/L+to/y/raP
8v62j/L+to/y/raP8v62j/L+to/y/raP8v62j/L+to/y/raP8v62j/L+to/y/raP84NAbANvXJQ3X3Dlr1t48UtbD
QH/Z2j2s0Lxc2sKrjPbCrZf3wa6d+MGvn/nBr6H5wK6i+sCuo/q/rqP7v66j+7+to/y/raP8v62j/L+to/y/ra
P8v62j/L+to/y/raP8v62j/L+to/y/raP8v62j/L+to/y/raP8v62j/L+to/y/raP839EZANnYlw7W3TIs1d48U9
beQIDY2j2tz71a28KrivbCrZb3wa6b+MGvn/nAr6H5wK6i+sCuo/q/rqP7v66j+7+to/y/raP8v62j/L+to/y/
/raP8v62j/L+to/y/raP8v62j/L+to/y/raP8v62j/L+to/y/raP8v62j/L+to/y/raP83tIWANjaIQ7W3jlt1d88Vd
XfQILV2T6vzr1X3cKsifbCrpT3wa+a+MCvnnAr6H5wK6i+r+uo/q/rqP7v66j+7+uo/u/rqP7v66j+7+u
o/u/rqP7v66j+7+uo/u/rqP7v66j+7+uo/u/rqP7v66j+7+uo/u/rqP7v66j+7+uo/u/rqP73NQTANfbIQ/V3z
Mu1OE8VtPhQITU2ECxZr1U3sKshfbBro/3wa+Y+MCvnnAr6H5v66i+r+uovq/rqP7v66j+7+uo/u/rq
P7v66j+7+uo/u/rqP7v66j+7+uo/u/rqP7v66j+7+uo/u/rqP7v66j+7+uo/u/rqP7v66j+7+uo/u/rqP72tYO
ANbclhDU4DMv0ul9WNLiQlBr1kKzzL1W4MKsdffBr4H3wK+Y+L+vnvm/r6H5v66i+r+uovq/rqP7v6
6j+7+uo/u/rqP7v66j+7+uo/u/rqP7v66j+7+uo/u/rqP7v66j+7+uo/u/rqP7v66j+7+uo/u/rqP7v66j+7+uo/
u/rqP72NkKAdXelxHS4jQx0eQ9W9DkQYjP1UW1yL9e3cKuUfe/sIP4v6+Z+L+vnvm/r6D5v66h+r+
uovq/rqL6v66j+7+uo/u/rqP7v66j+7+uo/u/rqP7v66j+7+uo/u/rqP7v66j+7+uo/u/rqP7v66j+7+uo/u/rqP
7v66j+7+uo/u/rqP71tsMAdPgJRPR5DQ0z+Y+Xs3mQozL1Em4xMRm07+4eOi9sln3vq+Z+L6vnn
m+r6D5vq6h+r6uovq+rqL6vq6j+76uo/u+rqP7vq6j+76uo/u+rqP7vq6j+76uo/u+rqP7vq6j+76uo/u+rq
P7vq6j+76uo/u+rqP7vq6j+76uo/u+rqP71N4PAdHiJhXP5zU3zOg/YsroQ5DG2EywwMtnx73Ce9i7u
4nju7iT6ry1me64tJfwtLOU8bCzkvKtspHzq7KR86mykvOpspLzqbKS86mykvOpspLzqbKS86mykv
OpspLzqbKS86mykvOpspLzqbKS86mykvOpspLzqbKS86mykvOpspLz0uETAs/mKRjM6jc7yetAZ8
XsR46/3lCnu9NmurnLeci4xYfSuMKQ2LO/j9ysvYrfpryH4aK7huOeu4bkm7qH5Jm6ieWZuonlmbqJ
5Zm6ieWZuonlmbqJ5Zm6ieWZuonlmbqJ5Zm6ieWZuonlmbqJ5Zm6ieWZuonlmbqJ5Zm6ieWZuon
lz+UXBMzpKxvJ7ThAxu9Dar/xToS65Fmbtttmq7TUd7izzoTBrMqDyKXIgc2gxODRm8WA05fEgNS
Tw4HVkcOD1o/ChtaPwobWj8KG1o/ChtaPwobWj8KG1o/ChtaPwobWj8KG1o/ChtaPwobWj8KG1o
/ChtaPwobWj8KG1o/ChtaPwobWzOkbBsjuLSDF8TtFv/ZIYrn3VHm07F+NseNpna3cc6mk13Kym9
NyuZTQcr6OznTBis11w4fMd8WFzHnGg8t8x4HLgMeBy4DHgcuAx4HLgMeBy4DHgcuAx4HLgMe
By4DHgcuAx4HLgMeBy4DHgcuAx4HLgMeBy4DHgcuAx4HLgMeBy4DHyO0gCcXyMCTA9z5Cuft
MWLT+WG2v9GR/qOtnj57kZZuU32aki9xnq4TZaa9/12yyfNZvtHrVcbV41XS2dtR3t3XUe7d11Hu3
ddR7t3XUe7d11Hu3ddR7t3XUe7d11Hu3ddR7t3XUe7d11Hu3ddR7t3XUe7d11Hu3ddR7t3XUe7d
11Hu38L8IAOzFOQzqyElk68hLRu7FVWzww1+T47hqtdOvgNTGqpbtwauf+MGsoPnBrKL5wK2i+s
Cto/rAraP6wK2j+8CtpPu/raT7v62k+7+tpPu/raT7v62k+7+tpPu/raT7v62k+7+tpPu/raT7v62k+7+tpP
u/raT7v62k+7+tpPu/raT78L8IAOzFOQzqyElk68hLRu7FVWzww1+T47hqtdOvgNTGqpbtwauf+MG
soPnBrKL5wK2i+sCto/rAraP6wK2j+8CtpPu/raT7v62k+7+tpPu/raT7v62k+7+tpPu/raT7v62k+7+tpP
u/raT7v62k+7+tpPu/raT7v62k+7+tpPu/raT78L8IAOzFOQzqyElk68hLRu7FVWzww1+T47hqtdOvg

NTGqpbttwauf+MGsoPnBrKL5wK2i+sCto/rAraP6wK2j+8CtpPu/raT7v62k+7+tpPu/raT7v62k+7+tp
Pu/raT7v62k+7+tpPu/raT7v62k+7+tpPu/raT7v62k+7+tpPu/raT7v62k+7+tpPu/raT7v62k+7+tp
zww1+T47hqtdOvgNTGqpbttwauf+MGsoPnBrKL5wK2i+sCto/rAraP6wK2j+8CtpPu/raT7v62k+7+tp
Pu/raT7v62k+7+tpPu/raT7v62k+7+tpPu/raT7v62k+7+tpPu/raT7v62k+7+tpPu/raT7v62k+7+tpPu/ra
yElk68hLRu7FVWzww1+T47hqtdOvgNTGqpbttwauf+MGsoPnBrKL5wK2i+sCto/rAraP6wK2j+8Ctp
Pu/raT7v62k+7+tpPu/raT7v62k+7+tpPu/raT7v62k+7+tpPu/raT7v62k+7+tpPu/raT7v62k+7+tpPu/ra
T78L8IAOzFOQzqyElk68hLRu7FVWzww1+T47hqtdOvgNTGqpbttwauf+MGsoPnBrKL5wK2i+sCto/
rAraP6wK2j+8CtpPu/raT7v62k+7+tpPu/raT7v62k+7+tpPu/raT7v62k+7+tpPu/raT7v62k+7+tpPu/ra
T7v62k+7+tpPu/raT78L8IAOzFOQzqyElk68hLRu7FVWzww1+T47hqtdOvgNTGqpbttwauf+MGsoP
nBrKL5wK2i+sCto/rAraP6wK2j+8CtpPu/raT7v62k+7+tpPu/raT7v62k+7+tpPu/raT7v62k+7+tpPu/ra
T7v62k+7+tpPu/raT7v62k+7+tpPu/raT78L8IAOzFOQzqyElk68hLRu7FVWzww1+T47hqtdOvgNTG
qpbttwauf+MGsoPnBrKL5wK2i+sCto/rAraP6wK2j+8CtpPu/raT7v62k+7+tpPu/raT7v62k+7+tpPu/ra
T7v62k+7+tpPu/raT7v62k+7+tpPu/raT7v62k+7+tpPu/raT78L8IAOzFOQzqyElk68hLRu7FVWzww1
+T47hqtdOvgNTGqpbttwauf+MGsoPnBrKL5wK2i+sCto/rAraP6wK2j+8CtpPu/raT7v62k+7+tpPu/ra
T7v62k+7+tpPu/raT7v62k+7+tpPu/raT7v62k+7+tpPu/raT7v62k+7+tpPu/raT78L8IAOzFOQzqyElk
68hLRu7FVWzww1+T47hqtdOvgNTGqpbttwauf+MGsoPnBrKL5wK2i+sCto/rAraP6wK2j+8CtpPu/ra
T7v62k+7+tpPu/raT7v62k+7+tpPu/raT7v62k+7+tpPu/raT7v62k+7+tpPu/raT78L
8IAOzFOQzqyElk68hLRu7FVWzww1+T47hqtdOvgNTGqpbttwauf+MGsoPnBrKL5wK2i+sCto/rAra
P6wK2j+8CtpPu/raT7v62k+7+tpPu/raT7v62k+7+tpPu/raT7v62k+7+tpPu/raT7v
62k+7+tpPu/raT78L8IAOzFOQzqyElk68hLRu7FVWzww1+T47hqtdOvgNTGqpbttwauf+MGsoPnBr
KL5wK2i+sCto/rAraP6wK2j+8CtpPu/raT7v62k+7+tpPu/raT7v62k+7+tpPu/raT7v
62k+7+tpPu/raT7v62k+7+tpPu/raT78L8IAOzFOQzqyElk68hLRu7FVWzww1+T47hqtdOvgNTGqpb
ttwauf+MGsoPnBrKL5wK2i+sCto/rAraP6wK2j+8CtpPu/raT7v62k+7+tpPu/raT7v62k+7+tpPu/raT7v
62k+7+tpPu/raT7v62k+7+tpPu/raT7v62k+7+tpPu/raT778AIAOvFOQzpyElk6sIKRe3GVGzvwF6T4
rlpttOvgNXFqpbvwaue+MGsoPnBraH5wK2i+sCto/rAraP7wK2j+7+to/u/raP7v62k+7+tpPu/raT7v62
k+7+tpPu/raT7v62k+7+tpPu/raT7v62k+7+tpPu/raT7v62k+7+tpPu/raT778AkAOvGOAvpyUEk6cpJ
RezHVGzuwV6T4rlottKvf9bEqpfxwaud+MGsn/jBraH5wK2i+sCto/rAraP7wK2j+7+to/u/raP8v62k+7
+tpPu/raT7v62k+7+tpPu/raT7v62k+7+tpPu/raT7v62k+7+tpPu/raT7v62k+7+tpPu/raT77sEkAOrGN
wvoyUAj6cpJROvHU2vuwl2T4rpnttKvf9bDqpzfzwauc+MGsn/jBraH5wK2i+cCto/rAraP7v62j+7+to/y/
raP8v62j/L+to/y/raP8v62j/L+to/y/raP8v62j/L+to/y/raP8v62j/L+to/y/raP8v62j/L+to/y/raP87sEjAOrHN
wvoykAj6MtlRorIU2vtw1yT4rtmttGwftfDqpf1wauc98GsnvjBraD5wa2i+cCuo/rArqP7v62j+7+to/y/ra
P8v62j/L+to/y/raP8v62j/L+to/y/raP8v62j/L+to/y/raP8v62j/L+to/y/raP8v62j/L+to/y/raP87cEjAOnHNg
vnyj8j58tlRorJUmvtxFuT4rxlt9GwftjCqpf3wqua98GtnfjBraD4wa6i+cCuo/rArqP7v62j+7+to/y/raP8v
62j/L+to/y/raP8v62j/L+to/y/raP8v62j/L+to/y/raP8v62j/L+to/y/raP8v62j/L+to/y/raP87cljAOnHNQvny
z4i58xHQ+nKUWrxfqS4rxkt9GwfjdjCqpX2wqyZ98KtnPfBrp/4wa6h+cCuovrArqP6v66j+7+to/y/ra
P8v62j/L+to/y/raP8v62j/L+to/y/raP8v62j/L+to/y/raP8v62j/L+to/y/raP8v62j/L+to/y/raP87MliAOjINqr
mzD4i5s1GQ+jKUWrrxVmS4r1jt9GxedjCqpP2wqyX9sKtm/fBrp74wa6h+cCvovrArqP6v66j+7+uo/u
/raP8v62j/L+to/y/raP8v62j/L+to/y/raP8v62j/L+to/y/raP8v62j/L+to/y/raP8v62j/L+to/y/raP87MMiAOjJ
NArlzD0h5c5GQuflUGnqxliS4r5ht9G0dNfcqpD1wqyV9sGumffBr534wK+g+cCvovq/rqP6v66j+7+
uo/u/rqP7v66j+7+uo/u/rqP7v66j+7+uo/u/rqP7v66j+7+uo/u/rqP7v66j+7+uo/u/rqP7v66j+7+uo/u/rqP
768QhAOfJMwrlzTwh5M9FQubMT2npYFeS4r9gt9K3btXDqov1wqyR98GulvjBr5z4wK+g+b+vovq/r
qL6v66j+7+uo/u/rqP7v66j+7+uo/u/rqP7v66j+7+uo/u/rqP7v66j+7+uo/u/rqP7v66j+7+uo/u/rqP7v66j+

7+uo/u/rqP768QgAObKMQnkzjsg49BEQeXOTmjoyVWR4sFdt9S+ZNLDqoL0wq2J98GwkPjAr5z5
v6+g+b+vofq/rqL6v66j+7+uo/u/rqP7v66j+7+uo/u/rqP7v66j+7+uo/u/rqP7v66j+7+uo/u/rqP7v66j+7+
uo/u/rqP7v66j+7+uo/u/rqP76sUfAOXLMAAnizzkg4dFCQOPQTGfmy1SR5MIWstTEV9LFr23wwq5o
97+wk/i/r5z5v6+g+b+vofq/rqL6v66i+r+uo/u/rqP7v66j+7+uo/u/rqP7v66j+7+uo/u/rqP7v66j+7+uo/u/r
qP7v66j+7+uo/u/rqP7v66j+7+uo/u/rqP76MceAOTNLgjh0Tcf39RAP+HSSmbjzlGQ3NNJsc3FXdH
DuGnpvbBo976vivi+r535vq+f+b6vofq+rqL6vq6i+r6uovq+rqP7vq6j+76uo/u+rqP7vq6j+76uo/u+rqP
7vq6j+76uo/u+rqP7vq6j+76uo/u+rqP7vq6j+76uo/u+rqP758gcAOLPKwje1DQd3Nc9Pd3WRmTf1E
uN0thGssbHZM7AvXfgvbaJ7L2zlfK9sZv1vbGe9rqwm/e3sJj3tbCX97KwlvewsbJb3r7CX96+wl/evsJf
3r7CX96+wl/evsJf3r7CX96+wl/evsJf3r7CX96+wl/evsJf35csZAN/RJgfb1zAc2ds
4PtnbP2fU4EORydhNscDLami8w3zWu76J3rq7kuO6uZfntbiT6bC3j+qst4zrqLaL7KW2i+2jtoztoba
N7aG2je2hto3tobaN7aG2je2hto3tobaN7aG2je2hto3tobaN7aG2je2hto3tobaN7aG2je2hto3t4c4UA
NvVHwbX2ywf1t45QtXfP2zK6UiSwdtUrbvQa765yX3KuMWJ0bfCkNezwl/brL6K3ae9h9+jvYbhn7y
F4py7heOau4fjmLuJ5Ji7ieSYu4nkmLuJ5Ji7ieSYu4nkmLuJ5Ji7ieSYu4nkmLuJ5Ji7ieSYu4nkmLuJ
5Ji7ieSYu4nk2NMJANbbGgjU3y4i0uL5R8vqQ3DB60+Pud9bpbbWbbO00Hu+tMyGxq/Jh8upxoTQ
pMWD0p/DgdSbw4HWI8KB15TCgtiSwYTZkMGG2ZDBhtmQwYbZkMGG2ZDBhtmQwYbZkMGG
2ZDBhtmQwYbZkMGG2ZDBhtmQwYbZkMGG2ZDBhtmQwYbZ1dUJANPfHgrR5DAnzeg7TchzS
m+571eJtOVhmrHebaew2HqxqNN4uaHQd7+bzXfDlct3xpHKeMmOyXnKjMI7y4rlfcylyIDNh8iCzYf
lgs2HyILNh8iCzYflgs2HyILNh8iCzYflgs2HyILNh8iCzYflgs2HyILNh8iCzYflgs2HyILN0tkJANDklg7
N6TlswvNBT7n6UWq09Vx9sOxmjqzkbZuj32ulmttrrJPXbLKM1W22iNNvuYTScbuB0XO8f9B1vX3
Qd7580Hq/e899v3vPfb97z32/e899v3vPfb97z32/e899v3vPfb97z32/e899v3vPfb97z32/e899v3vPfb
97z32/z+UOAczpJhLE8jcwufxIS7T/VV+w/GBxqfNjgZ/sYo6V5mGYjOJiolXfZKV/3Wepe9tpq3jabK
122m6udNlwr3LZc7Bx2HexcNh5sXDYebFw2HmxcNh5sXDYebFw2HmxcNh5sXDYebFw2HmxcN
h5sXDYebFw2HmxcNh5sXDYebFw2HmxyuoWAsfvKha6/D0utP9LQa/V1Sn/1plnPpZdJL0WYCH
7lmKfupbknfnXpdz5WKbb+NlnW3iaJ9r4mugauFtoWnhcKFn4XSiZ+B3omfgd6Jn4HeiZ+B3omfgd
6Jn4HeiZ+B3omfgd6Jn4HeiZ+B3omfgd6Jn4HeiZ+B3omfgd6Jn4Hei+rQaAPW7PQn0vkkf9b5QPv
e7W2D4tmiC7bF0n+Ksf7baq4vl0qqV1s2qm+DJqp/pxqef7sSknvLDo5/0wqKh9sKjo/fCpKb3wKW
m976lpve+paf2vqWn9r6lp/a+paf2vqWn9r6lp/a+paf2vqWn9r6lp/a+paf2vqWn9r6lp/a+paf2+rQaAP
W7PQn0vkkf9b5QPve7W2D4tmiC7bF0n+Ksf7baq4vl0qqV1s2qm+DJqp/pxqef7sSknvLDo5/0wqK
h9sKjo/fCpKb3wKWm976lpve+paf2vqWn9r6lp/a+paf2vqWn9r6lp/a+paf2vqWn9r6lp/a+paf2vqWn9
r6lp/a+paf2+rQaAPW7PQn0vkkf9b5QPve7W2D4tmiC7bF0n+Ksf7baq4vl0qqV1s2qm+DJqp/pxqef
7sSknvLDo5/0wqKh9sKjo/fCpKb3wKWm976lpve+paf2vqWn9r6lp/a+paf2vqWn9r6lp/a+paf2vqWn
9r6lp/a+paf2vqWn9r6lp/a+paf2+rQaAPW7PQn0vkkf9b5QPve7W2D4tmiC7bF0n+Ksf7baq4vl0qqV
1s2qm+DJqp/pxqef7sSknvLDo5/0wqKh9sKjo/fCpKb3wKWm976lpve+paf2vqWn9r6lp/a+paf2vqW
n9r6lp/a+paf2vqWn9r6lp/a+paf2vqWn9r6lp/a+paf2+rQaAPW7PQn0vkkf9b5QPve7W2D4tmiC7bF
0n+Ksf7baq4vl0qqV1s2qm+DJqp/pxqef7sSknvLDo5/0wqKh9sKjo/fCpKb3wKWm976lpve+paf2vq
Wn9r6lp/a+paf2vqWn9r6lp/a+paf2vqWn9r6lp/a+paf2vqWn9r6lp/a+paf2+rQaAPW7PQn0vkkf9b5Q
Pve7W2D4tmiC7bF0n+Ksf7baq4vl0qqV1s2qm+DJqp/pxqef7sSknvLDo5/0wqKh9sKjo/fCpKb3wK
Wm976lpve+paf2vqWn9r6lp/a+paf2vqWn9r6lp/a+paf2vqWn9r6lp/a+paf2vqWn9r6lp/a+paf2+rQaA
PW7PQn0vkkf9b5QPve7W2D4tmiC7bF0n+Ksf7baq4vl0qqV1s2qm+DJqp/pxqef7sSknvLDo5/0wq
Kh9sKjo/fCpKb3wKWm976lpve+paf2vqWn9r6lp/a+paf2vqWn9r6lp/a+paf2vqWn9r6lp/a+paf2vqWn
9r6lp/a+paf2+rQaAPW7PQn0vkkf9b5QPve7W2D4tmiC7bF0n+Ksf7baq4vl0qqV1s2qm+DJqp/pxq
ef7sSknvLDo5/0wqKh9sKjo/fCpKb3wKWm976lpve+paf2vqWn9r6lp/a+paf2vqWn9r6lp/a+paf2vqW

n9r6lp/a+paf2vqWn9r6lp/a+paf2+rQaAPW7PQn0vkkf9b5QPve7W2D4tmiC7bF0n+Ksf7baq4v10qq
V1s2qm+DJqp/pxqef7sSknvLDo5/0wqKh9sKjo/fCpKb3wKWm976lpve+paf2vqWn9r6lp/a+paf2vq
Wn9r6lp/a+paf2vqWn9r6lp/a+paf2vqWn9r6lp/a+paf2+rQaAPW7PQn0vkkf9b5QPve7W2D4tmiC7b
F0n+Ksf7baq4v10qqV1s2qm+DJqp/pxqef7sSknvLDo5/0wqKh9sKjo/fCpKb3wKWm976lpve+paf2v
qWn9r6lp/a+paf2vqWn9r6lp/a+paf2vqWn9r6lp/a+paf2vqWn9r6lp/a+paf2+rQZAPS8Pgnzvkgf9L9Q
Pfe8W2D3tmeC7bFzoOKtfrfZq4nJ0auT18yqmuPHqp7sxKqh8sKooPbBqKH5wKmi+sCspPrAraT6
wK2k+r+spPq+qqT6vqqk+r6qpPq+qqT6vqqk+r6qpPq+qqT6vqqk+r6qpPq+qqT6vqqk+r6qpPq+qq
T6+bUZAPS8Pgnzv0gf9L9PPfa8WmD3t2aC7bJyoOKue7fZrYbJ0auR2cqmqOXFq57vwquh98Cso
vrArKP6wK2j+sCtpPrAraT6wK2k+r+tpPq/raT6v62k+r+tpPq/raT6v62k+r+tpPq/raT6v62k+r+tpPq/ra
T6v62k+r+tpPq/raT6+bUZAPO9Pgnyv0ce88BPPPW9WV/3uGWC7bNxoOKwebfZroPK0KyP2smrl
+jEq53zwayh+cCsovrAraP6wK2j+sCtpPvAraT7v62k+7+tpPu/raT7v62k+7+tpPu/raT7v62k+7+tpPu
/raT7v62k+7+tpPu/raT7v62k+7+tpPu/raT7+LYYAPO9PgjyWEce8sBPPPW+WV/2uGWC7bNwoOK
xdrbZr4DK0K2M2sirlunCq532wayg+cGsovnAraP6wK2j+sCtpPu/raT7v62k+7+tpPu/raT7v62k+7+t
pPu/raT7v62k+7+tpPu/raT7v62k+7+tpPu/raT7v62k+7+tpPu/raT7+LYYAPK+PgjxwEYe8sFOPPW
+WF/2uWSC7bRwoeKzdLbZsH3J0K2K28irlOvBq534wayf+cGsofnAraP6wK2j+sCto/u/raT7v62k+7
+tpPu/raT7v62k+7+tpPu/raT7v62k+7+tpPu/raT7v62k+7+tpPu/raT7v62k+7+tpPu/raT7+LYYAPK+
PQjxwEYd8cFOPPS/V1/1umSC7bRvoeO0crXZsnrJ0K6H28erk+zBq5z4wayf+MGtofnAraL6wK2j+
7+to/u/raP7v62k+7+tpPu/raT7v62k+7+tpPu/raT7v62k+7+tpPu/raT7v62k+7+tpPu/raT7v62k+7+tpP
u/raT797cXAPK+PQjxwUUd8cJNO/S/V1/1umOC7bZtoOO2b7TZtHfI0LCE2seskOzCq5r3wayd+M
GtoPnArqL5wK6j+7+to/y/raP8v62k+7+tpPu/raT7v62k+7+tpPu/raT7v62k+7+tpPu/raT7v62k+7+tpP
u/raT7v62k+7+tpPu/raT797cXAPG/PAjwwUUd8cJNO/PAVI71u2KC7bdrn+O4bbPatnPH0LGA2se
sjezCq5f2wq2c98Gun/jBrqL5wK6j+r+to/y/raP8v62k+7+tpPu/raT7v62k+7+tpPu/raT7v62k+7+tpPu/r
aT7v62k+7+tpPu/raT7v62k+7+tpPu/raT797cXAPG/PAfwwUUd8MNNO/PAVI70u2KB7rlpnuS6arLa
uW7G0bR72MitievCrJT2wq2Z98GunvjBr6H5wK6j+r+uo/u/raP7v62k+7+tpPu/raT7v62k+7+tpPu/ra
T7v62k+7+tpPu/raT7v62k+7+tpPu/raT7v62k+7+tpPu/raT79rgWAPG/OwfvwkQc8MNMOvLBVV70
vGGB77tnneW9Z7DbvmjE0rh11smwgunCrl72wq6V98Gvm/jAr6H5v66i+r+uo/u/rqP7v66k+7+upPu
/rqT7v66k+7+upPu/rqT7v66k+7+upPu/rqT7v66k+7+upPu/rqT7v66k+7+upPu/rqT79rgWAPDAOwf
vw0Qc78RMOvLCVV7zvWCB8L5km+bBY67dxGLA08Bs08mzd+nCrH32wbCl+L+vnPm/r6D5v66i
+r+uo/u/rqP7v66k+7+upPu/rqT7v66k+7+upPu/rqT7v66k+7+upPu/rqT7v66k+7+upPu/rqT7v66k+7
+upPu/rqT79bkVAPDAOgfuw0Mc7sRLOfHDFV3zvV+B7sFgmOjHXqrdzlu80cRm1Mi5aufBsmL0vr
CP+L6vnPm+r6D5vq6i+r6uovq+rqP7vq6j+76upPu7rqP7u66j+7uuo/u7rqP7u66j+7uuo/u7rqP7u66j
+7uuo/u7rqP7u66j+7uuo/u7rqP79LoUAO/BOQftxElb7cVKOfDEUI3xv12A6sdbk+HRWKPU0Fi9ys
Rr08O7d+O+tYbtvbKU872wnPa9sJ74ua+a+LWvmfmzr5r4sbCc+LCwoPeusKD2rrCg9q6woPaus
KD2rrCg9q6woPausKD2rrCg9q6woPausKD2rrCg9q6woPausKD287sTAO7CNwbsxUEa7MdJOO/
FUVztxVh749BVi9jcSqXN0Fy+xMVx0b++ft69uovmvLeU67y1mu62tJXwsLOR8auzkPKos5HyprO
V8qa0m/GmtJ3wprSd8Ka0nfCmtJ3wprSd8Ka0nfCmtJ3wprSd8Ka0nfCmtJ3wprSd8Ka0nfCmtJ3w
8rwSAO3DNQbrxz8Z6shHN+zHUFvmzFJz2N1Jis/eTafF0WG9v8h1zbzCg9e6vo3euryU47a6lOau
uY3oqLiK6qO3ieugt4vrnreP6563leqet5fqreX6p63l+qet5fqreX6p63l+qet5fqreX6p63l+qet5fqre
X6p63l+qet5fq8b4QAOvFMwXoyTwY58tFNenKTlna2kZrz+RLjcfEuae/02a6u8t3x7nGhdC4w47Wt
sCS2q6/i92nvYffobyF4Z28heKau4bjl7uK45a7j+OWvJHilryR4pa8keKWvJHilryR4pa8keKWvJHilry
R4pa8keKWvJHilryR4pa8keKWvJHi7sAMAOHLwTmzDkX5M5BM9/URU/Q5EZvx+xPjr/fWKW51
mq0ttB5v7XLhcezylvNrcal0abEhNSgw4LWm8KB15fBgdiUwYPZkcGF2pDBitqPwYvaj8GL2o/Bi9q

PwYvaj8GL2o/Bi9qPwYvaj8GL2o/Bi9qPwYvaj8GL2o/Bi9qPwYva3cQHAOXLKQPh0DMU39Q8M
NLjP0/H7kpyvu1UjLjX5+0222sstV6tq3Rfr2lznvCn8x6xprKesmWyXrLksh7zl/HfM6Nx37Oi8eAz4n
GhM+IxoXPiMaFz4jGhc+IxoXPiMaFz4jGhc+IxoXPiMaFz4jGhc+IxoXPiMaFz4jGhc+IxoXP2McGA
ODQIALb1ioS1d81MMjuQIS+9k9xt/Bbh7LnZpev4HCip9xwq5/Yb7GY1W+2ktNwuY3RcLyK0HK+h
89zv4TPdcGCznfCgM55w3/NfcN+zX7Dfs1+w37NfsN+zX7Dfs1+w37NfsN+zX7Dfs1+w37NfsN+z
X7Dfs1+w37NfsN+zX7D1coHANjaEALV3ScWyu04Nr73SFS2/FZssfVhfqvitalyi52aXmuJln5LeZqa
L3GerhtpproLYa7F/12yzfNZutHvVcbV51XO2d9R2t3XUerd11Hu3ddR7t3XUe7d11Hu3ddR7t3XUe
7d11Hu3ddR7t3XUe7d11Hu3ddR7t3XUe7d11Hu30s4HANPfFgTM6Swbv/c+N7b+TICw/1xkqftgc6
D0XoCW7V2LjeldlIXpqa4mGfe+BkonffZqV03mimct1rqHHdbahv3HCpbtxzqmzcd6ts3HirbNx4q2
zceKts3HirbNx4q2zceKts3HirbNx4q2zceKts3HirbNx4q2zceKts3Hirb9QHAM7mHQJA9zletv9ENa//
U0io/1ZZnv9WZ5T6VnSK9FZ+gfBXh3nsWY1z6luSb+hflWznYpdq5mWZaeVommflaptm5G2bZeR
wnGTjdJ1j43adY+N2nWPjdp1j43adY+N2nWPjdp1j43adY+N2nWPjdp1j43adY+N2nWPjdp1j43adz
OIIAML0Jgq3/zgdr/9ILqb/Sj2c/0tNkv9MW4j/Tmh+/E9ydfdSem70VX9o8liDZfBbhmLvX4hg7mKKX
+5li17tZ4xd7WqMXOxtjVvscY1a7HOOwuxzjlrc45a7HOOwuxzjlrc45a7HOOwuxzjlrc45a7HO
OWuxzjlrc45a7HOO/6kQAP6xMQf8tUca/LdTNf20XIT9r2xy96x3jO2pgKDmqlev36aQvNqjlcBwn5j
O05ub1NGVndjPkaDbz4+k3c6Op97Ojazezo6x3syQtd7HkrXexJS23cSUtt3EILbdxJS23cSUtt3EIL
bdxJS23cSUtt3EILbdxJS23cSUtt3EILbd/6kQAP6xMQf8tUca/LdTNf20XIT9r2xy96x3jO2pgKDmqle
v36aQvNqjlcBwn5jO05ub1NGVndjPkaDbz4+k3c6Op97Ojazezo6x3syQtd7HkrXexJS23cSUtt3EIL
bdxJS23cSUtt3EILbdxJS23cSUtt3EILbdxJS23cSUtt3EILbd/6kQAP6xMQf8tUca/LdTNf20XIT9r2xy
96x3jO2pgKDmqlev36aQvNqjlcBwn5jO05ub1NGVndjPkaDbz4+k3c6Op97Ojazezo6x3syQtd7Hkr
XexJS23cSUtt3EILbdxJS23cSUtt3EILbdxJS23cSUtt3EILbdxJS23cSUtt3EILbd/6kQAP6xMQf8tUca
/LdTNf20XIT9r2xy96x3jO2pgKDmqlev36aQvNqjlcBwn5jO05ub1NGVndjPkaDbz4+k3c6Op97Ojaz
ezo6x3syQtd7HkrXexJS23cSUtt3EILbdxJS23cSUtt3EILbdxJS23cSUtt3EILbdxJS23cSUtt3EILbd/6
kQAP6xMQf8tUca/LdTNf20XIT9r2xy96x3jO2pgKDmqlev36aQvNqjlcBwn5jO05ub1NGVndjPkaDb
z4+k3c6Op97Ojazezo6x3syQtd7HkrXexJS23cSUtt3EILbdxJS23cSUtt3EILbdxJS23cSUtt3EILbdxJ
S23cSUtt3EILbd/6kQAP6xMQf8tUca/LdTNf20XIT9r2xy96x3jO2pgKDmqlev36aQvNqjlcBwn5jO05
ub1NGVndjPkaDbz4+k3c6Op97Ojazezo6x3syQtd7HkrXexJS23cSUtt3EILbdxJS23cSUtt3EILbdxJ
S23cSUtt3EILbdxJS23cSUtt3EILbd/6kQAP6xMQf8tUca/LdTNf20XIT9r2xy96x3jO2pgKDmqlev36a
QvNqjlcBwn5jO05ub1NGVndjPkaDbz4+k3c6Op97Ojazezo6x3syQtd7HkrXexJS23cSUtt3EILbdxJ
S23cSUtt3EILbdxJS23cSUtt3EILbdxJS23cSUtt3EILbd/6kQAP6xMQf8tUca/LdTNf20XIT9r2xy96x3j
O2pgKDmqlev36aQvNqjlcBwn5jO05ub1NGVndjPkaDbz4+k3c6Op97Ojazezo6x3syQtd7HkrXexJ
S23cSUtt3EILbdxJS23cSUtt3EILbdxJS23cSUtt3EILbdxJS23cSUtt3EILbd/6oQAP6xMQb7tkca+7d
TNf20XIT9sGty96x3jO2rfqDmqYSv36iOvNqolsfVpZrQ0p+b1tCZnNvOIJ/ezZGj4MyQp+HMkKvizJ
Cx4siTs+LDIbThwJW04cCVtOHAibThwJW04cCVtOHAibThwJW04cCVtOHAibThwJW04cCVtOH
AlbTh/6oPAP2yMQb6t0gZ+rhSNPy1XVT9sWpy9612je2se5/mq4Gv36qKvdmpk8nUqJrS0aSc2c6
enN7MmJ7iy5Sh5MqSpuxKKqvmYzSx5sOXseW/l7HlvJex5byXseW8l7HlvJex5byXseW8l7HlvJex
5byXseW8l7HlvJex5byXseW8l7Hl/6sOAPyzMQb5t0kZ+rlSNPy2XFP8sWly9650jO6ueZ/mrX6v36u
GvdmqkMnTqpft0Kmd2s2jneDLnJ3lyZeg6MiVpenllqzqxJiu6sCar+m7ma/puZmv6bmZr+m5ma/pu
Zmv6bmZr+m5ma/puZmv6bmZr+m5ma/puZmv6bmZr+m5ma/p/6sOAPyzMQb5uEkZ+bIRNPu2XF
P8smly97Byi+6wdp7mr3uu362DvdmsjMrTq5XUz6qb3MyonuLJoZ3nyJuf68eYpe3GmaztwZys7by
cre24m63stpqt67aareu2mq3rtpqt67aareu2mq3rtpqt67aareu2mq3rtpqt67aareu2mq3r/6wNAPu0M
Qb4uUkY+bpRM/u3W1P8smhy+LFwui+xdJ3msXit369/vNmticnTrJLUz6uY3cuqneTlpp7qxp+e7s

WbpPDDnarwvp+r8Lmeq/C2navvs5ys7rOcrO6znKzus5ys7rOcrO6znKzus5ys7rOcrO6znKzus5ys
7rOcrO6znKzu/6wNAPu0MQX4uUoY+LpRM/u3W1P7s2hy+LJuie+zcpzns3as4LJ7u9mvhcnTro/U
zqyW3curm+XHqqDrxaOe8MSeo/PAoKnzu6Gp87egqfKznqrxsZ2r77Gdq++xnavvsZ2r77Gdq++xn
avvsZ2r77Gdq++xnavvsZ2r77Gdq++xnavv/q0NAPu0MQX3ukkY+LpQM/q4WIP7s2dy+LRsiPC1cJ
vntXOr4LR3utqygjTslvTz62T3cqsmeXHq57txKmg8sKiova+pKf2uaOn9bWhqPSxn6nyr56q8K+eq
vCvnqrwr56q8K+eqvCvnqrwr56q8K+eqvCvnqrwr56q8K+eqvCvnqrw/a0MAPq1MQX3ukkY+LtQM/
q4WIL7tGdy+LVrh/C3bprot3Gq4bd1udq1fcbUs4fSz7CP3MtleXHrJvtw6ug9MGmovi7p6X4t6Sm9
7Oip/WvoKnzrZ+q8a2fqvGtn6rxrZ+q8a2fqvGtn6rxrZ+q8a2fqvGtn6rxrZ+q8a2fqvGtn6rx/K0MAPq1
MQX3ukkX97tQMvq5WVL6tGZy97ZphvG5bJjpum6o4rpyt9u5eMTVt4LQ0LSK2cyxkePIr5frxK2c8
8Gomvi4qKX6s6am+K+jp/asoanzqp+q8aqfqvGqn6rxqp+q8aqfqvGqn6rxqp+q8aqfqvGqn6rxqp+q8
aqfqvGqn6rx+64MAPq2MQX3ukgX97tQMvm5WVL6tWVx9rnhhPK7aZbqvWym479utN3AcsHYv3
vM0r2F1c+8jNzJuJDnw7OW8bivmviwrqX5raim96ulp/Wooqnzp6Cq8aegqvGnoKrxp6Cq8aegqvGn
oKrxp6Cq8aegqvGnoKrxp6Cq8aegqvGnoKrx+q4LAPm2MQX2u0gX97xPMvm6WFL5t2Nw9bplgv
G+Z5TrwWij5cVrsODJbbvazHXF0cWB08q+h97Dul/ovrSZ8LGxIPWssZ/1p66n9aWoqPSkpKnyo6
Kq8aOiqvGjoqrXo6Kq8aOiqvGjoqrXo6Kq8aOiqvGjoqrXo6Kq8aOiqvGjoqrX+K4LAPm2MQX2u0cX9
rxPMvm6WFL4uGFu871igO/CY5Hqx2Wf5M1nq9vTarfRy3XHysOA1MO+iN+/uZLnvLaa7ayzj/Kns
5jyo7Oj8p+vqfKeqKrxnaWr752lq++dpavvnaWr752lq++dpavvnaWr752lq++dpavvnaWr752lq++dpa
vv968KAPi3MQT1vEcW9b1PMfi7V1H2u19s8cBffevHYlzkz2GZ3Nlgp9LTabrKynbJxMOC1b++i96
8u5TktreU6qe1jO6htZLvo7We7Zu1pu6YsKzumKus7ZirrO2Yq6ztmKus7ZirrO2Yq6ztmKus7ZirrO2
Yq6ztmKus7ZirrO2Yq6zt9LAJAPe4MQT1vEYw9b5OMfe8VIH0vIxp7cVbeOXOXIXc2lqU1Nxcqsv
SarvEynnKv8SF1LzAjtuvJXhr7qO5qO4iemdt47qnbiX6pu4oOmUt6nqkrSu6pK0ruqStK7qkrSu6p
K0ruqStK7qkrSu6pK0ruqStK7qkrSu6pK0ruqStK7q8bAIPa5MQT0vUUV9L9NMPa9VVHxwldk58x
XcdzZVX7U4VaYzNterMTRbby+ynvlu8WH0bnCj9i1v5HdqLyJ4Z+7huSZuorImLqS5Zm7m+SVu6
PkkLqp5ZC6qeWQuqnlkLqp5ZC6qeWQuqnlkLqp5ZC6qeWQuqnlkLqp5ZC6qeWQuqnl7bEHAPW
6MQTzvKQV88BML/W+VFDryFNe3tZSZ9PiUYLM5VebxNphrb7ScLu6zH7GuMeJzrfEkNStwYrYo
7+F3Ju+hN6WvYbgk72M4JS9ld+Vvp7ekb6k3pG+pN6RvqTekb6k3pG+pN6RvqTekb6k3pG+pN6
RvqTekb6k3pG+pN6RvqTe57MFAPS7MQPxv0MU8cFLLvDDUEri0UxT1OFMasvpUoXE5Fmcvdtl
rLjTc7i2zoDCtMqJya7HiM6mxYXTnsOB1pfCgdiSwYTZkMGI2Y/Bj9mPwZjYkMKe15DCnteQwp7X
kMKe15DCnteQwp7XkMKe15DCnteQwp7XkMKe15DCnteQwp7X37UDAPK9MQPvwUET78NJLe
jLSUDW30ROzOINbsPvVYe85F6bt9xpqbTWdrSy0YG8qs5/wqPLfcicyXzLl8zpLGftCOxYHRjM
WE0orFitKKxZHRisaW0YrGltGKxpbRisaW0YrGltGKxpbRisaW0YrGltGKxpbRisaW0YrGltGKxpbR
2rcCAPDAMALtxD0R7MZGK9nbOjLN6UVTw/FQcLzwWYe25mOYst9spK/aeK6n1Xa1n9J1u5jQd
L+SznXCjcx2xYnLeMaGy3rIhMp+ylLKgsmByojJgMqNyIDKjciAyo3IgMqNyIDKjciAyo3IgMqNyIDKj
ciAyo3IgMqNyIDKjciAyo3I2LkCAOzDLwHpyDgP4tA6lc/nPDjE8khXu/hUcLXyXoOw6miSq+NtnaLe
bKaa22usk9hssY3VbbWl1G63hNNwuYDSrt+0XS8fNF3vXrQe7140IC+eNCFvnjQhb540IW+eNC
FvnjQhb540IW+eNCFvnjQhb540IW+eNCFvnjQhb540IW+1rsDAOfIJgDjzTEN0eMwH8XyPzy7+k1
XtP5ZbK/2ZHyo7maJnulijJXkYpuN4WOhh99pYLdZqI922irettqrXjabK512W6vdNlxsHLYdLBw2H
mxb9h8sW/YfLFv2Hyxb9h8sW/YfLFv2Hyxb9h8sW/YfLFv2Hyxb9h8sW/YfLFv2Hyx078DAODPGA
DV3R4KxvE1I7v7RD2z/1JTrv9fZaT7XXOZ9Vt+kO9ah4frW46A6VuUe+ZdmHfIYJt042OdceJln2/iZ
6Bt4WqhbOBsomrgb6Np4HOkAN93pGjfd6Ro33ekaN93pGjfd6Ro33ekaN93pGjfd6Ro33ekaN93p
Gjfd6Ro33ek0MMEANbZCQDH7igOu/s5JLP/Sjqr/1RMoP9TW5b/U2iN+1NyhPZUe3zyVYJ271eH
cO1Zi2zrXI5p6l+RZ+likmbpZJNk6GeUY+hqIWLnbZZh53GXYOd0I2DndJdg53SXYOd0I2DndJdg5

3SXYOd0l2DndJdg53SXYOd0l2DndJdg53SXzcgGAMrqFQG8+i8Qs/8/lqn/RzOe/0dCIP9lUlv/SlyC
/0xnef1Ob3H5UHZr91N7ZvVWfmPzWYFh8lyDX/FfhF3xYoVc8GWGW/Bnh1rva4hZ72+IWO9yiVjv
colY73KJWO9yiVjvcolY73KJWO9yiVjvcolY73KJWO9yiVjvcolY73KJys8GAL34lQOy/zAPpv81HJ3/
OSqT/zw4if8/RX//QlF2/0Vbbv9lY2f/TGlh/IBTxfTcFr7V3NY+lp0VvpddVX5YHZU+WN3U/hleFL4aX
hR9215UPdweVD3cHIQ93B5UPdweVD3cHIQ93B5UPdweVD3cHIQ93B5UPdweVD3cHIQ93B5/5
4KAP+IJwX/qToW/6pJLv+nV0n/omRk/5xuevecdovwmn2a6paEpuWSi7DhjJG43oiWvtyEm8LbgaD
F2n+mx9l/q8jZfrDJ2X+2ydl/vcnWgcLlz4TDyMqGw8fKhsPHyobDx8qGw8fKhsPHyobDx8qGw8fKh
sPHyobDx8qGw8fKhsPH/54KAP+IJwX/qToW/6pJLv+nV0n/omRk/5xuevecdovwmn2a6paEpuWSi
7DhjJG43oiWvtyEm8LbgaDF2n+mx9l/q8jZfrDJ2X+2ydl/vcnWgcLlz4TDyMqGw8fKhsPHyobDx8q
Gw8fKhsPHyobDx8qGw8fKhsPHyobDx8qGw8fKhsPH/54KAP+IJwX/qToW/6pJLv+nV0n/omRk/5x
uevecdovwmn2a6paEpuWSi7DhjJG43oiWvtyEm8LbgaDF2n+mx9l/q8jZfrDJ2X+2ydl/vcnWgcLlz4
TDyMqGw8fKhsPHyobDx8qGw8fKhsPHyobDx8qGw8fKhsPHyobDx8qGw8fKhsPH/54KAP+IJwX/
qToW/6pJLv+nV0n/omRk/5xuevecdovwmn2a6paEpuWSi7DhjJG43oiWvtyEm8LbgaDF2n+mx9l/q
8jZfrDJ2X+2ydl/vcnWgcLlz4TDyMqGw8fKhsPHyobDx8qGw8fKhsPHyobDx8qGw8fKhsPHyobDx8
qGw8fKhsPH/54KAP+IJwX/qToW/6pJLv+nV0n/omRk/5xuevecdovwmn2a6paEpuWSi7DhjJG43oi
WvtyEm8LbgaDF2n+mx9l/q8jZfrDJ2X+2ydl/vcnWgcLlz4TDyMqGw8fKhsPHyobDx8qGw8fKhsPH
yobDx8qGw8fKhsPHyobDx8qGw8fKhsPH/54KAP+IJwX/qToW/6pJLv+nV0n/omRk/5xuevecdovw
mn2a6paEpuWSi7DhjJG43oiWvtyEm8LbgaDF2n+mx9l/q8jZfrDJ2X+2ydl/vcnWgcLlz4TDyMqGw8
fKhsPHyobDx8qGw8fKhsPHyobDx8qGw8fKhsPHyobDx8qGw8fKhsPH/54JAP+IJwX/qTsW/6pJLv
+nV0n/o2Rk/55ueveedovwnH2a6piDpuWUi7DhjPC43omVv9yFmsPagqDG2YClyNmAq8rYgLDK2
IC3ytiBvsrUg8LKzobCycmHwsnJh8LJyYfCycmHwsnJh8LJyYfCycmHwsnJh8LJyYfCycmHwsnJh8
LJ/58JAP+mJwT/qjsV/6xJLf+pV0n/pGVk/6Juefijdorwon2a6p6CpuWairHglZC63Y+VwdqKmcbyH5/
K14SlzNaDqs7Wg7HO1oS4ztWFv87OiL/NylvAzcOLv83Di7/Nw4u/zcOLv83Di7/Nw4u/zcOLv83Di
7/Nw4u/zcOLv83Di7/N/aAIAP+nJwT/qzsV/61KLf+qWEn/pmVk/6ZvePiod4nwqH6Z6qWDpuWgibH
gm4+73JWUwtmQmcnWi57N1Yik0NSHqtHUh7HS1li60s+KvdHJjb3Rw4690b+OvdG/jr3Rv4690b
+OvdG/jr3Rv4690b+OvdG/jr3Rv4690b+OvdG/jr3R+6AHAP+oJwT/rDwU/65KLf+rWEj/p2Vk/6pvd/
msd4jxq3yY6qmCpuWmibHgoY+73JuUw9iVmMrVkKJ3P04yj09OJqtXSibLV0Yq71cqOu9TEj7vUvo
+71LqPu9S6j7vUuo+71LqPu9S6j7vUuo+71LqPu9S6j7vUuo+71LqPu9S6j7vU+qEHAP+oJwT/rTw
U/65LLP+sWEj/qWZk/q1vdvmtldfyrXqX66t/peWphrDgp5C726GUxNibmMvUIZzR0o6h1dGKqdfRir
PYzl652MWRute/kbnXupC517aQute2kLrXtpC617aQute2kLrXtpC617aQute2kLrXtpC617aQute2k
LrX+KEGAP+pJwT/rTwU/69LLP+tWUj/q2Zj/q9tdPqvcoXyr3eV6619o+Wrgq/gqYy63KeUxNihmMz
UmpzS0ZGg19CMqdrQjLXax5G42sCSuNq6krjatpG42rKRuNmykbjZspG42bKRuNmykbjZspG42b
KRuNmykbjZspG42bKRuNmykbjZ96IGAP+pJwT/rjwU/7BMLP+uWUj/rWZi/bBrc/uxclTzsXSU7LB6
ouauf67hrle53KqRw9iomMvToJvT0JSf2c+NqNzMjrbdwpo23LuTtt22k7bdspK23K+St9uvkrfbr5K3
26+St9uvkrfbr5K326+St9uvkrfbr5K326+St9uvkrfbr9qlGAP+qJwP/rj0T/7BMLP+vWUj/r2Zg/LJqcvqz
bYL0s3KS7bN3oOeyfKzislO33a2MwdiqlcrUp5zT0Jmd2s2PqN/GkrXfvJS137WUtOCxILXfrZO13q
uTtt2rk7bdq5O23auTtt2rk7bdq5O23auTtt2rk7bdq5O23auTtt2rk7bd9aMFAP+qJwP/rz0T/7FMK/+w
WUj+sWVf+7NocPm1a4D1tm+Q7rZ0nui2earjtX603rKHvtmvmkMjUrJnR0KKd2syRp+G/lbPitZWz4r
CVs+KslbPhqZS04KeUtd6nlLXep5S13qeUtd6nlLXep5S13qeUtd6nlLXep5S13qeUtd6nlLXe86MF
AP+rJgP/rz0T/7JNK/+wWkf+smNe+rVmb/i3aX71uW2N8Lpxm+q7dqblu3ux4LqCuty4jMPXtJXM0a
yc2MWVpOO0lrHmrJax5amWsuSnlrPjpZW04aOVtd+jlbXfo5W136OVtd+jlbXfo5W136OVtd+jlbXfo
5W136OVtd+jlbXf8qMEAP+rJgP/sD0T/7JNK/+xWkf9s2Jc+bdkbfa6Z3zzvGqK8b9ul+zBc6Lnw3ir4

8d+s+DJibvZw4/J0bqP17OgmOSjmq/popmw56GYseagmLLkoJez4p+WtOCflrTgn5a04J+WtOCflr
Tgn5a04J+WtOCflrTgn5a04J+WtOCflrTg8KQEAP+rJgP/sD4T/rNNK/+yWkf8tWBb+Llia/S9ZHnxw
GeH7cRrkurJcJzmz3ek4dmAqdnThrjPy4vlwsON1qWvjeSQqajqlaGv6JidsOaZm7Lkmpmz4pqYtO
CamLTgmpi04JqYtOCamLTgmpi04JqYtOCamLTgmpi04JqYtOCamLTg76QDAP+sJgP/sT4S/rRO
Kv+zWkf7t15Z9rxfPLAYXbuxmSC6MxpjOPTcJPe3neZ2N2Ap8/UhrfFy4zGusOR1Zy8heGKup7
mjLCx5o+mseWROLLkk52z4pSbtOCUm7TglJu04JSbtOCUm7TglJu04JSbtOCUm7TglJu04JSbtO
CUM7Tg7aUDAP+sJgP+sj4S/bROKv60Wkf5uVtW9L9dZe7FX3HozGJ74dVogtvfao7V4XCfzdt7rc
bThLm+y4zGtsST05u+hN6PvJXhjLyq4Ymzs+KLqbPijaO04Y6ftd+On7Xfjp+1346ftd+On7Xfjp+134
6ftd+On7Xfjp+1346ftd+On7Xf6qUCAP+tJgP+sj4S/LVPKv21WEb3vFIT8MNaYOnLXGvg1WBy2d9
igdPkZ5TM4G6jxNh4sb3Rgr24yovlrsWM05jAg9uQv4/dksCh24i+rd2FtrXeh6y13oimtt2lprbdiKa23
Yimtt2lprbdiKa23Yimtt2lprbdiKa23Yimtt2lprbd56YBAP+uJgL9sz8R+7dQKfy3Vkp0v1dP68hXWuH
TWWHY31tw0eVghcrMZjD3m2nvdZ4s7fQgr6zyovlpsaF0ZbCgteOwYrZsGY2l3CpdIFwbDZgrm
42oOwuNqDsLjag7C42oOwuNqDsLjag7C42oOwuNqDsLjag7C42oOwuNqDsLja46cAAP+vJgL7t
UAR+rhQKfi6VD/vw1NJ5NBTUNjeU13Q5ll1yetfisLkZZu83W6pttZ5tLPQg76pzILGnsh/zZTFgNKM
xlbUi8SR1lzEnNOKxKfTgsOy1X6+u9V+vrVfr671X6+u9V+vrVfr671X6+u9V+vrVfr671X6+u9V+
vrV3qgAAP6wJgL6tkAQ+bIPKPS+UTnoyk5A2dxKR9DmUmLI7VI5we1gjbvkZ5y13XCpsdZ6s6rSf
bugzXrDlsp6yY7IfMyJx4LPhseKz4bHIM6GyJ7NhMipzX7Hss9+x7LPfseyz37Hss9+x7LPfseyz37H
ss9+x7LPfseyz37Hss9+x7LP3KkAAP2yJQH4uEEP+LpOJ+7ETDHD1kUz0eVKTMjtU2bB81p8uuxi
jrTkaZyw3nKnqtl2sKDUdbiX0HS+j851wojMeMaDy33lgcqDyIDKi8h/y5XHgMugxn/MqcZ/zKnGf8yp
xn/MqcZ/zKnGf8ypxn/MqcZ/zKnGf8ypxn/MqcZ/zKnG2qoAAPu0JAH2ukIO9b1LJOTOQibS4kE1ye
5LUcD0VGm59V19tOxlja/lbJqo4G+kn9turJbYbrKO1G+3h9Jxu4LQdL1+z3i/e899wHrPhMB5z4zA
ec+Wv3nQn7550J++edCfvnnQn7550J++edCfvnnQn7550J++edCfvnnQn7550J++2awAAPi2lwH0
vUEN7MVCgtXeNB/K7UI6wPZNVLn6WGqz9mF7ru5piqboaJWc42eelN9npYzcaKuF2WqvgNhss
nvWb7R41XO1ddR4tnTUfbdz1IO3ctSMt3LVILZy1ZS2ctWUtnLVILZy1ZS2ctWUtnLVILZy1ZS2ctW
UtnLVILZy1ZS2164AAPW6lgDwwD4L3NUqDMvqOCTB9kQ+uPxRVbL/XGit+GV3o/FihJnsYI6Q5
1+WiORhnlLhY6F832WleN5op3Tda6lx3G6qb9tyq23bdqxs23usa9uCrGrbiqx24msatuJrGrbiqx2
4msatuJrGrbiqx24msatuJrGrbiqx24ms1LEAAPC/HgDnyS4FzeczEMH1Oye4/kg+sf9VUqr/XWO
g/FtxlvZZfl3xWYWE7VmMfepbkXfoXZVz5mCYb+Vjmm3kZpxr42mdaeNsnmficJ9m4nSfZOJ5oGT
if6Bk4n+gZOJ/oGTif6Bk4n+gZOJ/oGTif6Bk4n+gZOJ/oGTif6Bk4n+g0rUAAOnGFQDQ4hYCwvQw
E7j/Piiw/048p/9STZz/UVuS/1FnifRcYD4Unh49VR+cvJWg23wWIZp71qJZu5di2TtYlxi7GONYetm
jl/raY9e62yQXepxkVzqdZFc6nWRXOp1kVzqdZFc6nWRXOp1kVzqdZFc6nWRXOp1kVzqdZFc6n
WRz7kBANLQCADD8ilEuP41FK//Qyak/0U2mf9FRI7/RIGE/0dbe/9JZHT/TGpu/U5waPpRdGT5VH
dh91Z6X/ZZfFz1XH1b9V9+WfRif1j0ZYBX82iBVvNtglXycYJV8nGCVfJxglXycYJV8nGCVfJxglXyc
YJV8nGCVfJxglXycYJV8nGCy74CAMPaBAC4/igFrP8wEqH/NSCV/zcui/86OoH/PUV4/0BPcP9D
WGn/R19k/0pkX/9OZ1v/UWpZ/1RtV/5XbIX9WnBT/F1xUvxcgVH7Y3JQ+2ZzTvtrdE36b3RN+m90
TfpvdE36b3RN+m90TfpvdE36b3RN+m90TfpvdE36b3RN+m90wgcCALPpAQCP/xsEnv8iDpP/KB
mJ/y0lgP8xMHb/NTtt/zIEZf8+TF//QIJZ/0ZXVf9KWIP/TI1Q/1FfTv9VYE3/WGFL/1tiSv9eY0n/YWRI/
2RkR/9pZUb/bWZG/21mRv9tZkb/bWZG/21mRv9tZkb/bWZG/21mRv9tZkb/bWZG/21m/5QLAP+X
HwT/mjES/5o+KP+WTkD/kFtX/45laf+NbXj6iXSG9IR8ku+AhZzre42k6XeUqedzm63lcaGv5XCose
RvrbLkb7Sy5HC7suRwxLLjccyy3HTOsdV30bHQedGw0HnRsNB50bDQedGw0HnRsNB50bDQed
Gw0HnRsNB50bDQedGw/5QLAP+XHwT/mjES/5o+KP+WTkD/kFtX/45laf+NbXj6iXSG9IR8ku+Ah
Zzre42k6XeUqedzm63lcaGv5XCoseRvrbLkb7Sy5HC7suRwxLLjccyy3HTOsdV30bHQedGw0HnR

sNB50bDQedGw0HnRsNB50bDQedGw0HnRsNB50bDQedGw/5QLAP+XHwT/mjES/5o+KP+WTKD/kFtX/45laf+NbXj6iXSG9IR8ku+AhZzre42k6XeUqedzm63lcaGv5XCoseRvrbLkb7Sy5HC7suRwxLLjccyy3HTOsdV30bHQedGw0HnRsNB50bDQedGw0HnRsNB50bDQedGw0HnRsNB50bDQedGw/5QLAP+XHwT/mjES/5o+KP+WTKD/kFtX/45laf+NbXj6iXSG9IR8ku+AhZzre42k6XeUqedzm63lcaGv5XCoseRvrbLkb7Sy5HC7suRwxLLjccyy3HTOsdV30bHQedGw0HnRsNB50bDQedGw0HnRsNB50bDQedGw/5QLAP+XHwT/mjES/5o+KP+WTKD/kFtX/45laf+NbXj6iXSG9IR8ku+AhZzre42k6XeUqedzm63lcaGv5XCoseRvrbLkb7Sy5HC7suRwxLLjccyy3HTOsdV30bHQedGw0HnRsNB50bDQedGw0HnRsNB50bDQedGw/5QLAP+YHwP/mzES/5s+J/+XTj//kVxX/5JlaP+RbXj6jnSG9lh6k++EhJ3rfoyl6HqTq+Z2mq/kc6Cy5HKns+NxrbTjcbS143G8teNyxrXfdMy02HfOtNF6z7PMe8+zzHvPs8x7z7PMe8+zzHvPs8x7z7PMe8+zzHvPs8x7z7PMe8+zzHvPs8x7z7PMe8+5UKAP+ZHwP/nDES/5w/J/+ZTj//IFxX/5dlZ/+WbHf6IHOF9I96ku6Kgp3qhlqm53+RreR6mLLjd5+14nWmt+F0rbjhdLW54XS+ueB1ybjZeMy40nzNt8t9zbfGfsy4xn7MuMZ+zLjGfsy4xn7MuMZ+zLjGfsy4xn7MuMZ+zLjGfsy4/5YKAP+aHgP/njER/54/J/+aTj//mFtV/5tkZv+cbHb7mnOE9JZ5ku6QgJ3qioim5oSPruN+lrThep244Helut92rbzfdra833bBvNt5ybzTfcq7zH/Lu8WAyrvBgMq8wYDKvMGAYrzBgMq8wYDKvMGAYrzBgMq8wYDKvMGAYrzBgMq8/5YJAP+bHgP/njER/58/Jv+cTz//m1tU/59kZP+gbHT7n3KD9Jt4ke+Wfp3qkian5omNr+ODILXgfZy633qkvd54rL/eeLe/3nnEv9R9yL/Ngcm+xoLlv8CCyL+8gsi/vILlv7yCyL+8gsi/vILlv7yCyL+8gsi/vILlv7yCyL+8gsi//5cJAP+cHgP/nzIR/6BAJv+dTz//nlpT/6NKY/+kbHP8pHKC9aF4j++cfpzqloSm5o+Mr+Klk7bfgZq83XyJwNx6rMLcernC2XzGws+Bx8Hlg8fBwYTGwruExsK3hMbCt4TGwreExsK3hMbCt4TGwreExsK3hMbCt4TGwreExsK3hMbC/5cJAP+cHgP/oDIR/6FAJv+eTz//oVpR/6ZkYf+obHH9qHKA9qZ4jvCjfprrnIKl5pWKruKNkbfhZi93H+hwt8rMXbfLvF04DFxcmFxcTChcTFvlbExbeGxMWzhsTFs4bExbOGxMWzhsTFs4bExbOGxMWzhsTFs4bExbOGxMWzhsTF/5glAP+dHgL/oTIQ/6JAJv+fTz7/pFpQ/6pkYP+sbG/+rHJ996t4i/Gpfpjro4Kj55ylreKUj7beipa+24KgxNp+rMjZf7/lzlXDx8OHw8i8iMLltojCybKlwsiviMPlr4jDyK+lw8iviMPlr4jDyK+lw8iviMPlr4jDyK+lw8iviMPl/5glAP+eHgL/ojIQ/6JBjF+gTz7/p1pO/61kXv6va239r297+a51ifKte5XtqoGh6KSFq+OcjBxekZS+2oaextiArcrUgsHLxYnByryKwcu2isDMsYrBy62KwcuqisHKqorByqqKwcqqisHKqorByqqKwcqqisHKqorByqqKwcqqisHK/5kHAP+eHgL/ojIQ/6NBjF+hTz7/qlpN/7BkXP2yaGr7sm14+rJyhfsxd5Lur32d6auDqOSlirLfmpK82oybxtaCrc3Lhr/OvYu/zrWMv86wjL/Orly/zqmMwM2mjMDMpozAzKaMwMymjMDMpozAzKaMwMymjMDMpozAzKaMwMymjMDM/5kHAP+fHgL/ozlQ/6RBJf+jTz3/rFpL/rJiWfy0Zmj6tWp1+LZvgva2dl7xtXmZ7LN/o+euha3go4261JSXx8iKps+/ib3Stl690a2OvdGpjr3Rp46+0KSOvs+ijb/Noo2/zaKNv82ijb/Noo2/zaKNv82ijb/Noo2/zaKNv82ijb/N/5oHAP+fHgL/ozlP/6VBJf+mTzz/r1pJ/bRgV/q3ZGX3uWhx9btsffS8cYjyvXeS7L59nOS5gKrYrIW5yJ6QxrmSntGvjrvPVqJC71aWQu9SikbzToZC90p+QvtCej7/Pno+/z56Pv8+ej7/Pno+/z56Pv8+ej7/Pno+/z56Pv8+ej7/P/5oGAP+gHgL/pDMP/6VCJf+oTjr/slpH+7ZfVPe6YmH0vWZt8cBreO7DclHrx3eK5sh5mNzEd6nOtn+4vaiKxqycltKelKjYmpS52JqUutaak7vVmpO805mSvdGZkb7QmZG+0JmRvtCZkb7QmZG+0JmRvtCZkb7QmZG+0JmRvtCZkb7Q/psGAP+gHgL/pTMP/6ZCJP+qTjn9tFtF+LheUvS9YV3wwWVo7MZpcejNcHjk03SE39dul9PQcKjEwXq2s7ODxaGnjtGOnZ7ZiZ642Y6ZudiQl7rWkpa71JOVvdKtk77Qk5O+0JOTvtCTk77Qk5O+0JOTvtCTk77Qk5O+0JOTvtCTk77Q/JsFAP+hHQL/pTMP/6dCJP+sTzf7tlpD9rtcTvHBX1nrX2Ni5s9oaODZcG3a4GuC1ONrlsnCbKa6znS1qb98w5ezhtCDqpbXfK222IOkuteHnrrWipu71lyYvNKNlr3QjZa90l2WvdCNlr3QjZa90l2WvdCNlr3QjZa90l2WvdCNlr3Q+5wFAP+iHQL/pjMP/6hDJP+vUTX5uFg/875aSuzGXFLlZ2FZ3ttoW9jibWvS526Ayuhul/fcKWY1nOzoM50wYzCfs18vZLUdsGy1HyzvNN/qbzUgqO804WevdGGm77Qhpu+0labvtCGm77Qhpu+0labvtCGm77Qhpu+0labvtCGm77Q+ZwEAP+iHQH/pzMO/6IDJP6zUjL2u1Y77sNXRObNWUrd2l5M1uJk

XdDoam/K7W+AwupykrjgdK0t2Hixm9B1vojKesi7x4vPecekznfHvc55uL7Pe62+0H2mvs9/ob/Pf6G/z3+hv89/ob/Pf6G/z3+hv89/ob/Pf6G/z3+hv89/ob/P9p0DAP+jHqH/qDQO/6pEI/u2VC7yv1Q26MITPN3YVT7V4ltPzulhY8juZnXB8GuGuehwlbHgdaOm2XewltJ1u4bMecR9yoflfMuayHzLrcd0yr/JdbzBy3exwMx4qsHLeKrBy3iqwct4qsHLeKrBy3iqwct4qsHLeKrBy3iqwct4qsHL854CAP+kHqH/qTQN/6xFl/e5USrtxE8v4NNNMdXhUT/O6VIVx/Bfab/0ZXq47mqKseZvmanfc6Wc2XGvj9NyuYLPdsB7zYHDes6Rw3rOocF4zq/Bcc2/xHDBxMVyt8TGcrfExnK3xMZyt8TGcrfExnK3xMZyt8TGcrfExnK3xMZyt8TG76ABAP+mHAH/qzQN/7BGH/O9TSPIzEgl1t9GLc3pUETG8Vhav/Zebbj1ZH6x7GqNquVump/fbKST2mytiNVutX7Sc7p40Xy9dtGlvXXRlrx20qO7dNKwu23QwL1ryci/a8nlv2vJyL9ryci/a8nlv2vJyL9ryci/a8nlv2vJyL9ryci/6aEAAP+nGwH/rTUM+rZIGuzERhvZ2zgbzudHMsXxUEm+91het/tfcLH0ZoCq7GqNoOZnmJXhZ6KL3GipgdIrr3rWcLN11Xe2ctSBt3HUJLZx1Ze2cdWjtXDWsbNr1b22a9W9tmvVvbZr1b22a9W9tmvVvbZr1b22a9W9tmvVvbZr1b224KMAAP+pGwD/sDYL9LxGEuDRNQ/P5TsfxvBINr35UE22/VlgsPxhcan0ZX+f7WOLlehiYvYp2D4GWje91oqHXbbatx2nKubtl6r2zZg69r2Yyva9qXrmvao61r266sa9uurGvbrqxr266sa9uurGvbrqxr266sa9uurGvbrqxr266s3KUAAP+sGQD9szcK6cY3CNLhKw7G7z4kvfHJOrb/U0+w/1xhqP1gcJ72XnyU8F2Hiutcj4HnXpZ65GGbdeJln3DgaaJs326kad90pWfee6Zm3oKmZt6LpmbflqVI36CkZd+gpGXfoKRI36CkZd+gpGXfoKRI36CkZd+gpGXfoKRI36Ck2qcAAP6wFgDzvDMD1twVAsfuMhK9+UEntf9LPK//Vk6m/1henP9XbJL5V3el9FeAf/BYh3jtWo1x6lyRbehglGnnZJdm5mmZZOVummLldJth5HqbYOSAm1/kiZtf5ZKaX+WSmI/lkppf5ZKaX+WSmI/lkppf5ZKaX+WSmI/lkppf5ZKa16oAAPq0EgDgzW4AyeshBb34NWX1/0Qorf9PO6P/T0uZ/09Zj/9PZYX+UG59+VF2dfZTfG/zVoBp8ViEZfBch2LvYllg7mSLXu1ojFzsb1a7HGNWex3jljsfo5Y7IWOWOyFjljshY5Y7IWOWOyFjljshY5Y7IWOWOyFjljshY5Y7IWO1K4AANI9BADK2QcAvfgrB7T/Oxar/0AnoP9DNpX/RESL/0VQgf9HW3n/SWNx/0xpa/1Obmb7UXJh+IR1XvhXd1v3WnlZ9157V/ZhfFb2ZX1U9WI9U/VufIL0dH5R9HI/UfR5f1H0eX9R9HI/UfR5f1H0eX9R9HI/UfR5f1H0eX9R9HI/0LIAAMfEBAC8+Q4Asv8qB6j/MRSc/zQikv83L4f/Ojt9/zxGdP8/Tmz/QIVm/0VbYf9IX1z/S2JZ/09IVv9SZIT/VWhT/1hpUf9bak//XmtO/2FsTf5lbUv+am1K/m9uSv5vbkr+b25K/m9uSv5vbkr+b25K/m9uSv5vbkr+b25K/m9uyLoBALrRAQCT/woApP8eBpn/lxGO/ygcg/8sJ3j/MDFv/zQ5Z/84QWH/PEhb/0BNV/9DUVP/R1RR/0pWT/9OWE3/UVIL/1RbSf9XXEj/W11G/15eRf9iX0T/Z19D/2xgQ/9sYEP/bGBD/2xgQ/9sYEP/bGBD/2xgQ/9sYEP/bGBD/2xgvMgAAK7cAACH/wcAlP8PBIn/Fgt+/x0Udf8jHmz/KCdk/y4vXf8yNlb/NzXR/zBTv8/REv/Q0dl/0dJRv9KS0T/TkxD/1FNQf9VTkD/WE8//1xQPv9gUDz/ZVE7/2pSO/9qUjv/all7/2pSO/9qUjv/all7/2pSO/9qUjv/all7/2pS/4gMAP+IGAP/iikQ/4c3lv+DSDf/fVVK/39eWf99Zmj/eG50/XN5gPhugon1aoyQ82aVlfFknJjwY6Sa72KrnO9ispzvYrmd72LCne9izZzrZNWc5WfYnN9q25vZbd6a027fmtNu35rTbt+a027fmtNu35rTbt+a027fmtNu35rTbt+a/4gMAP+IGAP/iikQ/4c3lv+DSDf/fVVK/39eWf99Zmj/eG50/XN5gPhugon1aoyQ82aVlfFknJjwY6Sa72KrnO9ispzvYrmd72LCne9izZzrZNWc5WfYnN9q25vZbd6a027fmtNu35rTbt+a027fmtNu35rTbt+a027fmtNu35rTbt+a/4gMAP+IGAP/iikQ/4c3lv+DSDf/fVVK/39eWf99Zmj/eG50/XN5gPhugon1aoyQ82aVlfFknJjwY6Sa72KrnO9ispzvYrmd72LCne9izZzrZNWc5WfYnN9q25vZbd6a027fmtNu35rTbt+a027fmtNu35rTbt+a027fmtNu35rTbt+a/4kMAP+JGAP/iykQ/4k3lv+ERzf/gVRK/4NdWf+BZWf/fW10/Xd2gPhyglr0bYqR8mmTI/Bmm5vvZaOd7mSqnu1ksZ/tZLqf7WTDn+1kz5/oZ9Sf4WrYnttt257UcN2dz3Hdns9x3Z7Pcd2ez3Hdns9x3Z7Pcd2ez3Hdns9x3Z7Pcd2e/4oLAP+KFwP/jCkP/4o3lv+GRzf/hlNI/4hcV/+HZGb/g2x0/X1zgPh4fYr0coeT8G6Qme5qmZ3taKGg7GapouxmsaPrZrqj7GbFo+tn0aPja9Si3G7YotVx26HOctqiyXPaoslz2qLJc9qiyXPaoslz2qLJc9qiyXPaoslz2qLJc9qi/4sLAP+LFwL/jSkP/4w3lv+IRzf/iIJH

/41bVv+NY2X/iWpz/YRyf/h9eorzd4SU8HKOm+1ul6Drap+j62iopeposKbqaLum6mjHpuZr0abdb9
Wl1nLZpc902KXlDdimxHXYpsR12KbEddimxHXYpsR12KbEddimxHXYpsR12KbEddim/4sKAP+M
FwL/jikP/403lf+JRzf/jlFF/5JaVP+SYmP/j2px/opwfvCelrzfIKU73aLnOxxlaLqbZ6m6WunqOlpsKnoa
byp6WrKqeFu0qnYcta0HXWqMI21anDd9Wpv3jVqr941aq/eNWqv3jVqr941aq/eNWqv3jVqr941a
q/eNWq/4wKAP+NFwL/jykO/443lf+LRzb/kIFE/5ZaU/+XYmH/IWlv/pBvffiKdonzgn6U73yJnOx1kqP
pcJuo6G2mq+drsKzna72s5m3OrNty0qvRdtSrynJUrMN506y9edOtunrTrbp60626etOtunrTrbp6062
6etOtunrTrbp60626etOt/40KAP+OFwL/kSkO/483lf+PRjX/IIBC/5tZUP+cYV//m2ht/5due/mRdlf0iXu
S74KFnOt6j6TodJmq5m+kreZtsK/lbcCv4HHOr9R20q7KedKvw3rRr7170bC4fNGwtXzRsLV80bC1
fNGwtXzRsLV80bC1fNGwtXzRsLV80bC1fNGw/40JAP+OFwL/kikO/5E3lf+SRjP/mk9A/59YTv+hY
F3/oGdr/51tePqYc4X1kXmQ74mCm+uAjKToeJar5XKisORvsLLkcMSy2XXQssx70LHDFnCyyH3P
s7d9z7Syfs+zr37Ps69+z7Ovfs+zr37Ps69+z7Ovfs+zr37Ps69+z7Ovfs+z/44JAP+PFgL/kioO/5l3IP+
VRTL/nk8+/6NYTP+mYFr/pmdn/6RsdfygcoH2mXeO8JF+meyHiKPnfZOs5HWgsuNxsLXhcsi10Hn
OtcR9zrW8fs22tn/Nt7GAzbetgM22qoDNtqqAzbaqgM22qoDNtqqAzbaqgM22qoDNtqqAzbaqgM22
/48JAP+QFgL/kyoN/5M3IP+YRTD/oU88/6dZSf+qYVf/q2hk/6ptcfncn32onaJ8Zp7leuQhKHkhY+r3
Hucs9V2q7jRdsC5xnzMubuAzLm0gcu6r4LLuquCy7qogsy5pYLMuKWCzLilgsy4pYLMuKWCzLilg
sy4pYLMuKWCzLilgsy4/48IAP+RFgH/ICoN/5Q3IP+bRC//pFA6/6paR/+uYIP8sGlg+69ubPaucnjxq
3eE66V7kOScgp3aj4uq0lOWtcd9pLvCfLa9u3/JvbKDyb2shMm9qITJvaWEyryihMq7oITLu6CEy7u
ghMu7oITLu6CEy7ughMu7oITLu6CEy7ughMu7/5AIAP+RFgH/ISoN/5U3IP+eRS3/p1E4/65bRPyy
Y1D6s2dc97RsZ/G0cXPrsnZ95a56i9ykfZzQmlapxYyRtbqEnb2zga7Br4PHwamGx8GkhsfAoYflwJ
+HyL+dh8m9m4fKvJuHyrybh8q8m4fKvJuHyrybh8q8m4fKvJuHyrybh8q8/5AIAP+SFgH/lion/5U4IP
+gRSv/qll2/bFcQfm1Ykz2t2ZX8rlqYe27b2vnu3R23rVxidOreJrHolGou5WLta+Mlr+mh6bEooi9xZ6
JxcSbisXDmonGwpiJx8GXici/lojJvpalyb6WiMm+lojJvpalyb6WiMm+lojJvpalyb6WiMm+/5EHAP+S
FgH/lion/5Y4H/+jRir/rVM0+rRdPva4YEjzu2RR7r9oWujDbmLhw2x1171qiMuzdJm/qH2nsp6GtKW
UkL+ajp/Gll60yJKOw8eRjcTGkY3FxJGMxsKRi8fBklvIv5CLyL+Qi8i/klvIv5CLyL+Qi8i/klvIv5CLyL+
Qi8i//5EHAP+TFgH/lyoM/5c4H/+lRyj/sFQx+LdbOvO7XkPuwWFL6cdnUuTmaV7bz2Bzz8ZnhsS7c
Je3sXimqqeAs5udir+NlpjHhpWtyoSWwsmGk8PHiJHEXymPxcSKjsbCi43HwluNx8CLjcfAi43Hwlu
Nx8CLjcfAi43HwluNx8CLjcfA/5IHAP+UFgH/mCoM/5k4Hv+oRyb8s1Uu9bpZnu/AWz3px19D49Fm
RtzZXlzT2VtxyM5jhLzEa5avunOkorB7spKng76CoJHHeZ+mynehwsl8m8PHf5fExoGUxcSDksbCh
JHHwlsRx8CEkcfAhJHHwlsRx8CEkcfAhJHHwlsRx8CEkcfA/5IGAP+VFQH/mSoM/5w5Hf+qSCP
5tlYq8b1WMerGWDbi0V042t1dQ9PiXlrL4V1wwNlfrTPZpSoxW6jmbt0sliyft5rYvEcKyhx22vv8Zz
p8XFdqDFxHmbxcN8mMbCfZXHwH2Vx8B9lcfAfZXHwH2Vx8B9lcfAfZXHwH2Vx8B9lcfA/5MGAP
+VFQH/mioM/585G/+uSSD2uVMm7cJTKuPOVCza3Vkt0+NgQczpYljD6WNtueNigKzbYpGf0migk
MlurYDBd7hxxvla/ar2ewWfAvMBstsAb6zHwXGlx8F0n8jAdp3lv3acyL92nMi/dp3lv3acyL92nMi/dp3l
v3acyL92nMi//5QFAP+XFQH/mysL/6M6GP2yShzyvU8g58lNlraTSDS41gyzOpgRMXvZIW98Wdr
suxnfabnZY6X4GOdhtloqnfScbNr0IG5ZdGbu2PTt7pnysy5abvLu2uxyrxtqcq8b6XKvG+lyrxvpcq8b
6XKvG+lyrxvpcq8b6XKvG+lyrxvpcq8/5UEAP+YFQD/nSsL/6c6Ffi3SxfsW0kY3dVDF9LiTCXL6lc3x
PFeSr33Y1y192dtrPVpfaDtZ4yR5WSagt9opXTab65r132yaNaSs2fYp7Jn2L+xY9DPs2XCzrVnt82
2abDNt2mwzbdpsM23abDNt2mwzbdpsM23abDNt2mwzbdpsM23/ZYDAP+ZFAD/nysK/6w7EPK8
RxHjzTwP0+E+F8rqTCrD8IY9vPhcULT9YWGt/GVxovRjf5XsYY2l5mGYfOFmoXLdbahr23iraNuJr
Gjbm6to3KyqZdy+qmDY06xhydKvY8DRsGPA0bBjwNGwY8DRsGPA0bBjwNGwY8DRsGPA0bBj
wNGw+ZgCAP+bEwD/oSsJ/LM6C+rFOwnV3S0Ky+pAG8LyTS+7+VVCtP9bVK3/YGSj+19zl/Regl
ztXYuA6F6VduRjnG7haqFp33OkZd+ApWTfjqV352kZOCro2TgvKJd39KIXdTXpl3U16Zd1NemXd

TXpl3U16Zd1NemXdTXpl3U16Zd1Nem85oAAP+eEgD/pioG87s2BNvVHQHM6DIOwvNDILv6TjO
z/1VFrP9bVqL/WmaX+1IzjfVZfoLwWYd461yPcOhglWrlZ5lI5G+cYuN4nWHjg55g45CdYOScnV/kq
Zxf5bmaXeXHm13lx5td5cebXeXHm13lx5td5cebXeXHm13lx5td5ceb6Z0AAP+hEAD/ryUC5sggAM
7lHwPC8jcRuvfJLP/Tzar/1NHof9TVpf/U2SM/VNvgvhUeXnzVoFw8FmHae1djGTrY49g6mmSXul
xk1zoepRb6ISUWuiPIFrmpNa6aeSWeqkXVnqsZFZ6rGRWeqkXVnqsZFZ6rGRWeqkXVnqsZFZ6r
GR3qAAAP+IDQDzug4A0NYIAMPxKQW6+zwUsv9lJqn/Szaf/0tGIP9MvIv/TWCB/05qePxQcXD4
Unhp9VZ9Y/NagV/yX4Rc8GSGWe9rh1fvcohW73qJVe+CiVTvjllU75ilU++giFPvolhT76CIU++giFP
volhT76CIU++giFPvolhT76CI26MAANusAADNwwUAW/EPALn7MAex/z4Wp/9AJZz/QDSS/0FCh/
9DTn7/RIh1/0hhbf9LZ2f/Tmxh/FJwXftWdFn5WnZX+F94VPdkeVL3anpR9nB7T/Z3e072f3xN9ol7T
faRe032kXtN9pF7TfaRe032kXtN9pF7TfaRe032kXtN9pF716cAAM22AQDAzAMAtv4UAK3/Kwik/z
IVmf80l47/Ni+E/zk7ev88RXH/P05p/0JVY/9GWI7/SV9Z/01iVv9RZFP/VWdR/1loT/9dak3/YmtL/2dr
Sf9tbEj/c21H/nttRv6BbUb+gW1G/oFtRv6BbUb+gW1G/oFtRv6BbUb+gW1G/oFt0K4AAMHCAwC0
2AAAqf8QAKD/IAeV/yYSiv8oHYD/LSh2/zEybf81OmX/OEFe/zxHWf9AS1T/Q09R/0dSTv9LVEz/TI
VK/1JXSP9WWEb/WIE/15aQ/9jW0H/aFtA/25cP/9zXD//c1w//3NcP/9zXD//c1w//3NcP/9zXD//c1w//
3Ncw7kBALbPAACo5wAAm/8LAZD/EwWF/xoNe/8fFnH/JB9n/ygnX/8tLlj/MTRT/zU4T/85O0z/PD5
J/0BARv9DQkT/RkRC/0pFQP9ORj//UUg9/1ZJPP9aSTr/Xko5/2RLOP9pTDj/aUw4/2IMOP9pTDj/a
Uw4/2IMOP9pTDj/aUw4/2IMuMcAAKnaAACb+gAAjf8CAYD/BgN0/wslav8TD2H/GhZa/yAdU/8ml0
7/KyhK/y8sRv80MEP/ODJA/zw0Pv9ANjz/Qzc6/0c5OP9KOjf/Tjs1/1M8NP9XPDL/XD0x/2l+L/9mPi
//Zj4v/2Y+L/9mPi//Zj4v/2Y+L/9mPi//Zj4v/2Y+/3sLaf95FwX/dSIO/3lzHv9sQzD/bU8+/25YS/9rYVj/Z
2xk/2J3bv9dgnb/WYx8/VaWgPxUn4L7VKeE+IOvhfpTt4X6U8CG+IPLhfhU2YX0VuCF7lrjhOhd5YT
iX+eD3GLpg9lj6oPZY+qD2WPqg9lj6oPZY+qD2WPqg9lj6oPZY+qD/3sLaf95FwX/dSIO/3lzHv9sQ
zD/bU8+/25YS/9rYVj/Z2xk/2J3bv9dgnb/WYx8/VaWgPxUn4L7VKeE+IOvhfpTt4X6U8CG+IPLhfhU
2YX0VuCF7lrjhOhd5YTiX+eD3GLpg9lj6oPZY+qD2WPqg9lj6oPZY+qD2WPqg9lj6oPZY+qD/3sLaf
95FwX/dSIO/3lzHv9sQzD/bU8+/25YS/9rYVj/Z2xk/2J3bv9dgnb/WYx8/VaWgPxUn4L7VKeE+IOvhf
pTt4X6U8CG+IPLhfhU2YX0VuCF7lrjhOhd5YTiX+eD3GLpg9lj6oPZY+qD2WPqg9lj6oPZY+qD2W
Pqg9lj6oPZY+qD/3wLaf96FgX/dyIO/3MzHv9uQzD/cU09/3JXS/9vYFj/a2pk/2Z1b/9gf3f+Xlp9/FiUg
vpXnoX6VqaH+VWuiPIvt4j5VcCl+FXMiPdW2ojxWd+l6lziH+Rf5YbeYueG12XphtRI6YfUZemH1G
Xph9RI6YfUZemH1GXph9RI6YfUZemH/30Laf97FgX/eCIN/3UzHv9wQi//dkw8/3hVSv92Xlf/cGdk/2
tyb/9lfHj9Yld/+1yShPIZnlj4WKS91eti/dXtoz3V8CM9lfNjPRY3lzsXd+L5WDiit5k5YrYZueK0Wfoi8
5o6lvOaOiLzmjoi85o6lvOaOiLzmjoi85o6lvOaOiL/34Laf98FgT/eilN/3cyHf91QS7/e0o7/31TSP97X
Fb/d2Vj/3Fvb/9qeXj9ZISA+mCPhvhcmYr2WqON9lqsj/VZtY/1WcGP9VrPj/Bc24/oYN+O4GTijtlo5Y
3SaeaOy2rlj8hr5Y/la+WPYgVlj8hr5Y/la+WPYgVlj8hr5Y/la+WP/38KAf99FgT/eylN/3gyHf95Py3/gE
k5/4JSR/+BW1T/fWNI/3Zsbv9wdnj8aYGB+WSMiPZfl431XaGQ9FyrkvRbtZP0W8KT81zSk+tg25Li
ZN+R2mjikdJq5JHMBOSSxW3jk8Nt45PDbeOTw23jk8Nt45PDbeOTw23jk8Nt45PDbeOT/4AKAf9
+FgT/fSIN/3oyHf99Piv/hUg3/4dRRf+HWVL/hGFg/31pbf92cnj8b32C+GiJifVjll/zYJ+T8l6qlfJdtZbyX
cOW8V7WluZk25XcaOCU02zilMtt4ZbFbuGXv2/hl71w4Je9cOCXvXDgl71w4Je9cOCXvXDgl71w4
Je9cOCX/4AKAf9/FgT/filM/3wyHf+CPSr/ikc1/41PQv+NWFD/il9e/4Vna/99b3b8dXmB+G6FivRnkJ
HyY5yW8WComPBftZnwYMa62LWmd9o3JjUbeCXy2/fmcRw35q/cd6buXLem7dz3pu3c96bt3Pe
m7dz3pu3c96bt3Pem7dz3pu3c96b/4EKAf+AFQP/fyIM/30yHf+GPCj/j0cz/5JQQP+TWE3/kV9b/41
maP+GbHT9fHR/+HSAivRtjJLxZpmY72Omm+9itZ3sY8ed5WfYnNht3ZvMcd6cxHLdnr1z3J64dNyf
s3Xcn7J13J+yddyfsnXcn7J13J+yddyfsnXcn7J13J+yddyf/4IJAP+BFQP/gSIM/34xHP+KOyb/k0cx/
5hRPf+ZWUr/mF9X/5VmZP2PbHH4hnJ99H18iO90iJLrbZWZ52minuNnsKDhackg3WrZoM5x3KD

EdNuhvHXaorZ22qKyd9qjrnfa0qx42qKseNqirHjaoqx42qKseNqirHjaoqx42qKseNqi/4MJAP+CFQP
/gilM/4AxHP+OPCT/l0gv/5xROv+fWUb/n2BT/JxmYPeYa23ykHF57ld5hed+hJDhdpCa22+coNZsq
qTSbbulz27SpcVz2aS7d9iltHjYpq952KaredimqHrYpqZ62KWmetilpnrYpaZ62KWmetilpnrYpaZ62
KWmetil/4QJAP+DFQP/gylL/4MwG/+RPCP/m0gs/6FSN/+kWkL9pWBO96NmW/GgbGfsmnB05pJ
3gN6lgl7Wfouaz3aWo8lypKfEcrKpwXPHqbp31amzetWprXvVqqh71amlfNWponzWqKF81qihfNao
oXzWqKF81qihfNaoXzWqKF81qihfNao/4QJAP+EFQP/hCIL/4YwGf+UPCH/nkgq/6VSNP+pWj74
qmFK8qpnVeyobGHmpHBt35t0fdaRf13Mh4aaxH6QpLx5naq3d6uttHi+rbB6062qfdOtpX7TraF+06y
ff9SsnH/Uq5t/1aqbf9Wqm3/Vqpt/1aqbf9Wqm3/Vqpt/1aqbf9Wq/4UIAP+FFAL/hSIL/4kwGP+XPR//
oUkn/6ITMPutWzr0sGJE7rFoT+iwbVnhq2xp2KJwe86ZeYzEj4KauoaLpLKAl6ysfaWwqH22saV/z7
GggNGwnYHRsJqB0q+YgdKuloHTrZaB06yWgdOsloHTrJaB06yWgdOsloHTrJaB06yWgdOs/4UI
AP+FFAL/hiML/4wwF/+aPR3/pEkl/6xTLfixXDbwtGI/6rZmR+O3aVPbsWVo0alsesagdYq8l36Zso6
GpKiHka2hg56znIKvtZmExrSWHm+0IITPs5OE0LGShNGwkYPSrpCD0q6Qg9KukIPSRpCD0q6Qg
9KukIPSRpCD0q6Qg9Ku/4YIAP+GFAL/hyML/44wFf+dPRv/p0ki/K9UKfS2XDHsuV845r1jP969YFH
UuGBmyq9peMCncom1n3qXqpaCpJ+Pi66WiZi1j4ept4yJv7eKic23iojOtYuHz7OKhtCyioBrsIqF0q
+KhdKvioXSr4qF0q+KhdKvioXSr4qF0q+KhdKv/4clAP+HFAL/iCMK/5EwFP+gPRn/qkkf+bNVJfC5
WSvpv1wx4sVeOdnGVU/OvI5kxLZmdrmuboeupnaWo55+o5eXhq6LkJK2g46juX+PuLi+j8y4gl3Nt
4KLzrWDis+zhlnQsYSI0bCEiNGwhljRsISi0bCEiNGwhljRsISi0bCEiNGw/4cHAP+IFAL/iSMK/5Qw
Ev+jPRf/rkoc9bdVley+ViXlx1ko3c9TN9PNuk3lxtivb1jdLK1aoWornGUnKd5oY6fga2BmYy1eJedu
XSXsrpzmMy5dpTnt3mRzrV7j8+0fY3Qsn2M0bF9jNGxfYzRsX2M0bF9jNGxfYzRsX2M0bF9jNGx/
4gHAP+JFAL/iiMK/5cwEP+mPRT7sUoY8rtRG+jFUR3g0VMf1ttHNczVT0vCzldgt8Zfcqy+ZoOgt22
SILB0n4apfKt4plezb6KYt2uirrhopM22bZ7OtXGZzrVzlc+zdZLQsnaR0bF2kdGxdpHRsXaR0bF2kd
GxdpHRsXaR0bF2kdGx/4kGAP+KEwH/iyMK/5owDv+pPRH3tkoT7cFLFeLOShTW3Ukdz+JOM8b
dT0m71IRdsNBbcKXIYYCZwWiPjLpunH20dqdwr4KvaK6Us2SvqrNiscmyZqrQsmmj0LJrntCxbpn
RsG+X0bBvl9Gwb5fRsG+X0bBvl9Gwb5fRsG+X0bBvl9Gw/4oGAP+LEwH/jSMJ/58vDP+uPA3yu0
gO5clBDdjCpW3P5E8ax+IVML/IV0a04VdaqNtYbZzUXH2QzWKMG8dpmXXCcaNpvn2qYr6QrV6/q
K1cwcasyLnUq2Kv061kqNOTZ6LTrWig061ooNOTaKDTTrWig061ooNOTaKDTTrWig061ooNOT/4sFA
P+MEgH/kCMI/6MuCPqzOgjrWz0H2dktBc/kQRDH7FAfwe9aL7jtXkKv6l9Xo+dcaZXhWXmH3F2le
dZjlG3TbJ1j0XmjXdGOpVrTpqVY1cSkW83Zo13A2KZettanYK3WqGGq1qhhtaoYarWqGGq1qhh
qtaoYarWqGGq1qhhtao/4wEAP+OEgH/ISEF/6ksBfO6NAPgzyUB0OMwB8ftRBS/81AkuPVZNbH
0Xkao819WnPFdZo/tW3WA6luCudfjmbLaJZe5XWbW+SJnFnkn5xY5babV+XYmVjU3ZxayNyfW7
zboVy426FcuNuhXLjboVy426FcuNuhXLjboVy426FcuNuh/44DAP+QEED/mx8C/a8lAenFIQDR4R
kBx+01Cr/1Rhm4+1EpsfxYOqj8WUqc+1dakfWalt3VnV39Fd/a+9ciGPrZI5d6XCSWumAlFnpk5N
Y6aWSWOq5kVbq1JFV3t+UVs/hllfK4JdXyuCXV8rgl1fK4JdXyuCXV8rgl1fK4JdXyuCX/5ACAP+S
EAD/ohgA9LgSANLRCADH7CMCvvU6Drf8SB2w/1ltp/9TPZz/UU2R/1Bbhf9QZ3r8UnJv91V6ZvNa
gV/wYYZa7muJV+13i1XthYxV7ZaLVO6milPvtolT782IUujhiVLh44tS4eOLUuHji1Lh44tS4eOLUuHj
i1Lh44tS4eOL+pMAAP+WDgDjqwUA0cEGAMjYCAC+9i0Etv0/Ea//SSCm/0swm/9KP5D/SUyF/0IZ
e/9LY3H/Tmxo+1JzYfhXeFv2XnxX9GZ/VPNvgVHzeoJQ84eCT/OWgU/zpIFO9LKATfTHfk311H5
N9dR+TfXUfk311H5N9dR+TfXUfk311H5N9dR+8JYAAO6fAgDQswlAxcceALz4EQC1/zAHrf88E6
P/QCKZ/0Ewjv9BPYP/QUI5/0RUcP9HXWj/S2Rh/09pW/5Ublb8WnFT+2BzUPpodU35cXZL+Xt3S
vmHd0n5k3ZJ+aB2SPqudUj6tnVI+rZ1SPq2dUj6tnVI+rZ1SPq2dUj6tnVI+rZ135sAANOqAADFvAl
AutEDALD/FQCp/ysloP8zFJb/NiGM/zctgf84Oxb/O0Nt/z5MZv9CU1//RIIZ/0tdVf9QYVH/VWNO/1t
mS/9hZ0n/aGhH/3BpRf95akT/g2pC/45qQv+caUL/omlC/6JpQv+iaUL/omlC/6JpQv+iaUL/omlC/6Jp

2KIAAMe0AAC7yAEArtoAAKT/EwGc/yllk/8pEoj/LB19/y4oc/8xMmr/NTpi/zIBW/89R1X/QUtR/0VPT
v9KUkv/T1RI/1RWRf9ZWEP/X1IB/2ZaP/9tWj7/dVs8/31bO/+JWzr/j1s6/49bOv+PWzr/j1s6/49bOv+
PWzr/j1s6/49byq0AALzAAQCv0wAAo/EBAJf/DgGN/xgGg/8eDnn/lhhu/yUhZf8pKF3/Li9W/zl1Uf83
OU3/Oz1J/z9ARv9DQkP/R0RB/0tFP/9QRz3/VUg7/1tJO9hSjf/Z0o1/25LNP93SzP/fEsz/3xLM/98S
zP/fEsz/3xLM/98SzP/fEsz/3xLvrkBALHMAACj2wAAIf0AAIn/BgF9/wsEcv8SCmj/FxFf/xwYV/8iHID/
JiNM/ysnR/8vK0T/My1B/zcwPv87MTz/PzM6/0M0N/9GNTX/Szcz/1A3Mv9VODD/Wjku/2A6LP9nO
iv/azor/2s6K/9rOiv/azor/2s6K/9rOiv/azor/2s6s8YAAKXWAACV4QAAh/8AAHr/AAFu/wIEYv8GB1j/
CwtP/xEOSv8XE0X/HhdB/ylaPv8nHTv/KyA4/y8iNv8zIzT/NiUx/zomL/8+Jy3/Qigr/0cpKv9MKij/USs
n/1crJf9eLCT/YSwk/2EsJP9hLCT/YSwk/2EsJP9hLCT/YSwk/2Es/2wMA/9nGAj/YCIO/1oxGv9WQ
Sf/XEoz/1xUP/9ZYEr/VWtU/1B3Xf9Mg2P/SY1o/0eXa/9HoG3/Rqlu/0axbv9Gu2//RsVv/0bSb/9G4G
7/R+xu+UvubvNP8G3tUvJs51X0bOBY9WzgWPVs4Fj1bOBY9WzgWPVs4Fj1bOBY9WzgWPVs/2
wMA/9nGAj/YCIO/1oxGv9WQSf/XEoz/1xUP/9ZYEr/VWtU/1B3Xf9Mg2P/SY1o/0eXa/9HoG3/Rqlu/
0axbv9Gu2//RsVv/0bSb/9G4G7/R+xu+UvubvNP8G3tUvJs51X0bOBY9WzgWPVs4Fj1bOBY9Wzg
WPVs4Fj1bOBY9WzgWPVs/2wMA/9oFwf/YsIO/1sxGv9aPyf/X0ky/19TPv9dXkr/WGIV/1N0Xv9Pg
WT/S4xp/0mWbf9In2//SKhw/0excf9HunH/R8Vx/0fScf9I4nH9Sutw9k7tcO9S8G/pVfJv41jzbtxa9G/b
WvRv21r0b9ta9G/bWvRv21r0b9ta9G/bWvRv/20MA/9pFwf/YyIN/10xGv9fPSb/ZUcx/2VQPv9iW0r/
XmZV/1hxXv9TfWb/Tols/0uVb/9KnnL/Sqdz/0mwdP9JunT/ScV0/0nUdP9K5XT5Tep08VLtc+pV73
LjWfFy3VvyctVc83TVXfN01V3zdNVd83TVXfN01V3zdNVd83TVXfN0/24LA/9rFgf/ZCEN/18wGf9j
OyX/aUQw/2tOPf9oWEn/Y2NU/11uX/9Xemf/UoZt/06Scv9MnXT/S6Z2/0uud/9Lunf/S8d4/kvYd/xM
53f0Uep27FXsduRZ73XeXPf11I7xds9f8njPX/N4z1/zeM9f83jPX/N4z1/zeM9f83jPX/N4/3ALAv9sF
gf/ZiEN/2AwGf9oOST/b0lv/3BMO/9uVkj/aF9U/2NrX/9cd2j/VoNv/1KPdP9Pm3f/TaV5/02vev9Nunv
+Tch7/U3be/dQ5nrv/Vel65Insed5d73jWX/B60GDxe8li8XzIYvF8yGLxfMhi8XzIYvF8yGLxfMhi8XzIY
vF8/3EKA9v9tFQb/ZyAM/2lvGf9tOCP/dEAt/3dKO91U0b/cF1S/2lnXv9ic2j/W39w/1aLdv9SmHr+UK
N8/k+ufv5Pu379T8t++0/ffvJU5X7oWel9317sfNzh7n3PYu9+yWTvf8JI7oDCZe6AwmXugMJI7oDC
Ze6AwmXugMJI7oDCZe6A/3IKAv9vFQb/aSAM/2YuGP9zNiH/ekAr/35KN/99U0P/eFtQ/3FkXP9qb
mf/YXpx/1qHeP5VIH39U6CA+1KsgflSulL3U8eC81XbguxZ5YHhXumA2GLsgM5k7YLHZu2DwWf
shLxo7IW7aOyFu2jshbto7IW7aOyFu2jshbto7IW7aOyF/3MJAv9wFAB/ah8M/2stF/94NR//gD8p/4R
KNP+EUKD/gVpN/3tiWv9ya2b9anZw+mOCePddj3/0WZuD8Vinhe9Ys4btWcKG61vVheVe5oTZZ
OqDzmrhsZo6oe/aeqlumrpibVr6Ym1a+mJtWvpibVr6Ym1a+mJtWvpibVr6Ym1a+mJ/3QJAv9xFA
X/bB8L/28rFv99NR3/hj8m/4tJMf+LUj3/iVIK/4RhVvp9aGP2dHJU8mx9eO5liYDqYJaF516iiOVeroni
X72J4GHQidti5onPaOmJxWroi71r54y4bOeNs23mja9u5o2ubuaNrm7mja5u5o2ubuaNrm7mja5u5
o2ubuaN/3UJAf9yFAX/bR4L/3QpFP+CNBv/iz8k/5BJLv+SUjn+kVIF+Y1gUvSHZ1/vfm5r6nZ5duVu
hIDgaJCH3GSdjNhjqo7VY7iP0mXKj85n4Y/Fa+aOvG3IkLVv5JGwcOSSrHHkkqh5JGoceSRqHH
kkahx5JGoceSRqHHkkahx5JGoceSR/3YJAf90FAX/bh4L/3gpE/+HNBn/kD8h/5VKKv+YUjX6mFIA
9JVgTe6QZInoiWxm4oF2c9x3gH/Vb4uJz2uXj8tpo5PHabCUxGrBIMJr2JS7b+OUs3Hila1y4papc+
KWpnTilaJ04pWidOKVonTilaJ04pWidOKVOnTilaJ04pWidOKV/3clAf91EwX/cB4K/3wpEf+KNBf/ID
8e/5pJJ/yeUjD1nlk7751fR+mZZVPik2xg24lzcNKAfH7LeIWKxHKRkr9vnZa7bqqZuG+5mbVwzpmx
cuCZqnTfmaZ135qidt+Zn3fgmZx34Jicd+CYnHfgmJx34Jicd+CYnHfgmJx34Jicd+CY/3glAf92EwT/
cR0K/38pEP+ONBX/mD8b/55JI/ijUizxpVk16qRfQOSiZEvcmmhe05Fwb8qleH3CglGKunqLk7V1I5
mwdKSdrXSynqp1xp6nd92donjdnZ553Z2bed2cmXnenJd535qWed+alnnfmpZ535qWed+alnnfmp
Z535qWed+a/3glAf93EwT/ch0K/4MoDv+RNBP/mz8Z/aJJIPSnUifsqlkv5qtfON6oX0jVoGRczJhtbc
OQdXy6iH2JsoGGIKt8kZymeZ6gonmsop96vqKce9ihmXzboZZ826CUfNyfknzcnfF83Z2RfN2ckXz

dnJF83ZyRfN2ckXzdnJF83ZyRfN2c/3kIAf93EwT/cx0K/4YoDf+UNBH/nz8W+qZJHPGsUSLpsFgp
4rJdMdmTWEbQpmFaxp9qa72Xcnuzj3qlq4iClKODjJ2cf5iil36mpZR+t6WRf9Clj3/ZpI1/2qOMf9qijH
7bolt+3J6Lftyei37cnot+3J6Lftyei37cnot+3J6Lftye/3oIAf94EwT/dB0J/4goDP+XMw//oj4T9qplGO2x
UR3Itlgi3bhUL9OzVUTKrF9YwKVnabedb3mtlneHpJB+IJuKh52ShZKkjlOgqliDsaiFhMmohIPXp4S
D2KWFgtmjYHaoowWB26CFgNughYDboIWA26CFgNughYDboIWA26CFgNug/3sHAF95EwT/dx0I/
4snCv+aMw3+pT0Q865HFOM2UBfivFQa2L9JLs64U0LEsVxWu6tlZ7GkbHennnOGnZd6k5ORgp
2Ji42lgYmbqX2lrKp6isOqeYnVqXuH16d9htilfoXZo36E26F/hNuhf4TboX+E26F/hNuhf4TboX+E26
F/hNuh/3sHAF96EwP/ehwH/44nCf+dMgv7qTwN77NGD+a7TBHdxEGY08RFLMm+UEC/uFITtbFh
ZauraXWhpW+Elp92kYuZfpyAk4ild5CWqnKQp6tvkb2qb5HVqXKO1qd0i9imdonZpHiH2qJ4h9qielf
aoniH2qJ4h9qielfaoniH2qJ4h9qi/3wHAF97EgP/fhwG/5lmb/+hMAj3rToJ67hDCuHDQwrYzzYWzct
DKcPFTT65v1ZRR7leY6WyZXOarWuBj6dyj4OheZp3nlOjbpqRqGmapKpmm7mpZZvWp2mW16Z
sktilbo/ZpHGM2qJxjNqicYzaonGM2qJxjNqicYzaonGM2qJxjNqi/30HAP98EgP/ghsF/5UIBf+lLgXysj
YG5r88BdzPMgTR1TMUx9JAJ73NSjuzx1NOqcFaYJ67YXCTtWd+iLBtjHurdZdwp3+fZ6WNpGKlo
KZfpralXqfYpGGg2aNkm9mjZ5faommT26FpktuhaZLboWmS26FpktuhaZLboWmS26Fpktuh/34GA
P9+EgP/hhoD/5kiA/ypKgPsuC8C38knAdTcJAPK3TYRwdo+JLfVRzis0E9LostWXZfGXG2Mv2J7g
LpoiHO2cJJos3qaYbKJn1yynaBas7OfWLXUnVut3Z5eptyfYKDcn2Kb3J9im9yeYpvcnmKb3J5im9
yeYpvcnmKb3J5im9ye/4AGAP9/EgL/ixgC/54eAfWvlgDkwRsA1NgKAMvjLgPD5UEPuuNIIbDgSzS
l20xHmtZRWY/RV2iDzF12d8djgmVea4xiwnaTW8GGI1jCm5hWw7GXVMTSIVa84ZZYs+CYWqzg
mVyl35pcpd+aXKXfmlyl35pcpd+aXKXfmlyl35pcpd+a/4EFAP+BEQL/kRUB/6QVAO22DgDTywgAy
+UWAMLqNAa77EYQtOtRHqvqVTCg51NCIONSVIffU2N621hxbthefGTVZoVb1HKKVtSDjVPVmy
5R1q+NUNjQi1HR54tTxOaOVbrlfFay5JJWseSSvrHkkIax5JJWseSSvrHkkIax5JJWseSS/4MEAP
+EEQH/lw8A2qsEANK9BgDJ0AcAwewiAbrxOgmz8kkVrPJPI6LyUjOY8FJCjO5SUyDsU19z6lVqZ
+hadF7nY3tX5m+AU+d/gIHok4JQ6qmBT+rCgE3q5n9O2umCUM7rhVHC6ohRweqlUcHqiFHB6o
hRweqlUcHqiFHB6ohRweql/4YDAP+LDQDmoAMA0bADAMfCBAC/1ggAufQqArL4PQyq+kUZovp
JJ5f5SzaN+EtEgfdMUXX2Tlxq9IFmYPVWblj1XnNS9Wh3TfZ3eUv3iXIJ+J55SPizd0f50HZJ9OI1S
+bseEzZ7ntM1+57TNfue0zX7ntM1+57TNfue0zX7ntM1+57/4kBAPuUBADUqAAAYLgDAL7KBAC
17AwAr/wrBKj/OQ+g/z8clv9CKYz/RDeB/0VDdv9GTmv/SVdh/0xfWf9SZVL/WWpO/mJtSv1ub0f9f
HBF/Y1wRf2fb0T+sm5D/sltQv7nbEby72xH8e9sR/HvbEfx72xH8e9sR/HvbEfx72xH8e9s/4wAANm
fAADKsAAAv8ECALTSBACr/xMApf8pBp3/MxCU/zgdiv86KX//PDV0/z4/av9ASGH/Q1BZ/0dWU/9
NW07/VF5K/1xhRv9IY0P/cGRB/3IIP/+NZT//nmQ+/61kPf/BYzz/32l8/+JiPP/iYjz/4ml8/+JiPP/iYjz/4
ml8/+Ji4JUAAM2pAADAuQEAtcsBAKnbAQCG/xUBmP8jBpD/KxCG/y8bfP8yJnL/NTBo/zc4X/86Q
Ff/PkZR/0NKTf9ITkj/T1FF/1VTQf9dVT//Zlc8/3BYOv98WDj/ilg3/5hYN/+nVzb/uVc2/7xXNV+8Vzb/v
Fc2/7xXNV+8Vzb/vFc2/7xX0aEAAMGzAAC2xAAAqtQAAJ7xAwCU/xEBi/8bBoL/Ig54/ycXbv8qIGT
/LShb/zEvU/80NU7/ODIJ/z09Rv9DQEL/SEM//05FPP9URjn/W0c3/2RJNP9tSTL/eEow/4NKL/+QSi
7/oEou/6JKLv+iSi7/okou/6JKLv+iSi7/okou/6JKxK0AALi+AAcrzWAAndsAAJH7AQCG/woBe/8RBH
H/GApo/x0SX/8hGVb/JR9P/ykkSv8tKUX/MixB/zYvPv87MTv/QDM4/0U1Nf9KNjP/UTgw/1g5Lv9fO
Sz/aDoq/3E7KP96Oyb/iDsm/4k7Jv+JOyb/iTsm/4k7Jv+JOyb/iTsm/4k7ubgAAKzKAACe1wAAj+EA
AIP/AAB3/wIBbP8IBGL/DQhY/xIMUP8XEEn/HBVE/yAZP/8IHDz/Kh85/y4hNv8ylzP/NiQw/zomLv8/
Jyz/RCgp/0opJ/9QKiX/Vyoy/14rlv9mKyD/cCwg/3EsIP9xLCD/cSwg/3EsIP9xLCD/cSwg/3Esr8QAA
KDTAACQ3QAAgOUAAHT/AABo/wABXf8ABFP/AwdK/wgKRP8ODT7/Ew86/xgRNv8dEzL/IRQw/
yQVLf8oFiv/Kxco/y4XJv8yGCT/Nhki/zsaIP9AGx7/Rhsc/0scG/9SHBn/Wh0Z/1sdGf9bHRn/Wx0Z/1
sdGf9bHRn/Wx0Z/1sd/1wRBf9VGwr/TSYQ/0lxFv9FOx7/SkYo/0pSM/9HXDz/Q2hF/0B0S/88gVD/

Oo5T/zmZVv85pFf/OKxY/zi2WP84wlj/OM5Y/zjcWP846lj/OPVY/zz6V/9A/Ff5Q/1W80b/Ve5J/1XpS
/9W6Uv/VuL/1bpS/9W6Uv/VuL/1bpS/9W/1wRBf9WGwr/TiUQ/0MxFv9IOh7/TEUo/01RM/9KXD3/
RmZF/0JyTP8+f1H/PlxV/zqYV/86oln/Oqxa/zm2Wv85wVr/Oc1a/zndWv8561r/Ovda/z75WfxC+1j2
Rf1Y8En/V+tL/1jmTf9Z5k3/WeZN/1nmTf9Z5k3/WeZN/1nmTf9Z/14QBf9YGgr/TyUQ/0UwFv9MOR
7/UUMo/1JOMv9PWj3/SmZG/0VwTv9BfFP/PolX/z2WWv88oVz/O6pd/zu1Xf87wV3/O85d/zveXf87
7V3/PfZc/kL4XPhG+lvxSfxb603+WuVP/lzgUP5d4FD+XeBQ/l3gUP5d4FD+XeBQ/l3gUP5d/18QBf
9ZGgn/UCUP/0cwFv9QOB3/VkEn/1ZLMv9UVz3/T2NG/0lvT/9FeVX/QYZa/z+TXf8+n17/Palf/z20Y
P89wGD/Pc5g/z3fYP8+7GD/QPVf+kX3X/NJ+l7sTfxd5VD8Xt9S/WDZU/1g2VP9YNIT/WDZU/1g2V
P9YNIT/WDZU/1g/2APBP9aGQn/UiQP/0suFf9VNRz/Wz4m/1xJMf9ZVDz/VF9G/05sT/9JeFb/RINc
/0GQX/9AnWH/P6hj/0CyY/9AvmP/Qcpj/0LbY/9D6WP9RPRi9Un3Yu1N+WHmUvth3IT7Y9dV/GT
SVvxl0lb8ZdJW/GXSVvxl0lb8ZdJW/GXSVvxl/2EOBP9bGQn/UyQP/1AsFP9bMxv/Yjwl/2RHL/9hU
Tv/XFG/1ZoUP9PdVj/SoFe/0aMYv9FmWX/RKRm/0WuZ/9Fumf/RsZn/UfWZ/pl5mb4SfNm7072Z
eZT+WTdVvIm1Vf6aM5Z+2nJWvtpyVr7acla+2nJWvtpyVr7acla+2nJWvtp/2IOBP9dGAj/VSMO/1U
qEv9hMBr/aTsj/2xFLf9qTzn/ZVIE/15kT/9XcFj/UX1f/02JZf9LIGj+SqBq/Eqqa/tKtWv5S8Fr9kzRavN
O42rwT/Fp51T2aN1X92rTWvhszFv5bcZc+m7CXfluwl35bsJd+W7CXfluwl35bsJd+W7CXflu/2MNB
P9eGAj/VyIO/1snEf9oMBj/cDoh/3REK/9zTjb/bldC/2dhTf9gbFj9Wnhg+ISFZ/hRkGv2UJtt80+mbvJ
QsW/wUb1v7ILNbupT4G3mVfBt3Vn1bdJb9m/KXfdxw1/3cr5g93O6YfZzumH2c7ph9nO6YfZzumH
2c7ph9nO6YfZz/2UMA/9gFwj/WSIO/2AID/9vLxb/dzke/3tDKP97TTP/eFU//nFeS/pqaFb2Y3Ng8lx/a
O9Yi23sVpdw6lWicudWrXPIV7ly41jJcuBZ3XHbW+5x0l30c8hg9XXAYfR2u2P0d7Zk9HiyZfR4smX
0eLJI9HiyZfR4smX0eLJI9HiyZfR4/2YMA/9hFwf/WiEN/2YkDv91LhT/fjgb/4JCJP+DSy/9gVQ6+Htc
R/N0ZFPubG9e6WV6aOVghm/iXZJz3luddtxbqXfZW7V31IzEeNNd2HjPX+x3x2Hzeb5k8nq3ZfF7s
mbxfK5n8XyraPF8q2jxfKto8XyraPF8q2jxfKto8XyraPF8/2cLA/9jFgf/XCEN/2wjDP96LRH/gzgY/4hC
lf6KSyv4iVM28oVaQuX+Yk7md2xa4W92ZttogW/WY4x20mGYes5go3zLYK99yWG8fsZiz37DY+V
+vGbWfrVo74Cvae6Aq2rugadr7oCla+6ApWvugKVr7oCla+6ApWvugKVr7oCla+6A/2gLA/9kFgf/YB
8M/3AiC/9/LQ//iTcV/45BHfqRSibzkVlw7l5ZPOaIYEjfgmpV2HhyZNFxfHDLA4Z5xmeRfsJmnYG/Za
iDvGa1hLpnxoS3aN2Dsmrsg6xs7lWnbeyFo27shaBu7lSeb+yEnm/shJ5v7lSeb+yEnm/shJ5v7lSe
b+yE/2kKA/9lFQf/ZB0L/3Uicf+DLA3/jTYS/pNAGfWXSSHumFEq55ZYNeCSXkHYiWZT0lFvY8h5
eHDCc4F6vG6LgbhsloW0a6KlSvuvia5sv4msbdSJqW7piaNw6YmfcemJnHHpiZlx6oiYceqHmHH
qh5hx6oeYceqHmHHqh5hx6oeYceqH/2oKAv9mFQb/ZxsK/3ghCP+HKwv/kTUQ+pg/FfGcSBzpnIA
k4p5XLduYWz/RkGNRyYlSycGBdG+5enx6s3WGg65ykYmqcJ2Mp3CpjRwul6hccyOn3LmjZpz5
42XdOeNIXTnjJN06luSdOmKknTpipJ06YqSdOmKknTpipJ06YqSdOmK/2sJAv9nFAB/axoJ/3wgB/
+LKgn/ITQN95w+Eu2hRhflpE4e3qRRKtWeVz3MI2FPw49pYLqIcW6ygnl6q3yBhKV4i4ugdpePnX
Wkkpl1spKXdcWSIHbhkpF35JGPd+WQjnfjmj4x25o2MdueMjHbnjlx254yMdueMjHbnjlx254yMdue
M/2wJAv9oFAB/bhol/38gBv+OKQf+mTMK86E8DuqmRBLiqkwX2alKKM+jVTvGnV9NvZZnXrSPb
2ysiXZ5pIN9hJ1+hoyXe5KSk3mflY95rZaMecCWinrblYh64pSHeuOTh3nkkYZ55ZCGeeaOhnnmjo
Z55o6GeeaOhnnmjoZ55o6GeeaO/2wJAv9pFAB/cRkH/4lfBf+RKAb7nDEI76U6CuarQg3esEQT1K
1HJsQoUznBolxLuJxlXK+WbGumj3N4nop6g5aEgo2PglyUiX6ZmIV+qJmCfrqZf3/Umh9+4Zd/fuKV
f33jk3985JKAe+WQgHvIkIB75ZCAe+WQgHvIkIB75ZCAe+WQ/20IAv9qEwb/dRkG/4YeA+UJgT3
oC8F7Kk2B+KxPQjZtjkSz7JFJMWtUDa8qFpJs6JiWqqcaWmhlnB2mJB3go+Lfo2Gh4iVf4SUmXu
DpJt3hLWbdYXPmnWE35l3guGXelHilXmA45N5f+SSeX/kkn/5JJ5f+SSeX/kkn/5JJ5f+SS/24IAv9r
EwX/eBgF/4kcAv+XJAL0pCsD6K4xA963NATUujUQyrdDIsCzTjS3rVdGrqhfV6WjZ2ebnW10kZdz
gYiSeox+joOUdouQmnGKn5xti7Gca4vJm2uL35puiOCYclbhlNGE4pRzguOSc4LjknOC45JzguOS

c4LjknOC45JzguOS/28HAv9sEgX/fBcD/4wbAf6bIAHvqCUB47MnAdnAIQLOwDIOxb1AILu5SzKy
s1REqK9cVZ+qY2SVpGpyi59wfoCad4l2ln+SbpSLmGiTm5tlk62bY5TFmmKT4Jhlj+GXalzhlmqJ4
pRsh+OTblfjk2yH45Nsh+OTblfjk2yH45Nsh+OT/3AHAf9tEgX/gBYC/5AYAPqfGgDqrBsA3bsUANL
GHAHlxi8Mv8Q9HbXASC+su1FBorZZUZmxX2GPrGZvhKdse3mjc4ZvoHuPZp6HIWGdmJdenqq
XXJ7Cllue4pVemOKVYJTilGOQ45NkjeSSZI3kkmsN5JJkjeSSZI3kkmsN5JJkjeSS/3EGAf9vEQT/
hBQB/5QUAPsJdWdYswgA0sIIAMvMFwDCzisKucs5Gq/IRCymw009nL9UTpK6W12HtWfRfbFnd
3KtboJoqneKYKmEj1uplZFYqaiRVqq/kFaq4o5Xo+WQWp7lkFyZ5ZBeleWPXpXlj16V5Y9eleWPX
pXlj16V5Y9eleWP/3MFAf90EAP/iREA+JkMANipBQDQtwYAysYGAMPUEQC61iclstQ1F6nRQCif
zkk5lclQSorEVImAwF1mdbxjcmq5anxht3ODWraAh1a2kolUt6aJUre9iFG34YZSsOqIVKnpiVak6lp
Xn+iKV5/oilef6lpXn+iKV5/oilef6lpXn+iK/3UFAf97DQH/jwsA2qACANCuBADlvAQAwcwGALrcEgC
04C4Fq985E6LdPiSX2UU0jdVLRILRUVN3zVdgbMpea2PIZXRBxbN6VcZ9fILGkH9Qx6R/Tsi7fU3I
33xNwfB+T7jvgFCw7YJSq+2DUqvtg1Kr7YNSq+2DUqvtg1Kr7YNSq+2D/3cEAF+CCgDplglA0qYB
AMizAwDAwgMAuNIHALLIHQCs6TQGpelBEZzoRx+S5UkvhuJKPnrfTUxv3VJZZNtZY1zZYWtV2
WxwUNI6c07ajXNM26JzS9y4ckrc23FI2PNySsv0dUvB9HdMuvN5TLrZeUy683IMuvN5TLrZeUy68
3IMuvN5/3oDAP+KAwDWnQAAYqwbAMG6AgC4ygQAr9kHAKntlwKj8TQJm/E9FZLwQiKI70Yvfe
5IPHLsS0hn609SXepUWlbyXWFR6mhlTet2Z0vsh2dK7ZtnSe2vZUjuymRG7etkROP3Z0bZ92pHz
/hsR8/4bEfP+GxHz/hsR8/4bEfP+GxHz/hs/30CAN2TAADNpQAawrMBALjCAgCu0QQApvIQAKD2
JQOZ+TEMkfk4F4f5PCN9+UAvcvhDOmj4RkRe90pMV/dQUID4WFdL+GFbR/htXUT5e15C+o5eQ
fuiXUD7t1w/+9NbPvrwWkDx+ltB6PxdQej8XUHo/F1B6PxdQej8XUHo/F1B6Pxd9lgAANKdAADER
QAaubwBAK/LAQck2AMAnPwUAJX/IgSN/ywNhP8yGHR/NiJw/zosZ/89NV7/QT1W/0VDUP9LSEr/
UkxF/1pPQf9kUT7/cFM7/39TOP+RUzj/plM3/7pSNv/YUTX/8FA2//1PNv/9Tzb//U82//1PNv/9Tzb//U
82//1P2ZUAAMenAAC7tgAAsMYAAKTTAACZ5AIAkP8TAYj/HQWA/yUNd/8rFm3/Lx9k/zMnW/83L
IT/OzRO/0A5Sf9FPUP/S0A//1JCO/9aRdf/ZEY0/3BGMf9/Ry7/kUcu/6RHLf+4Riz/z0Yr/+pFK//qRSv
/6kUr/+pFK//qRSv/6kUr/+pFyqEAAL2xAACxwQAAPc4AAJjaAACM9AIAg/8OAXn/FgRx/x0KaP8jEI
//JxIX/ysfUf8wJUv/NCIF/zktQf8+MD3/QzI4/0k0Nf9QNjH/WDct/2I4Kv9uOSf/fDol/4w6JP+cOiP/rTkj/
785I//+OSP/vzkj/785I//+OSP/vzkj/785v60AALO8AACmywAAmNYAAIrfAAB/+wAAdf8HAWv/DgRi/
xQIWP8ZDFH/HhFL/yMWRf8nGkH/LB48/zAgOf81IzT/OSUx/z4mLf9EKCr/Sykn/1MqJP9cKyH/Zys
f/3MsHf+ALBv/jiwb/5ssG/+bLBv/mywb/5ssG/+bLBv/mywb/5sstbcAAKjHAACa0wAAi9wAAHzkAA
Bw/wAAZv8AAVz/BgRT/wsHTP8RCkX/FQ0//xoQOv8eEjb/lhQy/yYVL/8qFiv/Lhco/zIZJf82GiL/PBs
g/0McHf9KHRv/Ux0Z/10eF/9mHhX/cR4U/3ofFP96HxT/eh8U/3ofFP96HxT/eh8U/3ofq8MAAJzQA
ACM2gAAfOIAAGzpAABh/wAAV/8AAU7/AARG/wEGP/8GCDn/Cws0/xANL/8VDiz/Gg8p/x4QJ/8h
EST/JBIh/ycSH/8rExz/LxMa/zQUF/86FBX/QBQT/0gUEf9PFA7/VxQN/14UDf9eFA3/XhQN/14UDf9
eFA3/XhQN/14U/0wVB/9FHgz/PCkR/zQ0Ff84ORf/OkIf/zpNJ/84WS//NGY2/zF0O/8ugT//K49B/yub
Q/8rpKp/K69E/yy6RP8txUT/LdFE/y7eRP8u6kT/LVE/zD9Q/8x/0P/Nf9C/Dn/Qfc8/0HyP/9D8D//Q/A
//0PwP/9D8D//Q/A//0PwP/9D/00VB/9GHgz/PSgR/zcyFP87Nhf/PkAf/z5KJ/88VjD/OGM3/zRxpF8wf
0H/LoxD/y6ZRf8vpEb/L61H/y+3R/8wwkf/Mc1H/zHbR/8y6Eb/M/NG/zT8Rv81/0X+Of9E+Dz/RPM//
0XtQv9G60L/RutC/0brQv9G60L/RutC/0brQv9G/04UB/9HHQv/PigQ/zowFP8+NBf/Qj0e/0JIKP9AV
DD/PGE4/zhuPv81fEL/MoIF/zKWR/8yoUj/MqpJ/zO0Sf80vkn/NMpJ/zXYSf825Un/N/FJ/zj7SP85/0f
6PP9H9ED/R+5D/0joRf9J5kX/SuZF/0rmRf9K5kX/SuZF/0rmRf9K/08UB/9IHQv/PycQ/z4uE/9DMh
b/Rzoe/0hGJ/9GUJd/QI45/z5rP/86eET/N4ZI/zaSSv82nkv/N6dM/zexTP84u0z/OMZM/znTTP8640z
/O+9L/jz5S/s9/0r1QP9J7kT/S+hG/0zhSP9N30n/Tt9J/07fSf9O30n/Tt9J/07fSf9O/1ATBv9KHAHv/QS
cQ/0MrE/9ILxX/Tjkd/09FJ/9OUTD/SVs5/0RoQf9AdUb/PYJK/zuOTf87mk7/O6RP/zytUP89t1D/PcJ

Q/z7PT/0/30/7QO1P+EH4TvVC/03vRv9O50j/UOBK/1HZTP9S1k3/UtZN/1LWTF9S1k3/UtZN/1LWTF9S/1ISBv9LHAv/QiYQ/0koEf9PLRT/Vjgc/1hDjF9WTi//Ulo5/0xlQf9HcUj/RH1N/0GKUP9BIVL/QaBT/kGpVP1Cs1T7Qr5T+kPLU/dE21P0RepS8Eb2Ue1I/1HmS/9T3U3/VdVP/1bOUf9XzFH/V8xR/1fMUf9XzFH/V8xR/1fMUf9X/1MSBv9NHAr/RCUQ/04mEP9XLBL/XjYa/2FAI/9fSy3/W1Y4/1ViQf9PbUn/S3IP/EiFU/pHkVb4R5xX9kemWPRHr1jzSLpX8UnHV+9K11brS+hW5031VeRN/1fbUP9Z0VL/W8tU/1zEVf9dwlB/XcJW/13CVv9dwlB/XcJW/13CVv9d/1URBv9OGwr/SSMO/1QjD/9eKhD/ZjQX/2k/IP9oSSv/ZFM1/15eQPtYakn3U3VQ9E+AVvFNjFvnTZdb7U2hXOtNq1zpTrZb50/DW+VQ1FrhUuZa3VP0WtIS/13OVf9fxlf/YcBY/2G7Wv9iuVr/Yrla/2K5Wv9iuVr/Yrla/2K5Wv9i/1YRBf9QGwr/TiEN/1shDf9mKQ7/bjMV/3I9Hf9yRyf9bIAy+GhaPfNhZUfvW3FQ61d8V+dUh1zIU5Je4IOdX+BTqGDeVLNg3FTAYNIV0GDVVuRg0VjzYcxX/mPCWv9lu1z+ZrZd/meyXv5nsV7+Z7Fe/mexXv5nsV7+Z7Fe/mexXv5n/1cQBf9RGgn/Ux8M/2EgC/9sJwz/dTES/3k7Gv56RSP3eE4t8XNXOetrYUTmZWxP4V94WN1bgl7ZWY5i1iIZZdNYo2bQWK1nzlm6Z8xayWfJW91nxFzvZ8Bc/Gm4XvtrsmD7a65h+2yqYvtsqWL7bKli+2ypYvtsqWL7bKli+2ypYvts/1kQBf9TGQn/WBwL/2cgCv9yJgr/ezAP/4A6FviCQx7xgUwo631VM+R3Xj/ecGIL2GhzWNJfWDNX4hmyl6SasddnWzEXahtwV6zbr9ewW68X9RuuWHpbrVh+W+uY/hwqWT4cKZI+HCiZvhwoWb4cKFm+HChZvhwoWb4cKFm+HChZvhw/1oOBf9UGQn/XBoJ/2wfCP94JQj/gS4M/lc4EvOJQRnsiUoi5YZSLd6BXDrWeWVKz3FuV8hreGLDZ4Jpv2SMbtrjl3G4YqJztmKtc7NjunSxZMx0rmXjdKtl9XSIZ/V1oWj1dZ5p9XWbafZ0mmn2dJpp9nSaafZ0mmn2dJpp9nSaafZ0/1sOBP9VGAn/YBgl/3EeBv98JAb/hi0J+lw2Du+QPXtnkUgc4I9QJtelWDfPgWJlx3prVsBzdGK6bn1rtWqGcbFokXWuZ5x4q2enealotHmmaMV5pGnceaFq8Xmca/N5mWzzeZZs83iUbPR3k2z0d5Ns9HeTbPR3k2z0d5Ns9HeTbPR3/1wOBP9WGAj/ZBcH/3UdBf+BlgX/iisG9JE0CuqWPQ/imEUW2pVLJNGPVjXliF9GwlFoVbl7cGGydXhrrXGBc6hui3ikbZZ8oWyifp9trn6cbb5+mm7Ufpdu7H6UbvB9kW/xfY9v8XyNb/J6jW/yeo1v8nqNb/J6jW/yeo1v8nqNb/J6/10NBP9XGAj/aBYG/3gcBP+EIQP7jigE8JYxB+abOQvenkAR1JpIlsyUVDPDj1Du4hlU7OCbWCsfHVrpXh9dKB0hnubcpF/mHGdgpVxqoOScbmDkHHOgo1x6IKLcu6BiXLvgIhy73+HcfB9h3HxfYdx8X2HcfF9h3HxfYdx8X2HcfF9/10NBP9YFwj/axYF/3wbA/+IHwL3kiUC7JotBOKhNQbZozkP0J9GH8eaUjG+IFtBtY5jUa2la1+mg3Jrn355dJh6gnyTd4yCj3WYhYt0pYaldbSHhnXJhoN15laCdeyEgnXtg4F17oGBdO9/gXTvf4F073+BdO9/gXTvf4F073+BdO9//14MBP9bFgf/bxUE/38ZAv+LHQH0ISEB6J8oAt6mLgPUpzUOy6REHsKfTy65mlk/sZRhT6iPaF2giW9qmYR2dJGAfn2LfleDhnqTiI5oIp/ea+KfHnEiXp64Il6eeqHennshXp47YN6d+6Be3fugXt37oF7d+6Be3fugXt37oF7d+6B/18MBP9eFAB/chQD/4IXAf+OGgDwmRwA5KMgANqrlgHQqzMMxqhBHL2kTSy1n1c9rJpfTaSVZlubkG1ok4tz4uGen2DgoOEFX+OiXh+nlx1fquMcn+/jHB/24txfumJcn3rh3N87IV0eu2DdHrtgnR67YJ0eu2CdHrtgnR67YJ0eu2C/2AMA/9hEgb/dhMC/4UVAPuSFQDsnRQA36cUANSvHgHLrzEKwa0/GbmpSyqwpVQ7p6BdSp6bZFmWImpmjZFwcYSMD3t8iX+EdlaKim+FI41shaeNaYW6jWeG1otohemKaoLqiGyA64Ztf+yEbn7tg25+7YNufu2Dbn7tg25+7YNufu2D/2ELA/9IEAX/eRIB/4kSAPeVDwDeoQkA1qwJAM+zGgDFtC4JvLI8F7SvSCirq1I4oqZaR5miYVaQnWdjH5htb32UdHI0kHuCbI6GiGeNIlxjjaSMYY23jF+O04pfjOmJYonqh2SG64ZmhOyEZ4PshGeD7IRng+yEZ4PshGeD7IRng+yE/2IKA/9pDQP/fRAA/4wNAN6aBgDUpQYAz7AHAMi4FgDAuisHt7g5Fa61RSWmsU41nK1WRJOpXVOKpWRggaBqbHeccHZtmXh/ZZeDhWCWkYhclqKJWpe1iFiX0ldYluuGW5HrhV2N7IRfiu2DX4ntg1+J7YNfie2DX4ntg1+J7YNfie2D/2QKA/9uCwL/ggwA7JEGANWeBADOqQUAyLQFAMG+EQC5wCcFsb82Eqm8QSGguUsxlrVSQI2xWU+DrWBceqlmaHCmbHJmo3R6X6J/gFqhjoNXoZ+DVaKyglOizYFSoe2AVJvugVaW7oFYku6BWZHugFmR7oBZke6AWZHugFmR7oBZke6A/2UJAv90CgH+hgcA2pYAAM+jAwDHrgMAwLkCALnECgCyyCIDq8cxD6LFPR6ZwkYtkL5OPIa6VUp8tltXcrNhYmmxaGxgrnFzWa18eFWti3tSrZ17UK6wek+uy3hOrex4Tqfy

ICi8ntSnPF7U5vxfFOb8XxTm/F8U5vxfFOb8XxTm/F8/2clAv96BwDljAAA05sAAMmoAgDAswlAub
8CALHLBgCr0RwBo9EsDJvPOBqSzEEoiMIJN37FUER0wlZRasBdW2G+ZGRavG1rVLt5b1C7iH
FOvJtxTLyvcEu8yW9Lu+tuSbb4cEuv93JMqPZ0Taf2dE2n9nRNp/Z0Taf2dE2n9nRNp/Z0/2oGAv+
BAQDYkwAAy6EAAMKuAQC5uQEAscUEAKjRBwCi2xgAm90pCJPbMxWJ2Twjf9ZEMXXTSj1s0
FFJYs5YU1vNYFtUzGlgT8x2ZEzMhmVKzZlSc6tZEFoYGNHzepiRMn8ZUXA/WdHuPxqR7X8ake
1/GpHtfxqR7X8ake1/GpHtfxq/3MAAOCJAADQmgAAxKgAALu0AACywAEaQmWEAJ/YCACA5R8
BIOcuB4znOBGD5j4deeRDKW3iRzV4UxAXN9TSVXfXE9P3mZUS99yV0nfglhI4JVXRuGpV0Xiw
FVE4eNUQd34Vz/Y/1pBzf9dQcn/XkHJ/15Byf9eQcn/XkHJ/15Byf9e/n0AANaSAADHogAAvK8AAL
O8AACpyAEAn9MEAjbsDQCQ7yACifEsCYHxNBN48TkebvA+KGTvQzFb7kg5VO5OQE7uVkvK
7mBJRu9rS0TvekxC8ltMQfGfTEDys0s/8s5JP/HuSDvt/0o65v9OO+P/Tzv/0874/9PO+P/Tzv/0874
/9P3okAAMucAAC+qwAAtbcAAKrEAACfzwAAInkCAIz2EQCF+R4DffsnCnX7LhNs+zQcY/s4JFr7
PSxU+0lyTftlN0j7UDtE/Fg+QPxiQD39bkl6/X1COF6QQjj/pEl3/7IBNv/TQDX+7z80+/8+NPr/PjT6/z4
0+v8+NPr/PjT6/z40+v8+0ZUAAMGmAAC2swAAq8EAAJ/MAACT1QAAh98AAID/EAB4/xoEb/8hC
Wf/KBBf/y0YV/8yHIH/NyRL/zwpRv9CLEH/SDA9/1AyOv9YNDb/YzYz/282MP9+Ny7/kTct/6Q2Lf+4
Niz/0DUr/+01Kv/zNCr/8zQq//M0Kv/zNCr/8zQq//M0xKEAALiWAACTvQAAockAAJPSAACG2wAAe
+8AAHP/CwFq/xQEYf8aCFn/IA1T/yUSTP8qF0f/LxtC/zQePv85ITr/PyQ2/0UmMv9NJy//Vikr/2AqK
P9sKib/eysj/4wrl/+eKyL/ryoi/8gqlf/PKiH/zyoh/88qlf/PKiH/zyoh/88quqwAAK+6AACixwAAINAAAIBZ
AAB44AAAbfgAAGX/BQFc/wwEVP8TB03/GApH/x0OQf8iEDz/JhM4/ysVNf8wFjH/NBgt/zkZKv9A
Gif/Rxwk/08dlf9aHh7/ZR4b/3MeGf+CHxj/kR8X/6lFf/+oHxf/qB8X/6gfF/+oHxf/qB8X/6gfsbYAAKTE
AACWzgAAhtcAAHffAABp5gAAX/4AAfb/AAFO/wQDR/8KBkH/EAg7/xQLNv8ZDTL/HQ4u/yEQK/
8IESf/KRIk/y0SIP8xEx3/NxQa/z4UF/9GFBX/UBQS/1sUEP9mFA3/chQL/4AUCv+FFAr/hRQK/4UU
Cv+FFAr/hRQK/4UUp8EAAJjNAACI1QAAeN4AAGjIAABa6wAAUP8AAEn/AAFB/wADO/8ABTX/B
Qcv/wolKv8OCif/Egsk/xCMIP8aDR3/HQ0a/yAOF/8kDhT/KA8R/y4PDv80Dwv/OxAI/0MQBv9MEAT
/VRAB/2APAf9jDwH/Yw8B/2MPAf9jDwH/Yw8B/2MP/z0YCF81lg3/LC0Q/ywyEv8tNxX/LD8X/ypKH
P8pVyP/J2Uo/yVzLP8kgS7/JI4w/yWZMf8mozL/Jqwy/ye0Mv8nvjL/KMky/ynVMv8p4jL/Ku0y/yv3Mf
8r/zH/LP8w/y3/MP8u/zD5Mv8x9TT/MvU0/zL1NP8y9TT/MvU0/zL1NP8y/z4YCF82IQ3/LS0Q/y8xEv
8xNRT/MDwY/y9IHf8uVST/LGMp/ypxLf8ofjD/Klsy/yiXM/8poTT/Kao0/yqyNP8ruzT/K8Y0/yzRNP8t
3zT/Lew0/y72M/8v/zP/MP8y/jD/Mvsy/zL1Nf808Tf/NfE3/zXxN/818Tf/NfE3/zXxN/81/z8XCF83IQz/Li
wQ/zluEv81MhT/NDox/zVGHf80UyT/MWAq/y9uL/8tezL/Llg0/yyUNv8tnjb/Lac3/y6vN/8vuTf/L8M3/
zDON/8x3Tb/Mek2/zL0Nv4z/jX7NP80+jX/NPY2/zbwOf836zv/OOs7/zjrO/846zv/OOs7/zjrO/84/OAX
CP85IQz/MCsQ/zYrEf85LxT/OzgX/zxDHf87UCX/OF0s/zVqMf8zdzT/MYQ3/zGQOP8ymzn/MqQ6/
zOsOv8ztTr/NL86/zXKOv812Tn+Nuc5+zfyOPk4/Dj2Of839Dn/Oe87/zrpPf885D//PeQ//z3kP/895D//
PeQ//z3kP/89/0IWCP86IAz/NCgP/zwoEP8/LBP/QjUW/0RBHf9CTSx/P1kt/zxmMv85czf/N4A6/ze
MPP83lz3/OKA9/zipPf85sj3+Obs9/TrGPfo71D34POQ89T3wPPI9+zvvPv877T7/PudA/z/gQv9B2k
T/QdpE/0HaRP9B2kT/QdpE/0HaRP9B/0MWCP88Hwz/OiUP/0EID/9HKHh/SzMV/0w+HP9LSiX/S
FYt/0RiNP9Bbn/Pns9/z2HP/09kkD7PZxB+T6lQfg/rkH2P7dB9UDCQfNB0EDwQuFA7UPuP+IE+j/
nQ/9B5EP/Q9xF/0XVR/9Gz0n/R89J/0fPSf9Hz0n/R89J/0fPSf9H/0UVCp89Hwz/PyIN/0giDv9PKQ//
VDET/1U8Gv9URyP/UVIs/01eNPxJajv5RXY/9kSCQvRDjUTyQ5hF8EShRe9FqkXtRbNF7Ea/Rep
HzETnSN5D40ntQ+BJ+UXcSf9l2En/Ss9L/0vJTf9Mw07/TMNO/0zDTv9Mw07/TMNO/0zDTv9M/0
YUB/8/Hgv/RB8M/04gDP9YJw3/XS8R/185GP9eRCH+Wk8q+VZbM/VRZjvxTXJB7Ut9RetKiUjoSp
NJ5kqdSeRLp0njTLBJ4U27SN9NyUjcTtl2E/rSNNP+EvQTV9Oyk7/UMNQ/1G9Uv9RuVP/UrIT/1K
5U/9SuVP/UrIT/1K5U/9S/0gUB/9AHQv/SRwL/1UeCv9fJQv/ZS0O/2g2FP5oQR33ZUsn8l9WMexa

YjroVW1C5FJ4SOBQhEveUI9N21CZTtlQo0/WUKxP1FC3UNJRxFDPUtVQy1ToUMdU91LDU/9U
vIT/VrhW/1ezV/9Xr1j/V69Y/1evWP9Xr1j/V69Y/1evWP9X/0kTB/9CHQv/ThoJ/1wdCf9nJAn/bSsL/3
E0EfhxPhnxb0gi6mpSLeRkXjffX2pB2lp0SdVXfk/RVYISziWTVcxVnVbJVadXx1axV8VWvVfDV85X
v1jiV7tZ8li4WP9as1n/XK1b/1yqXP9cplz/XKZc/1ymXP9cplz/XKZc/1ymXP9c/0oTB/9FGwr/Uxcl/2lc
B/9tlgf/dCkl+3gyDfJ6OxTreEUd43RPJ91vWzPVaGVAz2JvSspeeVLGXINXw1uNWsBal1y9WqFd
u1urXrlbt162XMZetFzbXrBd7l+tXf1gqF7/YaRf/2GhYP9hnmH/YZ5h/2GeYf9hnmH/YZ5h/2GeYf9h/
0sTBv9JGQn/WBUH/2gaBf9zlAX/eicG9n8vCe2COA/IgUIX3X5MldV4VzHNcWE/xmtrS8FmdFS8Y
35auGGHX7VgkWKyX5xjsF+mZK5gsWWrYMBIqWHTZaZi6WWjYvlmn2P+Zptk/maZZP9llmT/ZZ
Zk/2WWZP9llmT/ZZZk/2WWZP9l/0wSBv9MFwj/XRQG/20ZBP93HgP8fyQD8YUrBuilNQrgiT4R1o
VIH85/VC7GeV49v3NnSrlucFSzanlcr2eCYqxliGapZZZopmShaqRlrGqhZbpqn2XMapxm42qZZvZr
lmf8apNn/GqRZ/xpj2f9al9n/WiPZ/1oj2f9al9n/WiPZ/1o/00SBv9QFQf/YRME/3EYA/98HAL4hCEC7
YonA+OPMAbajzkO0YtGHciGUizAgFs7uXpkSbJ1bFSScXRdp219ZKNrhmmgapBtngmbbpppp2+
XabRwlWnGcJjp3m+QavNvjWr5botq+m2KavpsiWr7a4lq+2uJavtriWr7a4lq+2uJavtr/04SBv9TEwb
/ZRID/3QWAv9/GQH0iB0A6l8iAd+UKgLVIDUMzJFEGsOMTyq7h1k5s4FiR6x8aVOmd3FdoHN5Z
ptxgmyXbotwlG2WcpFtonSOBk90i23BdlIt2XSHbfBzhW33coRt+HCDbflvgmz5boJs+W6CbPlugmz
5boJs+W6CbPlu/08RBv9WEQb/aRED/3gUAf+DFgDwixgA5JMcANqZlgHQmTMKx5ZCGL6RTSi2
jFc3rodfRaeCZ1Kgfm5dmnp1ZpR2fW2Pc4dyi3GSdohwnneFcKt4gnC8elBx03d+ce12fXH1dX1w9
nN8cPdyfG/4cHxv+HB8b/hwfG/4cHxv+HB8b/hw/1ARBf9ZDwX/bBAC/3sSAPyGEgDsjiA35cTAN
WdHgDLnTEJwps/F7qXTcayklU1qo1dQ6KIZFCbhGtclH9yZo17em6HelJ0g3aNeH91mXp8dKd7e
XS3e3Z1znp1dep5dXX0d3V09XV1c/Z0dnL3cnZy93J2cvdydnL3cnZy93J2cvdy/1EQBf9cDQT/cA4
B/34OAPWJDADgkgkA2JsJANChGwDHOi4HvqA9FbacSSSul1QzppNcQZ2OYk6WimlajoVvZYeB
dm6AfX51enuJenZ6lXxyeaN9cHqzfW16ynxseud7bHnzeW549Hdvd/V1b3b2dG929nRvdyV0b3b2
dG929nRvdyV0/1IQBf9fCgP/cwwA/oELAOKMBgDWlgYA0Z4HAMukGADCpiwGuaQ7E7GhRyGp
nVEwoZlZPpmUYEYrKgzYiYttY4GHc2x5hHt0coGEem2AkX1qgKB+Z4CwfmWAXn1jgOR8ZH/ye
mZ983hne/R2aXr1dWl69XVpevV1aXr1dWl69XVpevV1/1MPBf9jCQL/dgoA8YQGANAiQAwdRmQ
UAzKIFAMWoFAC9qikFtak4EK2mRB+lo04tnJ9XPJSbXkmMlMrVg5JqYXqOcGpyi3dza4mBeW
WHjXxih5x+X4etfV2lw3xciOF7XlbyeV6D83hggfR2Yn/1dWJ/9XVif/V1Yn/1dWJ/9XVif/V1/1QOBf9o
BwH/egcA3YgAANOuAgDMnQMAxaYDAL+tEAC3ryYDsK41DqisQRyfqUsqL6VUOI6hW0aGnmF
SfZpnXXSWbmdslHVvZJF+dl6QinlbkJP7WZCreleQwHIWkd93VY/0d1eL9HZZiPV1W4X2dFuF9n
RbhFZ0W4X2dFuF9nRbhFZ0/1cNBP9tBgD2fgEA2lWAAm6YAQDGogIav6oBALiYcGcxtSICqrUxC
6KzPhmasEcnkaxQNIipV0KApl1Od6JjWW6gamNlnXFrXpt7cFmaiHRVmpd1U5qpdVKbvnNRm91
yT5n2clGU93NTkPdyVI33clSN93JUjfdyVI33clSN93JUjfdy/1wJA/9yAgDfgwAA0pEAAMidAADApw
AAuK8AALC3BACrux0BpLwtCZy6ORWTuEMji7VMMIKxUz15rllJcKxfU2eqZlxfqG5kWKZ4aVSmh
WxRppVtT6anbU2mvGtMptxqSqT1a0ug+mxNmvptTpb6bU6W+m1OlvtTpb6bU6W+m1Olvt/2ID
Af14AADZiQAAzJcAAMKjAAC6rAAAsrUAAKm9AgCiwXyAnMUoBpXENBGMwT8eg79HK3q8Tjdx
uVRCaLdbTGC1YIVZs2pbU7N1YE+ygmNNs5NjS7OmY0qzu2FJs9pgR7H0YUWt/2NHp/9lSKP+Z
kij/mZlo/5mSKP+Zkij/mZlo/5m/2kAAOJ/AADRkAAAxZ4AALypAACzsgAAqrsAAKHEAwCYzAwAk
88hA4zPLg2EzTkZe8tBJXLISDBpxk87YMRWRFnDXktTwmdRTsFyVUvBf1dJwpFXR8KkV0bDuV
VFw9hURMDzVUG9/1hBt/9aQrH/XEKx/1xCsf9cQrH/XEKx/1xCsf9c/3IAANmHAADJmAAAvqUAA
LWvAACsuQAAoslBAJjLBACP1AgAidsaAYPcKAh62zITctk7HmjXQihg1UkyWdRROILTUBN02
NFSdNuSEfTfEpF1I5KRNShSULVt0hB1dZHQdLwRz3P/0o7zf9NPMX/TzzF/088xf9PPMX/TzzF/0
88xf9P430AAM+RAADBoAAAt6wAAK62AACkwAAAmckBAI7RBACE3gkAgOYdAXnoKQdy5zEP

aec4GGDmPiFY5UQpUuVMMEzIVDVI5V44ReVpO0Pmdzx5og8QeebO0Dorjo/6Mg5PufpODrj+
zw34v8+Nd//QTXf/0E13/9BNd//QTXf/0E13/9B1okAAMWaaAAC6qAAAsLQAAKW+AACaxwAAjs8A
AIPXAgB77w0AdflcAm3zJQdl8ywOXvMyFlfzOB1Q8z4iSvNFJ0bzTStC9FYuPvRgMTz1bDI59Xsz
OPaNMjf3oTi297UxNffQMDT27S8z8/8vMPD/MTDw/zEw8P8xMPD/MTDw/zEw8P8xypUAAALyIAA
CysQAAp7wAAJvGAACozQAAgtUAAHbdAABv+w0AaP0YA2H+IAdZ/iYMU/8sEU3/MRZH/zcbQv
89Hj7/RCE7/0wkN/9VJjT/YCcx/2woL/98KC3/jigt/6loLP+2Jyv/0Scq/+omKf/8Jin//CYp//wmKf/8Jin//
CYp//wmv6EAALSuAACpugAAAnMQAAI/MAACB0wAAAdNsAAGnnAABi/wkBW/8SA1T/GQZO/x8K
SP8kDUL/KhA+/y8TOv80FTb/Ohcy/0EZL/9JGiz/Uhwp/1wdJv9pHST/eR4i/4self+eHSH/sB0g/8Ud
H//fHB//3xwf/98cH//fHB//3xwf/98ctqsAAKu4AACewwAAkMsAAIHSAABz2gAAZuEAAfz0AABV/w
MBTv8LA0j/EgVC/xclPf8dCjj/IQ00/yYOMP8rECz/LxEP/zQSF87EyL/QhMf/0sUHf9WFBryYxQY/3E
UFv+CEXx/khMV/6ITFP+yExT/shMU/7ITFP+yExT/shMU/7ITrrUAAKHBAACSygAAgtIAAHPaAAB
k4QAAV+cAAE/8AABl/wABQv8CAzz/CAQ2/w4GMf8TCC3/GAkq/xwLJv8gDCL/JAwf/ygNHP8tDhj
/Mg4V/zkPEv9CDw7/TA8L/1gPCf9IDwb/cg8E/4AOBP+ODgT/jg4E/44OBP+ODgT/jg4E/44Oo78A
AJTJAACE0QAAC9oAAGThAABV5wAASe0AAEL/AAA7/wABNf8AAi//AAQq/wMFJv8IBiL/DACf/xEI
HP8VCRn/GAkV/xsKEf8fCg3/lwsK/ygLBv8uCwP/NgsA/z8LAP9KCwD/VQsA/18LAP9rCwD/awsA/
2sLAP9rCwD/awsA/2sL/y8cCv8mJw3/Hi4P/yIwEP8iNBL/HzwU/xxIFv8bVxf/G2Ub/xpzHv8agB//G4
wh/xyXIf8doCL/Hqgi/x6wlv8fuCL/H8li/yDMLv8h2SL/IeYi/yLxIf8i+yH/I/8g/yT/IP8k/yD+JP8h/CX/Ivo
m/yP6Jv8j+ib/I/om/yP6Jv8j/zAcCv8nJg3/IS0P/yUvEP8IMxL/IzoU/yFGFv8gVBj/IGlc/x9wH/8efSH/H
4oi/yCUI/8hniT/IqYk/yKuJP8jtiT/I78k/yTKJP8I1iT/JeQj/ybvl/8m+SP/J/8i/Sj/Ivso/yL6KP8k+Cn/JfYq/
yb2Kv8m9ir/JvYq/yb2Kv8m/zEbCv8oJQ3/JSsO/yosD/8rMBH/KTgU/yhEF/8nURn/JI8e/yVtlf8keiP/J
IYI/yWRJv8mmyb/JqQm/yerJv8osyb/KLwm/ynGJv8p0yb/KuEm/yvtJfws+CX6LP8k+C3/JPYt/yb1L
P8n8i7/KfAv/ynwL/8p8C//KfAv/ynwL/8p/zIbCf8pJQ3/KSkO/y4pD/8wLRH/MjYU/zFBF/8vThr/LVwf/y
xpl/8qdib/Kolo/yuNKP8rlyn/LKAp/y2oKf8tsCn/Lrkp/y/DKf4vzyn8MN4o+ThrKPYy9ifzMv8n8jP/KPA
y/yrvMv8s6jT/Leg1/y3oNf8t6DX/Leg1/y3oNf8t/zQaCf8rJAz/LiUN/zQIDv83KRD/OjMT/zo+Fv84Shv/
Nlgh/zRIJf8ycin/MX4r/zGJLP8ykyz+M50t/TOILfw0rS36NbUt+TW/LPg2yzz1N9os8jjpK+849CvsOf8
r6jn/Lug4/zDmOP8x4Dr/Mt07/zLdO/8y3Tv/Mt07/zLdO/8y/zUaCf8tlwz/NCIM/zoiDf9AJw7/QzAR/0I
7Fv9BRxv/PIMi/zxgJ/86bSv8OXku+jifL/g5jzD2Opkw9DqhMPM7qTDyPLIw8Dy7MO89xy/tPtYv6T
7mL+ZA8y7jP/4x4D7/M90+/zXaPv8200D/N9BB/zjQQf840EH/ONBB/zjQQf84/zcZCf8vlgz/Oh4L/0
EeC/9JJQ3/TC4Q/0w3FP9KQhr/SE8i+0RbKPdCaC30QHqX8T+AM+9AijTtQJQ060GdNOIBpjToQ
q405kO4NOVExDPjRdMz30XIMttG8jTXRf0300T/OtFE/zvMRP88xkb/PcRH/z3ER/89xEf/PcRH/z3
ER/89/zgYcf80Hwv/PxsK/0kcCv9RJAv/VSwN/1Y0Ef9UPxj5UUog9E1XKO9KYy7rSG8z50Z6NuV
GhTjiR5A44EeZON5IojdSKs420i1OdIjWtNWSdA500riOc9L8TvKSv0+x0r/QMRJ/0G/Sv9Cukz/Q7
hM/0O4TP9DuEz/Q7hM/0O4TP9D/zoYCP85HAr/RRcJ/1AbCP9Zlgj/XikK/18xDvleOxTyW0Yd61Z
SJuzTXy7hUGo03U52OdpMgDzWTIo+1EyUP9JMnUDPTaZaZU2vQctOukHJTslBxk/bQcNQ7UK/
UPPeu0//RrhP/0e0UP9Ir1L/SK5S/0iuUv9IrL/SK5S/0iuUv9I/zsXCP8+GQn/ShQH/1cZBv9hHwb/Zi
YH+2guC/JoNxDqZklZ5GFOIt1dWyzXWGY10IvWPM5TekHKUoREyFKORsVSI0fDUqBlwVKqSL
9TtEi9U8JJulTTSbdV50mzVfdLsFX/TK1U/02pVv9NpVf/TaRY/02kWP9NpFj/TaRY/02kWP9N/zwX
CP9CFgj/TxIG/14XBP9nHQT/bSMF9XArB+xxMwzkcD4T3GxLHdRmVirNYWI1yF1rPsNadUW/W
H5JvFeITLpXkU24V5tOtVekT7NYr0+xWltQr1nMUKxZ4VCpWvNRplr/UqNa/1KfW/9SnFz/Uptc/1
KbXP9Sm1z/Uptc/1KbXP9S/z0WCP9GEwf/VREE/2MWA/9tGwL7cyAC73cnBOZ5LwfdeToN1HV
HG8xvUynFal41v2VnP7phcEe2X3IMsl2CULBcjFOtXJVUq1yfValdqlamXbZWpF3GVqJe21aeXu9
XnF7+V5le/1eWX/9XIF//VpNf/1aTX/9Wk1//VpNf/1aTX/9W/z8WCP9KEQb/Wg8D/2gTAv9yGAH2e

BwB6n0iAeCAKgPXgDYLznxEgCz3UCa/clozuG1jP7JpbEiuZXRpQmN9VKZihlekYZBaOWGaW59
hpVycYrFcmMLAXJhi1FyVYutck2L8XJBi/1yOYv9bjGL/Woxi/1qMYv9ajGL/Woxi/1qMYv9a/z8VCP
9NDwX/Xg4C/2wRAf92FADxfRcA5YlCAnuGlwHRhJMKyINBFsB+TSS5eVcysnRgPqpxwaEimbHB
Qoml5Vp5ngVubZotemGaVYJVloGGTZaxhkWW7Y05l2GMZedhimX5YYhl/2CGZf9fhWX/XYVI/1
2FZf9dhWX/XYVI/12FZf9d/0AVB/9QDAT/YgwB/28OAPx5EADsgBAA4IYTANWLHwDMizElw4k/F
LuESyK0f1UwrXtePKZ2ZUegcm1Qm291WJZsfV2TaoZhkGmRZI1pnGwKaKhmiGi3ZoVoymaDa
ORmgWn3ZIBp/2N/af9ifmj/YH5o/2B+aP9gfmj/YH5o/2B+aP9g/0IUB/9TCgP/ZQsB/3ILAPF8CgDeg
wgA2IkJANCPHADHkC4Hvo49EreKSSCvhVMuqIFbOqF9Y0aaeWpQIXVxWJBxeV+Lb4Jkh22MZ
4RsmGmBbKRqf2yyanxsxmp6bOBpeGz1aHhs/mZ4bP9kd2v/Y3dr/2J3a/9id2v/Yndr/2J3a/9i/0USB
v9WBwP/aQKA+3UIAN9/AwDWWhwUA0o0HAMuSGQDDICwFupl7EbKPRx6ri1Eso4dZOZyDYUSV
fmdPj3puWII2dWCEdH5lf3Klanxxk2x5cKBtdnCvbXNwwW1xcNxsCHHzanBw/WhwcP5mcW//ZXF
u/2Rxbv9kcW7/ZHFu/2Rxbv9k/0cQBv9aBgL/bAcA73gDANmCAgDSiwQAazZEGAMaWFgC+mSo
Etpc5D66URRynkE8pn4xXNpilXkORhGVNioBrV4N8cl99eXpmeHeEa3N2j25wdZxvbXWrb2t1vW
9oddhuaHXxbGh1/GppdP1oanL+Zmpy/mZqcv5manL+Zmpy/mZqcv5m/0oOBf9dBQH/bwQA33sA
ANSGAQDNjgMAyJUEAMGaEgC6nScDspw2DaqZQxqjlk0nm5JVNJOOXECMimNLhlZpVX2Cb1
52f3dmCH2Aa2t8i29oe5lwZXuocGJ7unBge9NvYHvwbWB6+2tiePxpY3f9Z2R2/mdkdv5nZHb+Z2R
2/mdkdv5n/04MBP9hAwD5cgAA2n8AANCKAADJkgIAwpkCALueDgC1oSQCraEzC6aeQBeem0o
klphTMY+UWj2HkGBJf41mU3eJbFfxwhnRkaYR8amSDiG5ggpVwXYKlCfuCt29ZgtBuWYLtbFmB
+2tbfvxpXHz9aF17/Wdde/1nXXv9Z117/Wdde/1n/1IIA/9mAQDpdgAA1YQAAMuOAADElwAAvZ4A
ALajCQCvpiABqaYwCaGkPRSZoUchkZ5QLombVzqBl15FeZRkUHGRallpJnFhYox5Z12LhWtZipN
tV4qjbVWLtWxTi85qUorraVKJ/GIUhf1oVoL+Z1aB/mZWgf5mVoH+ZlaB/mZWgf5m/1YEA9qAAD
eewAA0lgAAMeTAAC+mwAA6t6IAAK+oBACpqxwAo6wtBpuqORGUqEQejKVNKoSiVDZ7n1tBc5x
hS2uZZ1Rjl25cXZZ3YliUgmZUIJFnUpShZ1CU2ZPlcxITpTqZEyS/WVNjv9IT4r/ZFCJ/2RQif9kUln/
ZFCJ/2RQif9k/1sAAfdvAADYgAAAY4AAMGYAAC5oQAAsacAAKitAACisRYAnLloBJWxNQ6Or0
Aaha1JJn2qUDF1qFc8baVdRWWjZE5doWtVv6B0W1OfgF5Qn45gTp+fX0ygs15LoMtdSp/pXEec/
V5lmf9fSZT/X0qT/19Kk/9fSpP/X0qT/19Kk/9f/2EAAOJ2AADRhwaAXZMAALyeAAC0pgAAq60AAK
KzAACZuA4AlbojAo65MAqGuDsVfrZEIHazSyttSvI1Za9ZPI6uYEZYrGhNUqtxUk6rfVVLq4xWSqu
dVUissFRHrMITR6voUkSo/FVBp/9WQ6H/WESf/1hEn/9YRJ//WESf/1hEn/9Y/2kAANT+AADKjgAA
v5oAALakAACurAAApLMAAJq5AACQwAYAi8MbAIXDKgd+wjUQdsA+G26/RiVlvU0uXrxUNle6W
z1SuWRDTbluR0q4ekIHuYIKRrmbSkS5rklDucdHQ7jnRkC2+0k+tP9MPbD/Tj6u/04+rv9OPq7/Tj6u
/04+rv9O5nIAANGGAADDIQAaAuaEAALCqAACnsgAAAnLkAAJK/AACHxwQAf80QAHvPIgN0zy4Lb
M43FGTMPx1dy0clVspOLVHJVjNMyV83SMhqO0XJdzxDyYY9QsqZPUDKrDw/ysQ7P8nmOTzG
+Tw6xP8/N8L/QTfB/0l3wf9CN8H/QjfB/0l3wf9C230AAMmPAAC8nQAAs6gAAKmxAACfuQAAk8A
AAIjGAAB9zQQAc9YIAG7cGgBp3SYFYt0wDVvcOBVV3EAAct9tllkrbUSdG21orQ9tmLUDbcy8/3Il
vPt2ULz3dqC483r8tOt3gLdja9S412P8xM9X/MzLV/zQy1f80MtX/NDLV/zQy1f80z4kAAMCZAAC1
pgAArLAAAKG5AACVwAAAicYAAH3MAABY1AIAaOQJAGTpGQFe6iQFWOsrCIHrMhBM6zkVRu
tAGkLrSR0/61lgPOxdlzrsaSQ47XckN+6JJDunCQ177AjNO/KIjPu6iEw6v0iLuj/JS3o/yYt6P8mLej/
Ji3o/yYt6P8mxJUAALijAACurgAAo7gAAJfAAACKXgAAfcwAAHDTAABI2wAAXvQKAFn2FgJT9x8
FTfgmCUj4LA1D+TIQPvk4Ezr5QBU3+kgXNPpRGTH7XBov+2kbLfx4Gyz8ihsr/Z4bKv2zGin9zhoo
/ekZJvr8GSb5/xkm+f8ZJvn/GSb5/xkm+f8ZuqAAALGsAACmtwAAmMAAAIvGAAB8zAAA9MAAG
PaAABY4QAAU/4GAE3/EQJI/xgEQ/8fBz7/JAk6/yoMNV8wDjH/NQ8u/zwRK/9EEij/TRII/1gTlv9IEy
D/dRMf/4gSH/+cEh7/sBlD/8QRHP/jERz/6xEc/+sRHP/rERz/6xEc/+sRs6oAAKi1AACavwAAjMYAA

H3NAABu0wAAydoAAFXhAABM7QAAR/8BAEH/CQI8/xADOP8WBTP/HAcv/yEJK/8mCij/Kgsk/y8
Mlf81DR3/PQ0a/0YOF/9RDhX/Xg4S/24OEP+ADRD/kw0Q/6QMD/+2Cw7/vgsO/74LDv++Cw7/vgs
O/74LqrQAAJ2+AACOxgAAfs0AAG/UAABg2wAAUulAAEjoAABA+AAAO/8AADX/AAEx/wcDLP8
MBCj/EgUk/xYGI8bBx7/Hwga/yMIFv8nCRP/LQkQ/zQKDP89Cgj/SAoF/1UKAv9jCgD/cwK/4IJAP
+SCQD/mAkA/5gJAP+YCQD/mAkA/5gJn70AAJDGAACAQAAb9UAAGDcAABR4wAARekAADv
vAAA0/wAALv8AACn/AAEI/wACIf8CAx3/BgQa/wsFF/8OBRP/EwYQ/xYGC/8aBgj/HgcE/yMHAP8
qBwD/MgcA/zwlAP9IBwD/VQcA/2EHAP9uBwD/cwcA/3MHAP9zBwD/cwcA/3MH/yEhCv8XKQz/Fi
wN/xguDv8WMg//EToQ/w9HEf8OVhL/DmQS/w5yEf8OfxL/EYoT/xKUFP8TnRT/FKUU/xSsFP8Vs
xT/FrwU/xbFFP8X0BT/F94U/xjqE/8Z9RP/Gf4T/xr/Ev0a/xL8Gv8U/Br/Ffsa/xb7Gv8W+xr/Fvsa/xb7
Gv8W/yIlgCv8YKQz/GSoN/xwsDf8aMA//FjgQ/xZFEv8VUxL/FGET/xRvE/8UfBT/FYgV/xesFf8Ymx
b/GKIW/xmqFv8asRb/GrkW/xvCFv8czRb/HNsV/x3oFf8d8xX9Hv0U+x//FPkf/xX4H/8W+B//F/cf/xj3
H/8Y9x//GPcf/xj3H/8Y/yMgCv8ZKQz/HigM/yEqDf8fLQ7/HjYQ/x1CEv8cUBP/G14U/xtsFf8aeRb/G4
QX/x2OGP8dlxj/HqAY/x+nGP8grhj/ILYY/yG/GP8hyhj/lTgY/SPmF/sj8hf4JPsW9SX/FvQl/xjzJf8a8iT
/G/El/xzxJf8c8SX/HPEI/xzxJf8c/yQfCv8cKaz/lyYM/yYmDf8mKg7/JzMQ/yY/Ev8kTBT/l1oV/yJoF/8i
dBn/loAa/yOLGv8klBv/JZwb/yakG/8mqxv/J7Mb/ii8G/woxhR6KdQa9yrjGvQq7xnxK/oZ7yv/Gu0r/x3
sK/8e6iv/H+gs/yDoLP8g6Cz/IOgs/yDoLP8g/yYfCv8hJQv/KCIL/ywiDP8vJw3/MTEP/zA8Ev8uSBX/
LFUX/ytiGf8qcBz/Knwd/iuGHvwsKB77LJke+S2gHvguqB73L7Ae9S+4HvQwwx3zMdAd7zHgHewy
7RzpM/kd5jP/H+Qy/yLhM/8j3zL/JN4y/yXeMv8l3jL/Jd4y/yXeMv8l/yceCv8mIgv/LR4K/zldCv84JQz/
Oy4O/zs4Ef84RBX/NVEY/zReHPszax74MnYg9jOCIfQ0iyLyNJQi8DWdlu82pSHtN6wh7De1leo4v
yHpOc0g5jndIOI67B/fOvgi2zn/Jdg6/yfVOF8p0zn/KtE5/yrROf8q0Tn/KtE5/yrROf8q/ykdCv8Hgr/NB
oJ/zsbCf9CIgr/RCsM/0Q0EP9CPxT8P0wY9z1ZHfI7ZSHvO3Ej7Dt9JOo7hyXoPJAI5j2ZJeQ9oSXj
Pqkl4T+yJN9AvSTeQMok20DbJNZB6yXSQfcpzkD/K8tA/y3JP/8uxz//L8RA/zDEQP8wxED/MMRA/
zDEQP8w/yocCf8wGgn/OhYI/0MZB/9LIAj/TigK/04wDfpLOhL0SEcX7kVTHelDYCLIQmwm4kJ3KN
9CgincQ4wp2kOVKthDnSrWRKYr1ESuK9JEUcVQRcUszkXVLMpH5yzGR/Yvwkb/Mr9F/zO9Rf80
u0X/NbdG/zW3Rv81t0b/NbdG/zW3Rv81/ywcCf82Fwj/PxIG/0sXBf9THgX/VyUH/FctCvNWNgrUk
EV5U9OHN9MWyPaSmco1klyLNJlfc/QSIYwzUiPMctJmDLJSaAyx0qpM8VKsjPDSr4zwUvONL5
M4TS6TPE2tkz+OLNL/zmxS/86r0v/OqxM/zusTP87rEz/O6xM/zusTP87/y0bCf86Ewf/RQ4F/1IVBP
9aGwP/XyEE9WApBuxfMgrkXDwR3FpKGdVWViPPUmlry1BsMMdPdjtEToA2wk6JOL9Okjm9T5
o6u0+jOrlPrTu3ULg7tVDHO7NR2zuvUe08rFH7PqlR/z+mUf9ApFL/QKFT/0ChU/9AoVP/QKFT/0C
hU/9A/zEYCP8/EAb/TA0D/1kSAv9hFwH6Zh0C7mgkA+VoLQbcZzgM02NGF8xfUiLGW10swVho
M71WcTi5VXo8t1SDPrRUjECyVJVBsFSeQa5VqEKsVbJCqIXBQqhW00KIVuhDoVb4RJ5X/0WcV
v9Fmlb/RZhX/0SYV/9EmFf/RJhX/0SYV/9E/zUWCP9DDAX/UQwC/14QAf9mEwD0axgA6G4eAd5
wJgLUbzQKzGxCFcRoTyG+Y1osuWBkNLRdbDuwW3VArVp9Q6pZhkWoWY9HplmZSKRZo0iiW
q5Jn1q7SZ1azUmaW+NJI1r1SpVa/0qTWv9Kklr/SZBa/0mQWv9JkFr/SZBa/0mQWv9J/zgUB/9GC
gT/VgoB/2IMAP5qDgDubxEA4nMWANh2IAHOdzElxXQ/E75wSx+3a1YrsWdgNKxkaDyoYXFCpG
B5R6FfgUqfXopMnF6UTZpenk6YXqIPll62T5NeyE+RXt9Pjl7yT4xd/0+LXf9OiV3/TYhd/0yIXf9MiF3
/TIhd/0yIXf9M/zsRBv9KBwP/WggB/2YJAPFuCADicwgA2nYKANF8HQDIfS4GwHs9Ebh3SR2yclM
pq25cNKZrZT2haG1EnWV0SZIkfU2WYoZQIGKPUfFhmlOPYaVUjWGyVlphw1SIYdpUhmHwVIR
h/1ODYf9SgWH/UYFg/0+BYP9PgWD/T4Fg/0+BYP9P/z4PBf9NBAL/XgYA+GkFAN5xAgDYdwUA
03wHAMyAGgDDgiwFu4A6D7R9RhuteVENpnVaMqBxYjybbmlElmtxS5JoeFCPZoFTjGWLVolllliG
ZKFZhGSuWYFkv1l/ZNVZfWTtWHxk/IZ7ZP9VemT/U3pk/1J6ZP9SemT/Unpk/1J6ZP9S/0ENBf9Q
AwH/YQQA7GwAANI1AQDSfAMAZAGAMEFFwC+hykEt4Y4Da+CRBmof04lontXMZt3XzuVc2Z

EkHBtS4tttVGHa31WhGmGWYFpkVt+aJ1ce2irXXlou112aNFcdGjqW3Ro/FlzaP9Yc2j/VnNn/1Vz
Z/9Vc2f/VXNn/1VzZ/9V/0MKBP9UAgH/ZA EA328AANV5AADOGAlAyIUEAMKJEwC6iycDs4o2DK
ulQhekhEwjnYFVL5Z9XTqQeWNDi3VqS4VycVKAb3IYfG6CXHhtjV51bJpgc2ynYHBst2BubM1fb
GzoXmxs+1xsbP9abGv/WGxq/1dsav9XbGr/V2xq/1dsav9X/0YIA/9YAAD3ZwAA23MAANF9AADJ
hAEAw4kCAL2NEAC2kCQCr480CqiNQBWgiUohmYZTLZKCWjiMfmFChXtoSn93blJ6dXZYdXN/
XXFyiWBtcZZianGkYmhxtGJmclhZHHiYGRx+V5kcf9cZW//WmZu/1hmbv9YZm7/WGZu/1hmbv9
Y/0kGAv9bAADqagAA2HcAAM2BAADFiAAAv0AALiRDACxICIBq5QxCKSSPhOcj0gfiYtRK46IW
DWHhF9AgIFISXI9a1Fze3NYbni7XWl3hmFld5JjYnehY2B2sWNed8ZiXXbiYV12+F5ddv9dXnT/W
19z/1lfc/9ZX3P/WV9z/1lfc/9Z/0wDAv9fAADgbgAA03sAAMmFAADBjAAAupEAALOVCActmB8Ap
pkvB5+XOxGYIEYckZFPKImOVjOCilw9e4diRnSEaU9tgnBWZ4B4XGJ+gmBefY9iW32eYll9rmJXf
cNhVn3gX1Z89l5WfP9cV3r/W1l4/1pZeP9aWXj/Wl4/1pZeP9a/1AAAf1jAADccwAAzoAAAMSJAA
C8kAAAtZYAAK2aAwCnnRoAoZ4rBZqdOA6TmkMZjJdMJYSUVC99kVo6dY5gQ26MZktniW1TYY
d1WVuGf11YhYxfVYWbYFOFrF9ShsFeUIXeXE+E9VxPg/9bUID/WIJ+/1ISfv9ZUn7/WVJ+/1ISfv9Z
/1QAAO5nAADWeAAAYoUAAMCOAAC4lQAAsJsAAKefAACChohUAm6QnA5WjNQuOoUAWWhp5JI
X+bUCt3mVc1cJZeP2iUZEdhkmtOW5BzVFafPvHsJ4paUI+ZWk6PqllNj79YTI/dV0qN9FdJjP9XSo
n/V0uG/1dLhv9XS4b/V0uG/1dLhv9X/1kAAOJtAADRfgAAxloAALuTAACzmWAAq6AAAKCIAACZq
A8AlakjAY+pMQilqDwSgKZFHXijTSdxoVQwaZ9aOWKdYUFcnGhHVppwTVGZe1BOmYhSTJmY
UkqZqVJJmb5QSJncT0aX9FBEIf9RRJP/UkWQ/1JfKp9SRZD/UkWQ/1JfKp9S/2AAANt0AADLhA
AAv5AAALaZAACuoQAAPaYAAJurAACRrgcAjbAdAleWkWWArzcOea5AGHGGSsCFqqk8qYqlWMI
ynXTIWpmU/UaVtRE2leEdKpYZJSKWWSUelqEhFpbxHRKXaRUOj80ZAof9lP5//Sj+c/0s/nP9LP5
z/Sz+c/0s/nP9L52gAANN8AADEiwAAupcAALGgAACppwAAAn6wAAJSwAACHtQAAGrgUAH65JQ
N3uTEJcLg7Emm2QxtitUojW7RRKIWzWDBQsmE2S7FqOkixdTxFsYM9RLGUPUOypj1Bsro7QL
LYOj+v8js8rf8+Oqv/QDmq/0E5qv9BOar/QTmq/0E5qv9B3nIAAMuFAAC+kwAAAtJ4AAKymAACirQ
AAI7IAAlY2AACAuwAAAdEJAHLdHABtxCkFZsMzDGDDPBRZwkQbVMFLIU7AUydKwFwrRsBmL
kPAcjBBwlAxP8CRMT7BozA9wbgvPMHVLju+8S44vP8yNrr/NDS4/zY0uP82NLj/NjS4/zY0uP8203
0AAMOOAAC3mwAAr6UAAKWtAACaswAAjrgAAIK9AAB3wgAAa8kEAGPQDgBg0R8BW9lqBIX
RMwxQ0TwSS9FEF0fQTRxD0FYfQNBhlj3RbSM80XskO9KNJDnSoCM407QiN9PSITfQ7iE0zf0k
Mcv/Jy/J/ykvyf8pL8n/KS/J/ykvyf8pylkaALuYAACxpAAAqK0AAJ20AACQuQAAhL4AAHfDAABryQ
AAYNACAFfZBwBT4BYAUOEiAkviKwZG4jMKQuI8Dj/iRRI8404UOeNZFjfkZRc25HMYNeWEGDT
mlxcz5qsXMubDFJDI5BUt4fgXK+D/Gine/xsp3v8bKd7/Gyne/xsp3v8bv5QAALSiAACrrAAAn7QAAJ
O6AACFvwAAeMUAAGvKAABf0AAAVdgAAE3lQBK7hMARu8dAkLwJQU98SwlOfEzCjbyOgwz
8kMOMPNDy3zVxAr9GQRKfV0ESj1hRAo9poQJ/auDyX2yA8k9ugOI/L7DiHv/w8h7/8Ple//DyHv/
w8h7/8Ptp8AAK2rAACitAAAlbsAAIfBAAB4xgAAassAAF7RAABT2AAASd8AAEPzAwA//A4APP0X
Ajj9HgQ0/iQFMP8qByz/MAGp/zYKJv8+CiP/SAsG/1MMHv9gDBv/cQwb/4MLGv+YCxn/rQoY/8UJF/
/gCRb/9AgW//QIFv/0CBb/9AgW//Qlr6kAAKSzAACXuWAAiMIAAHnHAABqzQAAXNMAAFDaAAB
G4AAAPeYAADj8AAA1/wcAMP8OAS3/FQMp/xoEJv8gBSL/JAYe/yoGG/8wBxj/NwgV/0EIEv9MC
A7/WggL/2oICv99CAn/kQcJ/6QHCP+2Bgf/ygYH/8oGB//KBgf/ygYH/8oGp7IAAJm7AACKwgAAes
kAAGvPAABc1QAATtwAAEPiAAA56AAAMfEAAC3/AAAp/wAAJf8EASL/Cgle/xADG/8VAXj/GQQU
/x0EEf8iBQ3/KAUJ/y8GBf84BgL/QwYA/1EGAP9gBQD/cgUA/4MFAP+TBAD/ogQA/6IEAP+iBAD/
ogQA/6IEEnLsAAI3CAAB8yQAABNAAAFzYAABN3wAAQeUAADbqAAAt7wAAJvwAACL/AAAE/wA
AGv8AARf/AAEU/wMCEf8IAg3/DAMJ/xADBf8UAwH/GQQA/x4EAP8IBAD/LQQA/zgEAP9FBAD/U
wQA/2IEAP9vAwD/fAMA/3wDAP98AwD/fAMA/3wD/xMkCv8JKwv/DSol/w0rC/8JMAz/AjgM/wJGD

f8BVA3/AWMM/wFwDP8CfQv/BIgK/wWRCf8Gmgn/B6EJ/wioCf8Jrwn/CbYJ/wq+Cf8LyAn/C9UJ/w
zjCf8N7wj/Dfkl/g3/CPsO/wj6Dv8J+g7/CvkP/wv5D/8L+Q//C/kP/wv5D/8L/xUkCv8NKQv/ESgL/xEp
C/8NLQz/CjYN/wlDDf8IUQ3/CGAN/wdtDP8Jegz/CoUL/wyPCv8Nlww/DZ8L/w6mC/8PrAv/ELML/x
G8C/8RxQr/EtIk/xPgCv4T7Qr8FPcK+RT/CfcV/wr2Ff8L9RX/DPUV/w30Ff8O9BX/DvQV/w70Ff8O/
xYjCv8SJwr/FiUK/xcmC/8UKgz/EzMN/xJADv8RTg7/EFwO/xBqDv8Qdg3/EoEM/xOLDP8UIA3/FZ
wN/xajDf8Wqg3/F7EN/xi5DP8Ywgz+Gc4M/BrdDPka6gz2G/YL8xv/C/Ec/w3wHP8P7xz/EOwe/xHs
Hv8R7B7/Eewe/xHsHv8R/xgjCv8XJAr/HCIK/x0jCv8dJwv/HTEN/xw9Dv8aSg//GVgP/xIID/8Zcg//Gn
4O/xulD/8ckA//HZgP/h6gD/0epw/7H64P+iC1D/kgvw/4lcsO9SLaDvli6A7ul/QN7CP/D+oj/xHol/8S5
SX/FOMI/xTjJf8V4yX/FeMI/xXjJf8V/xkiCv8clQr/IR4J/yMeCf8nJQr/KS4M/yc5Dv8kRg//l1MQ/yJgEf
8ibRH9l3kR+iSDEvglijBL3JZQS9SacEvQnoxLzKKoR8SiyEfApvBHvKscR7SrWEekr5hDILPMR4iz
+E+As/xXclF8X2i3/GNgs/xnYLP8Z2Cz/Gdgs/xnYLP8Z/xshCv8hHQn/Jxol/yoaCP8xIgn/MysL/zl1D
f8vQQ//LU4R+ytbE/craBP0LHQU8Sx+Fe8tiBXtLpAV6y+YFeowoBXpMacU5zGvFOYyuRTkM8QU
4jTTE9405BPaNPIW1jT9Gdl1/xvPNP8czTT/Hcwz/x7LM/8eyzP/Hssz/x7LM/8e/x4fCf8mGQj/LRUH
/zQYB/87Hwf/PScJ/zwxC/06PA/3N0kS8TVWFO00YxbqNW4X5zV5GOQ2gxjiN4wY4DiVGN85nRj
dOaQY2zmsGNo6thjYOsIz1TrQGdl84hnNPPEcyTz9H8Y7/yHDO/8iwTv/l787/yS/Ov8kvzr/JL86/y
S/Ov8k/yIcCf8sFQf/MxAG/z4VBf9EHAX/RiQG/kUsCfVDNw3uQEMR6D5QFeM9XRjfPWka3D10G
9k9fh3WPocd1D6QHtl/mB/QP6AfzkCoH8xAsCDKQbsgyEHJIMVC2yHBQ+wivUL6JbpC/ye3Qf8ot
UH/KbRB/ymzQf8ps0H/KbNB/ymzQf8p/yYZCP8xEqb/Ogwe/0UTA/9MGQP/TyAE9k8oBu1NMQnl
ST0O3khLFNhGVxnTRWMez0RulcxEeCPJRIEkx0WKJcV/FkibDRpomwUaiJ79Hqye9R7Unu0fDK
Ll1Ci2Secpskj3K69l/y2sR/8uqkf/Lqhl/y6nSf8up0n/LqdJ/y6nSf8u/ysWB/82DQX/QQoD/00QAv9TF
QH6VhsB7lCjA+VWLAXcVDgK1FFGE81PUhvtV4hxExoJcFLcii+S3sqvEuELLLlJc24S5QttkydLrR
Mpi6yTLAvsE29L65NzS+rTulvp07zMaRN/zOhTv8zoE7/M55N/zOeTf8znk3/M55N/zOeTf8z/y8TBv
87CQT/SAgC/1IMAP9ZEADzXBUA514cAN1eJALUXTMly1tBEsRYThu/VVkjulNjKbdSbS20UXUw
sVB+Mq9QHjOtUY80q1GYNaIRoTWnUas2pVK3NqNSxzagU9w2nVPvOJpS/TiYUv85l1L/OJVR/zi
VUf84lVH/OJVR/ziVUf84/zMQBf8/BQP/TQYB/1cJAPddCgDsYAwA4GIRANVIHwDMZjAHxGQ+E
L1hShq3XVUjstlfKq5YaDCrV3E0qFZ5N6VWgTmjVoo6oVaTO59WnDydVqc8m1eyPZIXwj2WVtY
9lFbsPZFW+z6PVv89jX/PY1V/zyMVf88jFX/PlxV/zyMVf88/zYNBf9CAQL/UQQA+IsEAORhAwDa
ZQUA1mcJAM5sGwDFbiwFvWw7DrdpRxxmwZVlj2JcK6dfZTKjXW03n1x0Opxbd2aW4U/mFqO
QZZamEKUWqNCKlquQo9avUONWdFDilnoQ4hZ+kOHwf9ChVn/QYRZ/0CEwf9AhFn/QIRZ/0CE
Wf9A/zkKBP9GAAH/VQEA618AANtmAADTawMAz20GAMhyGADAdCoEuHM4DbFvRBerbE8hp
WIYKqBmYTKbY2k4mGFwPZVgeEGSXoFDkF6KRY1dIEeLXZ9HiV2rSIZduUiEXc1IglzISIBd+Ed+
Xf9GfVz/RX1c/0N8XP9DfFz/Q3xc/0N8XP9D/zwHA/9KAAD9WAAA4GIAANVrAADOCaIAyXMEA
MJ3FQC7eScDtHg2C612QhWmckwgoG9VKppsXjKWaWU5kWZtP45kdEOKYnxHiGGFSYVhkEu
CYJtMgGCnTH5gtUx7YMIMeWDiTHdg9kt2YP9JdmD/SHVg/0Z1YP9GdWD/RnVg/0Z1YP9G/z8F
Av9OAAZwWAA3GcAANFvAADJdQAAxHgCAL17EQC2fiUCr34zCal7QBOieEoem3VTKJZxWz
GQbml5jGtpQldpcUWDZ3hJgGWBTH1li056ZJdQeGSkUHVksIBzZMVQcGPeUG9k9E5vZP9Mbm
T/Sm5j/0luY/9lbpP/SG5j/0huY/9l/0ECAv9RAADnXgAA2GsAAM10AADFeQAAv30AALI/DQCygyl
Bq4MxCKWAPRKefkgcl3pRJPf3WDCCLc2A4hnBmQlFtbUZ9a3VLeWp9TnVpiFFyaJNTb2igU21or
1NraMFTaGjbUmho8lBnaP9OZ2j/TGhn/0toZ/9KaGf/Smhn/0poZ/9K/0QAAf9UAADhYgAA028AAM
l4AADBfgAAu4EAALSECgCuhx8AqlcvBqGFOxGag0Yak39OJY18Vi6HeV03gXZkP3tzakZ2cXJLc
W96UG5thFNqbZBVaG2dVWVtrFVjbL5VYWzYVGBs8FJgbP9QYgZ/TmFr/0xha/9MYWv/TGFr/0x
ha/9M/0cAAf1XAADdZwAA0HMAAMV8AAC9ggAAtoUAAK+IBgCpixwAo4wsBZ2LOQ6WiEQYj4V

MloiCVCyCf1s1fHxhPXZ5aERwdm5La3R3UGZzgFNjcoxWYHKaVI5yqVZccrtVWnLUVFly7INZcf9
RWXH/T1pw/01bb/9NW2//TVtv/01bb/9N/0oAAPFbAADZawAAzHcAAMKAAAC6hgAAsooAAKuM
AgCljxkAn5EpA5iQNguSjUEVi4pKIISIUSI9hVgydoJfO3B/ZUJqfWxJZHt0TI96fVJceYIVWXmXVId5
pIVVebhUU3nRU1N47VJTd/5QU3f/T1R1/01UdP9NVHT/TVR0/01UdP9N/00AAOVfAADUcAAAx3
wAAL6FAAC2iwAAro8AAKWRAACfIBQAmpYmApSVNAmNkz4ThpFHHH+OTyZ4i1YvcYlC OGqH
Yj9khWIGXoNxS1mBelBWgYZSU4CUU1GBpFJPGbZRTToHPUE2A609Mfv1OTH3/TU18/0xNe/9M
TXv/TE17/0xNe/9M/1IAAOBIAADQdQAAw4EAALqKAACyKAAAqpQAAJ+XAACYmQ8AIJsiAY6b
MAeHmTsPgJdFGXmVTCJyk1Mra5FaM2WPYDtejWdBWYtuR1SKeEtRioRNTToqSTkyKo01LirVM
SYrOSkiJ6kpHh/1KRob/SkaE/0pHg/9KR4P/SkeD/0pHg/9K+VgAANtsAADKewAAvocAALWPAAC
tlQAAPzoAAJmdAACRnwgAjaEdAleilASBoTcMep9BFXOdSR5sm1AmZZIXLI+YXTVZImQ7VJV
sQFCUdkRMIIJGSpSRRkiUokZHILRERpTNQ0WT6kJDkfxEQY//RUCO/0VBjf9FQY3/RUGN/0VBjf9
F6F8AANNzAADEggAAuo0AALGVAACpmwAAoKAAAJsjAACJpQEAg6gWAH+pJgJ5qDIlc6c8E
GylRBhlpEwgX6NTJ1miWi5ToGEzT6BqN0ufdDtln4A8Rp+PPUWfoDxDn7I7Qp/LOkGe6Tk/m/w7
PZr/PTyY/z47mP8+O5j/PjuY/z47mP8+4GgAAMx7AAC+iQAAtZMAAKycAACkogAAmaYAAI6pAA
CBraAAea8NAHWwHwBwsCwFarA2C2SvPxJerkYZWK1OIFKsVSVOq10qSatmLkarcTBDq30yQ
quNMkCrnjE/q7AwPqvJLz6q6C47qPsxOab/Mzek/zU3o/81N6P/NTej/zU3o/811nIAAMSEAAC5kQ
AAsJsAAKeiAACdqAAakqWAAIavAAB6sgAAbbcCAGi5FQBluiQBYLovBlu6OAxVuUASULIIF0u4U
BxHuFggRLdil0G3bSU/uHomPbiJJy4myY7uK4IOrjHJDm35iM3tPolNLL/KDKx/yoysf8qMrH/KjKx/y
oysf8qzH0AAL2NAACzmQAAq6IAAKGpAACVrgAAibIAAH21AABwuQAAZL4AAFrEBwBWxhgAU
8ckAk/HLwZLxzcKR8dADkPHSRJAX1IVPcdcGDvHaBk5x3UaN8iFGjbJlXo1yasZNMnDGDPH5Bc
xxPkZL8L/HC3A/x4swP8eLMD/HizA/x4swP8ewokAALeWAACtoQAAPkKAAJiwAACMtAAAf7cAA
HK7AABmvwAAWsUAAFDLAWBG0wgARNUWAELXlgE/2CwEPNg1BjnZPwk32UkLNNIUDTPaY
A4x220OMNt9Di/ckA4u3KQOLd27DSvc3Awp2PMOJ9X/ECXT/xll0v8SJdL/EiXS/xll0v8SupQAAL
CgAACnqQAAM7EAAI+2AACBugAAc74AAGbCAABaxwAAT80AAEXTAAA92gQAoeUOADfnGg
A16CQCMugsAy/pNAUs6j0GKupHByjrUggm7F8IJe1uCSTtgAgj7pQllu6pByDuwgce7uQGHen5B
hvn/wcb5/8IG+f/CBvn/wgb5/8lsp8AAKqpAACesQAakbcAAIO8AAB0wAAAZsUAAFnKAABNzwAA
Q9UAADvbAAAy4QAAMPQJAC72EwAr9xsBKPciAiX4KQMi+TAEH/k4BB36QgUa+00FGPtbBRb
8awUV/X4FFP2TBRP+qQQS/sIEEP3gAw789wMO+/sDDv7Aw77+wMO+/sDrKgAAKGxAACTuA
AAhb0AAHXCAABmxwAAWM0AAEzTAABB2QAAON4AAC/jAAAp7wAAJv8BACP/CgAh/xIBHv8
YARv/HQIY/yMCFf8pAxL/MQMO/zoDC/9GAwf/VAME/2UDA/94AwL/jgMB/6MCAP+3AgD/0AIA/9k
CAP/ZAgD/2QIA/9kCo7EAAJa5AACHvgAAAd8QAAGfKAABX0AAAStYAAD/dAAA04gAALOYAAAC
PqAAAf+QAAHP8AABn/AAAX/wYAFP8MARH/EQEN/xYBCf8bAgX/IQIB/ygCAP8xAgD/PQIA/0sC
AP9cAgD/bwIA/4ICAP+UAQD/pgEA/6sBAP+rAQD/qwEA/6sBmLkAAIq/AAB5xQAAaMwAAFjSAA
BJ2gAAPeAAADLmAAAO6gAAIO4AABjzAAAV/wAAEv8AAA7/AAAM/wAACf8AAAAb/BAEC/wcBA
P8MAQD/EQEA/xgBAP8fAQD/KAIA/zICAP9AAQD/UAEA/2EBAP9wAQD/gQEA/4UBAP+FAQD/h
QEA/4UB/wQnCf8CKAn/BCcJ/wEpCf8ALQn/ADYJ/wBECf8AUgj/AGAH/wBuB/8Aegb/AIUF/wCO
BP8AlgT/AJ0E/wCkA/8AqgP/ALEC/wC4Av8AwQL/AMwC/wDaAv8A5wL/APMC/AD8AfKA/wH4AP
8C9wH/A/cB/wP3Av8E9wL/BPcC/wT3Av8E/wUmCf8GJgn/CCUJ/wYmCf8AKgn/ADQJ/wBBCf8AT
wn/AF0I/wBrB/8Adwb/AIIG/wCLBf8AkW/AZsE/wGhBP8CqAP/Aq4D/wO2A/8DvgP/BMkD/wXWA/
0F5QP6BvEC9wb7AvQH/wPzCP8E8gn/BfEK/wXvDP8G7wz/Bu8M/wbvDP8G/wcmCf8LJAn/Dill/
wwjCP8HJwn/CDEJ/wc+Cv8FTAn/BVKJ/wRnCP8Fcfw/Bn4H/wilBv8JkAb/CZcF/wqeBf8LpQT/C6s
E/gyzBP0MwT8DcYE+g7TBPcO4gTzEO8D8BD6A+4R/wXsEv8G6hP/B+cV/wjmFv8I5hb/COYW

/wjmfV8l/wolCf8SIQj/FR8l/xQfCP8TJQj/Ey4J/xl6Cv8QSAr/D1UK/w5iCf8Obwn/EHol/xKEB/0TjAf7
E5QG+hSbBvkVogb4FakG9havBvUXuAb0F8IF8hjPBe8Z4AXrGe0F6Br5BuYb/wjjHP8J3x7/Ct4e/
wvdHf8L3R3/C90d/wvdHf8L/xAiCf8XHQj/GhoH/xoaB/8flgj/HywJ/x02Cv8bQwr/GVAL/xleCvsZagr4
GnUJ9ht/CfQciAjjHZAl8R6XCO8fnwjuH6UI7SCtCOshtQjqlb8H6CLMB+Uj3Qfhl+wH3iT4Ctol/wzV
Jv8N0yb/Dtlm/w/RJf8Q0SX/ENEI/xDRJf8Q/xYfCP8cGAf/IBUG/yQXBv8pHwb/KigH/ykyCf8mPgr7J
EsL9iNYDPIjZQvvJHAL7CV7C+omhAvoJ4wK5iiUCuUpmwrjKqIK4iqqCuArsgrfLL0K3SzKCdot3Ar
VLusM0S74Dswv/xDJLv8Sxy7/E8Uu/xTELv8UxC7/FMQu/xTELv8U/xobB/8hFAB/JhAF/y4UBP8z
HAX/NCQG/zMuB/gxOQnxLkYL7C1TDOgtYA3kLmsN4S92Dd8wfw3cMYgN2jKQDtkymA7WMP8
O1TOnD9M0rw/RNLkPzzXFD8011RDIN+cRxDf2FMA3/xa9Nv8Xuzb/GLo2/xm5Nf8ZuTX/Gbk1/x
m5Nf8Z/x8XB/8nEAX/LQsE/zcRA/88GAP/Ph8D+DwoBe86MwfoOD8K4jdNDd03Wg7YN2YQ1Dh
wEtI4ehPPOYITzTqKFMS6khTJO5oVyDuhFcY8qhXEPLMWwj2/Fr89zha8PuIXuD7yGbU+/xyyPf8
dsD3/Hq89/x6tPf8erT3/Hq09/x6tPf8e/yQTBv8sCgT/NggC/z8NAf9EFAH7RhoB8EUiAuZCLAXeQT
kl1kBHDdFAVBHMP18VyEBqF8VAcxnDQHwawUGEG79BjBu9QpQcu0KcHLpDpR24Q64dtkO5
HbREyB6xRNwerUTuIKpE/CKoQ/8jpUT/l6NE/yOiRP8jokT/l6JE/yOiRP8j/ygQBf8xBgP/PQYB/0YK
AP9KDgDzTBMA50wbAN1KJQHUSzMGzEpBDcZJThTBSFoYvUdkHLpHbR64R3Ygtkd+lBRHhiK
ySI4jsEiXl69loCStSakkq0m0JKlJwiWnSdUlo0rpJqBK+SedSv8om0n/KJpJ/yiZSP8omUj/KJll/yiZS
P8o/ywMBP82AQL/QwMA/0sFAPJPBgDoUAoA3IAQANRTHwDLVS8FxFQ9Db1SShS4UFUbtE5f
H7FOaCOuTXElrE15J6pNgSioTYkppk2SKqROmyqiTqUroE6vK55PvSucT88smU/ILJZO9i2UTv8t
kk3/LZFN/y2QTf8tke3/LZBN/y2QTf8t/zAIA/86AAH/SAaA8k8AAN5UAADYVwQA1FYIAMxbGwDE
XSsEvFw5DLZaRhSwV1EbrFZblahUZCaU2wpo1N0LKBSfC6eUoQvnFONMJpTljCYU6AXllKrMZ
RSuDKSUsoykFLhMo1S9DKLUf8yiVH/MohR/zGIUf8xiFH/MYhR/zGIUf8x/zMFA/8/AAD+TAAA4IQ
AANhbAADRXglAzF4FAMViFwC9ZSgDtmQ2CrBiQxOqX04bpVxXlqFbYCIDWWgsmIhwMJhXdzK
VVoA0k1aINZFWkjaPVpw3jVanN4tWtDeJVcY3hIXdOIRV8TeCVf83gVX/NoBV/zWAVP81gFT/NY
BU/zWAVP81/zYCAv9DAADyTwAA3VkaANJhAADLZQAAXmUCAL9oEwC4ayUCsWszCatoQBKI
Zkoan2NUlpphXSmXXmUuk11sMpBbdDWOWns4i1qEOolZjjuHWZg8hVmkPIJZsTyAWcl9fVjZPX
tY7zx6WP87eVj/OnhY/zl4WP84eFj/OHhY/zh4WP84/zkAAf9HAADIUgAA2F4AAM5mAADGagAA
wGsAALptDwCzcSIBrHExB6ZvPRCgbEgZmmIRlpVmWimRZGEvjWfPnllgcDiGXng7g12APYFdi
9+XZRAfFygQXpcrUF3XL5BdVzVQXNc7UByXP0+cVz/PXFc/ztxW/86cVv/OnFb/zpxW/86/zwAAf9
KAADiVwAA1GMAAMlqAADBbwAAu3EAALVyDACvdSAAqHYuBqJ0Ow6bckUYIm5OIJBvYiLaV
4vh2ZINYNkbTp/YnQ9fGF8QHlghkN2YJFEdGCdRHFgq0VvX7tEbV/RRGtf6kNqX/xBal//P2pf/z5q
X/89al//PWpf/z1qX/89/z4AAPpNAADeWwAA0GcAAMZvAAC+dAAAt3UAALB2CACqeh0ApHssBZ
55OQ2Xd0MWkXRMH4xxVCeGblwvgWtiNX1paTp5Z3E/dWZ5QnJlgkVvZl1HbGSaR2pkqEdnZLh
HZWPOR2Nj6EVjY/pDY2P/QWNj/0BjY/8/Y2P/P2Nj/z9jY/8//0AAAO9QAADaYAAAZGsAAMJzAAC
6eAAAs3oAAKx7BQCmfhoAoH8qA5p+NguUfEEUjXIKHYd2UiaBc1ktfHFgNHduZjpybG0/bmt1RG
ppf0dnaYpJZGmXSWJppUlgaLVJXmjKSFxo5kdcaPIFXGf/Q1xn/0FcZ/9AXGf/QFxn/0BcZ/9A/0MA
AOdUAADWZAAAYW8AAL93AAC3fAAAr38AAKd/AQChghYAnlQnApaDNAmQgT8SiX5IG4N8Ty
R9eVYsd3ddM3F0ZDlscms/Z3ByRGNve0dgboZJXW6TSltuokpZbrJJV27ISVZt5EdWbfhFV mz/RF
Zs/0JWbP9BVmz/QVZs/0FWbP9B/0cAAONZAADSaQAAXQAALt8AACzgQAaq4QAAKOEAAACc
hxMAI4kkAZKIMQeLhzwPhYRFGH6CTSF4f1Qpcn1bMGx7YTdmeWg9YXdvQl11eEZZdYNIV3S
QSVV0n0ITdLBIUXTFR1B04kZPc/dFT3L/Q09x/0JQcf9BUHH/QVBx/0FQcf9B/ksAAN5eAADNbgA
AwXkAALiAAACwhgAAqIkAAJ2JAACXjA4Ako4hAl2OLgWGjDkNgIpDFXmIsh5zhEmbIRYLWac
XjRhGUGU6XH5tP1d9dUNTfIBGUXyOR098nUZNFk5FTHzDREt74ENKevZDSXn/Qkl4/0FJd/9ASXf

/QEI3/0BJd/9A8FAAANIkaADJcwAAvX4AALSFAACsiwAAo44AAJeOAAcQkQkAjJMcAleUKwOB
kzYKepFAEnSPSBptjU8iZ4tVKWGXDBbiGM2VodqO1KFcz5OhX5BTIWMQkqFm0FJhaxAR4X
CP0aE3z5FgvU+RIH/PkOA/z5Df/8+Q3//PkN//z5Df/8+51cAANNrAADEeQAAuYMAALCLAACokQ
AAnpMAAJKUAACJlwlAhJkWAICaJgJ6mTIHdJg8Dm6XRBZnlUwdYZRSJFuSWSpWkWAwUZBo
NE2PcThKjnw6SI6KOKaPmjpfJ6s5Q4/BOEKO3jZAJPU4P4r/OT6J/zk+iP85Poj/OT6I/zk+iP854F8A
AMxyAAC/gAAAtYoAAKyRAACklwAAmJgAAI2aAACAnQAAeqAPAHehlAByoS0EbKA3CmafQBF
gnkcXW51PHIWcViNRm10oTJpILemZby9GmXsxRjMjMUKZmTFBmaowQJm/Lz+Z3S49I/QvO5
X/MTmT/zl4kv8zOJL/MziS/zM4kv8z2GkAAMZ6AAC5hwAAAsJAAAKiYAAcfnQAAk54AAlehAAB6p
AAAcKcGAGuoGABoqSYBY6kxBl6oOgtZp0IRVKdKF0+mURtLpVkgR6Vil0SlbCZBpXgnP6WGJz
6llic9pagmPKW9JTql3CQ5ovMIN6D/KDWe/yk0nf8qNJ3/KjSd/yo0nf8qz3IAAL+DAAC0jgAArJgAA
KOfAACyOwAAjKUAAlCoAABzqwAAZK4AAF6wDQBcsh0AWLIpAlSyMwZQsjLS7FDD0exSxNE
sVQWQbFdGT6waBs8sXQcOrGCHDmxkxw4saYbN7G7GjWx2Rk0rvlaMqz/HTCr/x8uqv8gLqr/IC
6q/yAuqv8gxX0AALiMAACvlwAAp58AAJ2IAACRqQAAhawAAHivAABrsQAAxUAAFG6AQBmVb
AAS70eAEi+KQJFvjlfQr47CD++RAs8vk0NOB5XDze+YhE1v28RNL99EjLAjxExwKIRMMC3EC6/
1A8tvfEQK7r/Eim5/xQot/8VKLf/FSi3/xUot/8VvYgAALKVAAcqnwAAoaYAAJWsAAClrwAAerlAAG
20AABguAAAVLwAAErAAABAxgMAOsoNADjMGgA3zSUANc0vAjPOOQMxzkmFL89NBi3PWQcr
0GYIKtF1CCnRhWgo0psHJ9KwByXSzQYkz+wGlsz8CCHJ/wogyP8LIMj/CyDI/wsgyP8LtpMAAK2f
AACkpwAAmK0AAluxAAB9tQAAb7gAAGG7AABVvwAASsQAAEDJAAA3zgAAL9MEACfaCQAI3
RQAJd4gACTfKgAj4DQBlue/ASHiSwlg41gCHuNnAh3keAlc5YwCG+WiAhnluglX5d0BFED2AhTf/
wMU3v8EFN7/BBTe/wQU3v8Er54AAKenAACbrgAAjbMAAH+3AABwuwAAyB4AAFTDAABJyAA
APs0AADXSAAAAt1gAAJdwAAB/kAgAe7Q0AHO4WABrvHwAZ8CcAF/EvABXyOQES80UBEPRT
AQ70ZAEN9XcBDPA NAQv2pAEJ9r0AB/beAATz9QAD8P8AA/D/AAPw/wAD8P8AqacAAJ6vAAC
QtQAAGbkAAHG+AABiwQAAMVYAAEfMAAA80QAAM9YAAcRbAAAI4AAAG+MAABfwAAAV/AQ
AE/0MABH+EwAN/xkAC/8hAAj/KQAF/zIAAv8+AAD/TQAA/14AAP9yAAD/iQAA/6AAAP+2AAD/0A
AA/+YAAP/mAAD/5gAA/+YAoK8AAJO2AAcEuwAAc8AAAGPEAABUygAARtAAADvWAAAw3AA
AJ+AAAB/kAAAX6AAAEOSAAA35AAAL/wAACf8AAAb/BgAC/wsAAP8SAAD/GQAA/yEAAP8qAA
D/NQAA/0UAAP9WAAD/awAA/4AAAP+VAAD/pwAA/7QAAP+0AAD/tAAA/7QAIbYAAIa8AAB2wg
AAZccAAFTNAABG1AAAOdwAAC3hAAAK5gAAG+kAABPtAAAM7wAABvYAAAP/AAAB/wAAAP
8AAAD/AAAA/wAAAP8CAAD/CAAA/w4AAP8YAAD/IQAA/ywAAP87AAD/TAAA/18AAP9xAAD/gg
AA/44AAP+OAAAD/jgAA/44A/wApCP8AJQf/ACQH/wAmB/8AKwb/ADQG/wBCBf8AUAX/AF4E/wBr
A/8AdwL/AIEC/wCKAf8AkGh/AJB/wCgAf8ApgD/AKwA/wCzAP8AuwD/AMQA/gDQAPwA3wD5A
OwA9wD3APYA/wD1AP8A9AD/APQA/wDxAP8A8QD/APEA/wDxAP8A/wAnCP8Alwf/ACIH/wAjB/
8AJwb/ADIG/wA/Bv8ATQX/AFsE/wBoA/8AdAP/AH4C/wCHAv8AjwH/AJYB/wCDAf8AowH/AKkA/w
CwAP0AtwD8AMEA+gDNAPgA2wD1AOoA8wD1APEA/wDvAP8A7wD/AOWA/wHpAP8B6QD/Aek
A/wHpAP8B/wAkB/8FIAf/BR4G/wAfBv8AJAb/AC8G/wA7Bv8ASQX/AFcF/wBkBP8AcAP/AHoC/w
CEAv8AjAl/AJMB/gCaAf0AoAH8AKYB+gCtAPkAtAD3AL4A9QDJAPMA2ADwAOcA7AD0AOoD/
wDoBf8B5Qj/AulK/wLhCv8C4Qr/AuEK/wLhCv8C/wYhB/8LHQf/DBsG/wgbBv8Jlgb/CCwG/wc4Bv
8FRQb/BFIF/wNfBP8EawT9BXyD+wZ/A/kHiAL4B48C9giWAvUJnQHHzCaMB8gmqAfEKsQHvCro
B7gvGAewM1QDnDOUA4w7zAeEQ/wLdEv8D2RT/BNgU/wTWFP8F1hT/BdYU/wXWFP8F/wwEB
/8RGQb/EhUF/xlXBf8WIAX/FSkF/xM0Bv8RQAb/EE0F+w9aBfcPZwT0EXEE8RJ7A+8ThAPuE4s
C7BSTAusVmQLpFaAC6BanAuYWrwLIF7gB4xjDAeEY0wHdGeQB2RvzA9Qd/gTPHv8GzR7/B8
we/wfKHv8Iyh7/CMoe/wjKHv8I/xlaBv8WEwX/Fw8E/x0UBP8hHAT/ISUE/x8vBfwcOwb2G0gG8Rp

VBe0bYQXpHGwF5x12BOQefwTiH4cD4SCPA98glgPdIZ0D3CKkA9oirAPYI7YD1iPBA9Qk0APQ
JelEyyfxBsco/gjEKP8KwSj/C78o/wu+J/8Mvif/DL4n/wy+J/8M/xCVBv8bDQT/HgkD/ycQAv8rFwL/Ky
AD+yoqBPMnNQXsJUIF5iVPBullXAbeJ2cF2yhyBdgoewbVKYMG0yqLBtlrkqfQLJkHziygB8wtqAf
KLrAlyC+7CMYvygjDMNwlvzHtC7sx+w24Mf8OtjD/D7Qw/xCzMP8QszD/ELMw/xCzMP8Q/xsRBf8
hCQP/KAcC/zAMAf80EwH9NBoB8jlkAukwLwPhLjsE2y9JBdUvVgfRMWEJzjJsCszdQvJNH0Lxz
SFDMU1jAzDNpMNwTabDb83ow2+N6wNvDi2Dro4xA64OdUOtDnpELA5+BKtOP8Uqzj/FKg5/xW
nOf8Vpzn/Fac5/xWnOf8V/yAMBP8mAwwL/MAQB/zglAP87DQD0OxMA6DkcAN83JgHWODQDzzl
DB8o6UAvFOIsNwjtlD787bxC9PHcRuzx/Erk9hhK3PY4Ttj6WE7Q+nhOyPqcUsD+xFK4/vhSsP88
VqUDkFqY/9RiiQP8ZoD//Gp8//xqdPv8anT7/Gp0+/xqdPv8a/yQJA/8rAAH/NwAA/j4CAPBABQDoP
wkA3z0QANRAHwDMQi8DxUM9CL9DSg27QIYRt0JgFLRCaRWyQ3EXsEN5GK5DgRmtRlkZq0S
RGqlEmRqoRalapkWsG6RFuRuiRckbn0bfHJtG8R2ZRf8el0T/H5VE/x+UQ/8fIEP/H5RD/x+UQ/8f/y
gEAv8xAAD/PAAA7EIAAN1GAADXsAMA1EQIAMtJGwDDTCsDvEw5CbZLRQ6ySIETrkpbF6tJZ
BqpSWwcpkl0HaVJfB6jSYMfoUmMIJ9KICCdSp4hm0qolZIKtCKXsSqiUralpJK7iOPSf0kjkN/Jlxl/y
OLSP8ji0j/l4tl/yOLSP8j/ywAAv82AAD3QQA4EkAANZPAADPUQAAY04EAMNRFgC8VScCtVU1
CK9TQQ+qUkwVpIBWGqNQXx2gT2cgnU9vlpTdySZTn8lI06HJpZOkCaUTponkk6kJ5BOsCiOTs
Aoi07VKIIN6yiGTfsphU3/KINM/yiDTP8ng0z/J4NM/yeDTP8n/y8AAf87AADnRQAA208AANBWAA
DIWAAAw1cBALxYEgC2XCMBR1wyB6lbPg6kWUkVn1dTG5tWWx+YVGMjIVNrJpNTciiRUnoj1K
DK41SjCyLUpYsiVKhLYdRrS2EUbwgtlHQLYBR6C1+UfotfFD/LHtQ/yt7UP8relD/KnpQ/yp6UP8q/z
IAAP8+AADjSwAA1VUAAMtcAADCXwAAvV4AALdeDgCwYiABqmMvBqRhOw2eX0YUmV1QG5
VbWCGSWWAljhnKYxXbyuJVnYuh1V+L4VViDCCVZlXgFWdMn5VqjJ8VbgyeVTMMndU5TJ1VP
gxdFT/MHNU/y9zVP8uc1T/LXNU/y1zVP8t/zUAAPZCAADfUAAA0VoAAMZhAAC+ZAAAUgQAAL
FkCgCraB0ApWksBJ9oOAuZZUMTIGNNGpBgVSGMXI0miF1kKoVbay6CWnMxf1I7M31ZhDR6W
l41eFiaNnZYpzZzWLU2cVjJNm9X4jZtV/Y0bVf/M2xX/zFsV/8wbFf/MGxX/zBsV/8w/zcAAOtFAADb
VAAAzV8AAMJmAAC6aQAAs2oAAK1pBgCnbRoAoW4qA5ttNgqVa0ESkGhKGYtmUiCGY1omg
mFhK35gaS97XnAzeF14NnVcgThyXls5cFyXOm5cpDprXLI6aVvGOmdb3zlmW/Q3ZVv/NWVb/zR
IW/8yZVv/MmVb/zJIW/8y/zkAAOdKAADWWQAAYWMAAL9qAAC3bgAAsG8AAKhuAwCicRcAnX
MnApdyNAiRcD4QjG5IGIZrUB+BaVcmfGZeK3hlZTB0Y200cWJ0N25hfTprYlg8aGCUPGZgoT1jY
LA8YV/DPF9f3TtfX/M5X1//N15e/zZeXv80XI7/NF5e/zReXv80/z0AAOROAADTXQAAXWgAALxvA
AC0cgAArHMAAKRzAACedRQAmXglAZN3MQeNdTwPh3NFFoJwTR58blUld2xcK3NqYjBuaGk1
amdXOGdlejtkZYQ9YWWQPI9knj5cZK0+WmTAPVhk2j1YY/E7WGP/OVhi/zdYYv82WGL/NVhi/zV
YYv81/UAAAOBTAADPYgAAwmwAALizAACwdwAAqHgAAJ93AACZehEAlXwiAY98LwWJejoNg
3hDFH12Sxx3dFljcnFZKW1wYC9obmc0ZGxuOGBrdzxdae+WmqNP1hqmz9Waas/VGm9PIJp2
D1SaPA7Umj/OVJn/zhSZ/82Umf/NIJn/zZSZ/828kQAANxYAADLZgAAv3EAALV3AACtfAAApHwA
AJp8AACUfgwAkIEfAlqBLASFgDclfn5BEnh8SRpzelAhbXhWJ2h2XS1jdGQyXnNrN1pxdDtWcH4
9VHCKPIJwmT5QcKk9TnC7PE1w1TtMb+86TG7/OUxt/zhMbP82TGz/Nkxs/zZMbP8260oAANddA
ADHawAAu3UAALJ8AACqgAAAn4EAAJWBAACOGwgAioYbAlWgKQJ/hjQleYQ+D3OCRhdtgE0
eaH9UJGJ9Wipde2EvWHpoNFR5cThReHs6TnelO0x3ljtLd6c6Sxe5OUh30zhHdu43RnX+N0Z0/z
ZGc/81RnP/NUZz/zVGc/815IAANFjAADCCQAaUHoAAK+BAAClhQAAM4YAAI+GAACliQMAg4s
WAH+MJQF5jDEGdls6DG6JQxNoiEoaYoZRIF2FWCZYg14rU4JmME+BbjNMgHk1SYCGNkiAITZ
GgKU1RIC4NEOA0jNCfu0zQX3+M0B7/zNAev8zQHR/M0B6/zNAev8z4FcAAMxqAAC+dwAAAtIAA
AKuHAAChiQAALosAAIqMAACAjwAAepEQAHeSIABzky0DbZI3CWeRPw9ij0cVXI5OG1eNVSFTj
FwITotjKkqKbC1HincvRYqEMEOKky9CiqQvQYq3LT+K0Cw+iOwsPIb9LTuE/y47g/8uO4P/LzuD/y

87g/8v2WAAAMZxAAC5fQAAslcAAKiNAACcjwAAkZAAAIWSAAB3lQAACZclAG2ZGgBqmicBZZo
yBWCZOwtbmEMQVpdKFVKWURpNIVkeSZRhIkaUaiVDIHUnQZSCJ0CUkic+IKMmPZS1JTyUzy
Q7kuwkOJD9JjeO/yc2jf8oNY3/KDWN/yg1jf8o0GkAAMB5AAC1hQAARl4AAKOTAACXIAAAi5YAA
H+YAABynAAAZ58AAGGhEQBfoiAAXKlraIeiNQZToT0LT6FFD0ugTRNHoFUXQ59dGkGfZxw+n
3ldPJ+AHjufjx46n6EdOZ+zHDefzRs2nesbNJv9HTKZ/x8xmP8gMjJ/IDCY/yAwmP8gyHIAALqBAA
CwjAAAqJUAAJ6ZAACRmgAAhZwAAHifAABsogAAXqYAAFWpBgBSqhYAUKSiAE2rLQJJqzYFR
qs+CEOrRww/qk8OPapYETqqYhI4qm4TN6p8FDWvjBQ0q54TM6uxEjGryhExqekRLqf7EyyI/xUro/
8XK6P/Fyuj/xcro/8XwH0AALSKAACrIAAAo5wAAJegAACKoQAAfqQAAHGnAABlqgAAWK0AAEu
wAABEsWGABqBQWAD+1lgA+tiwBO7Y1Azm2PgU2tkcHNLZRCDK3XAkwt2gKL7d2Ci63hwosuJk
KK7itCSm3xgggptuYIJrP6CiWy/wwwsP8NI7D/DiOw/w4jsP8OulcAAK+TAACnnQAAnaQAAGnAA
CDqgAAAdq0AAGivAABbsQAAT7QAAES4AAA6uwAAMcAGAC7CEgAtwh4ALMMoACrEMQApXT
sBKMVGAIbGUgllxl4DJMdtAyPHfgMiyJEDIMimAh7lvglDxuECHMP4AxvB/wQav/8GGr//Bhq//wYa
v/8GspIAAKmdAACHpQAAlKoAAleuAAB4sAAAarMAAFy1AABQuAAARbwAADu/AAAxAAAKcg
AACHNBAAa0QoAGNMUABfUHwAX1SkAftU0ABXWQAAV2E4AFNldABPabwAS2oMAEduZAA
7bsQAM2dEAC9bvAAvU/gAL0v8BC9L/AQvS/wEL0v8BrJ0AAKSIAACXrAAAirAAAHuzAABstgAA
XbkAAFC8AABEWAAAOsQAADDJAAAnzQAAH9EAABjVAAAR2gEAC98GAARlEAAJ5hoACOCjA
AfoLgAG6ToABOpJAALqWgAB6m0AAOqDAADpmwAA6LQAAOjTAADn7gAA5v4AAOb/AADm/w
AA5v8ApqYAAJqtAACNsgAAfrYAAG65AABevQAAUMEAAEPFAAA4ygAALs8AACXTAAAd2AAA
FtsAAA7fAAA14gAAA+sAAAHZBQAA8wwAAPMVAAD0HgAA9CgAAPU0AAD2QgAA9IUAPZpA
AD2gAAA9pkAAPawAAD2yAAA9uQAAPbmAAD25gAA9uYAna0AAI+zAACAUAAAcLwAAGDAAA
BQxQAAQ8oAADfQAAAs1QAAI9oAABreAAAS4QAAC+QAAATnAAAA6QAAAPUAAAD8AAAA/A
AAP0GAAD9DQAA/hcAAP8hAAD/LAAA/zsAAP9NAAD/YwAA/3oAAP+RAAD/pAAA/7YAAP+4A
AD/uAAA/7gAkrQAAlO5AABYvgAAYsMAAFHIAABDzwAANdYAAcncAAAg4QAAF+UAAA7oAAA
H6wAAAO0AAADvAAAA8wAAAP0AAAD/AAAA/wAAAP8AAAD/AAAA/wUAAP8NAAD/GQAA/yQ
AAP8yAAD/RQAA/1oAAP9vAAD/gQAA/5IAAP+UAAD/lAAA/5QA/wAmBv8Alwb/ACIF/wAjBf8AJw
T/ADID/wBAA/8ATgL/AFwB/wBoAf8AdAD/AH4A/wCGAP8AjgD/AJUA/wCbAP4AoQD9AKcA/Act
APsAtAD6AL0A+gDHAPgA1QD3AOQA9gDwAPQA/ADzAP8A8QD/AO8A/wDtAP8A7AD/AOWA/w
DsAP8A/wAjBv8AlAb/AB4F/wAfBf8AJAT/AC8D/wA9A/8ASwL/AFgB/wBIAf8AcAD/AHoA/wCDAP8
AiwD9AJIA+wCYAPkAngD4AKQA9wCqAPYAsQD1ALkA9ADDAPIA0ADxAOAA7wDuAO0A+QDs
AP8A6gD/AOYA/wDIAP8A5QD/AOUA/wDIAP8A/wAgBv8AHAX/ABoF/wAbBP8AIQT/ACwD/wA5A
/8ARwL/AFQB/wBhAf8AbAD/AHYA/AB/APkAhwD3Al4A9QCVAPMAmwDxAKEA8ACnAO8ArgDu
ALYA7ADAAOsAzADpANwA5wDsAOUA9wDjAP8A3wD/AN0A/wDcAP8A2wD/ANsA/wDbAP8A/w
EdBv8EGAX/AhYE/wAXBP8AHwP/ACkD/wA1A/8AQgL/AE8C/wBcAfwAZwH5AHIA9gB7APMAgw
DxAIoA7gCRAOWAlwDqAJ4A6ACkAOcAqwDmALMA5AC9AOIAyQDgANoA3gDqANsB+ADWBv
8A0wj/ANEI/wHQCP8Bzwn/Ac8J/wHPCf8B/wgZBf8KFAT/CBAD/woUA/8MHQP/CiUD/wcwA/8FP
QL7BEoC9gNXAfIDYwHvBG0A7AV2AOOGfwDoBoYA5geNAOUHIADjB5oA4QihAN8lqADdCbEA
2wq7ANkKyADWC9oA0g7sAM0S+QDKFP8BxxX/AsUV/wLDFv8Dwxb/A8MW/wPDFv8D/w0VBf8
PDQP/DgkD/xUQAv8YGQL/FyIC/xQrAvCRNwLwEEQB6xBRAecQXQHjEWgA4BJyAN4TegDcFI
A2hSJANGVkADWFpcA1BeeANMYpgDRGa4Azxq3AM0axADKG9UAxh7oAcEf9wK+IP8EuyD/Bb
kg/wW4IP8GuCD/Brgg/wa4IP8G/xIQBP8VBWP/GAUB/x8LAf8hEwH/IRwB9R4mAewbMQHIGj4B3
xpLAdscWAHWHWMB0x9sAdEgdQHPIX0CzSKEAssjiwLJJJCxyWZAsYmoQLEJqkDwiczA8Aov
wO9Kc8DuinjBLYq8wazKv8HsCr/CK8q/wmsKv8Jqyv/Casr/wmrK/8J/xcKA/8aAAL/lglB/ygHAP8pD

AD2KBQA6yUeAOEjKQDaIzYA0yVEAc4oUQPKKV0DxytnBMQsbwXCLXcFwC5/Br4vhga9MI0Gu
zCVBrkxNAe4MaUHtjKuB7QyugeyMsolrzPeCKsz8AqoM/4MpTP/DaMz/w2iM/8NojL/Dqly/w6iMv8
O/xsFAv8eAAH/KQAA/i8BAPlvBADsLQoA4CkTANYqIQDOLzABYDE+A8IzSwW+NfChuzVhCLk2
agm3NnlKtTd5C7M4gQuyOlGlsDmQDK45mAytOaAMqzqqDak6tQ2nOsQNpTvZDqE77A+eO/sR
mzr/EZo6/xKZO8SmDn/Epg5/xKYOf8S/x8AAv8mAAD/MAAA6jQAAN43AADYNwMA1TEJAMw2
GwDEOisBvTw5Bbg8Rgi0PVELsT1bDa89ZA6sPmwPqz50EKK/exGnP4MRpj+LEqRAkxKjQJwSo
UCmE59BsROcQb8TmkHTFJdB6BWUQPkWkj//FpA//xaPPv8Wjz7/Fo8+/xaPPv8W/yMAAf8sAAD
xNQAA4D0AANVBAADOQgAAyj0EAMNAFgC7RCYBtUU0BbBFQQmsREwNqERWEKZEXxKjR
GcUoURvFZ9FdhAErX4XnEWGF5pFjhiYRpcYl0aiGZVGrRmTRrsZkEXOG05F5RqLRfcbiUT/G4d
E/xuGQ/8bhkP/GoZD/xqGQ/8a/yYAAP8xAADmPAAA2UUAAM5KAADGSwAAwUgBALtIEQC0TC
IBrk0wBalNPQqkTEgPoUtSE51LWxaaSmMYmEpqGpZKchuVSnkck0qBHZFKih6PSpMejUqeHot
KqR+JSbcfh0nJH4RJ4SCCSfQggEj/H35l/x9+SP8efUj/Hn1l/x59SP8e/ykAAPc1AADhQgAA00wAA
MhRAADAUwAAulAAALRPDQCuUx8AqFQtBKNUOQmeU0QPmlJOFJZQVxiTT18bkU9mHY5Ob
h+MTnUhiK19lohNhiOGTY8jhE2aJlJNpiSATbMkfk3FJHtN3iV5TPIkd0z/JHZM/yN1TP8idUv/IXVL/
yF1S/8h/ysAAOw6AADcSAAAZIIAMNYYAAC7WQAAtVcAAK9WCQCpWRwAo1sqA51bNgmYWU
EPIFdlFJBWUxmMVFsdIvNjIldSaiOEUnEkgIF5JoBRgid+UYwofFGXKHpRoyl3UbApdVDCKXJQ
2ilxUPAob1D/J25P/yZuT/8lBk//JG5P/yRuT/8k/y8AAOg/AADYTQAAYlcAAL9dAAC3XwAAsF0AAK
pcBQCkXxgAnmEnAplgNAiTz8Oj11IFlPbURqGWVgeg1hgloBXZyV9Vm4ne1V2KXhVfit2VlgsdF
STLXFUoC1vVK0tbFS/LWpT1i1pU+4saFP+KmdT/ylnU/8nZ1P/J2dT/ydnU/8n/zMAAOREAADTUg
AAxlwAALxiAAC0ZAAaQ2IAAKVhAQCFZBUAmmYIAZRmMQaPZDwNimJGE4VgThmBXIYefV1dl
3pbZCZ2Wmspc1lyLHFYey5uWIUvbFiQMGIYnTBnWKswZVe8MGJX0zBhV+wuYVf9LGBX/ytgVv
8qYFb/KWBW/yIlgVv8p+jYAAOFIAADPVgAAwmAAALImAACwaAAAp2cAAKBmAACaaBIAImSiAZ
BrLwWLaTkLhWdDEoBISxh8Y1Med2Jal3RgYSdwX2grbV1wLmpdeDBnXlIxzFyNMmJcmjNgXKk
zXVu5Mltb0DJaW+owWlr8Llpa/y1aWv8rWlr/K1pa/ytaWv8r8DoAAN1NAADMWwAAv2UAALZqAA
CsbAAAo2wAAJxqAACWbQ4AkW8fAlxwLASHbjcKgW1BEXxrSRd3aVAdcmdXIm5IXidqZGUrZm
JtLmNhdTFgYX8zXWCLNFtgmDRZYKY0V2C3M1VfzjNUX+kxVF/7L1Re/y5UXv8sVF7/LFRe/yxU
Xv8s7D8AANISAADIXwAAvGkAALNvAACpCAAAn3AAAJdvAACRcQsAjXQcAlh1KgOCdDUIfXI+
D3dwRhVybK4bbWxVIWlrWyZkaWlqYGHqLI1ncjFZZnwzV2WINVVIITVTZaQ0UWW1NE9IzDNOZ
OcxTmP6ME5j/y5OYv8tTmL/LU5i/y1OYv8t6EQAANRXAADEZAAAUW0AALBzAACIdAAAm3QAA
JJ0AACMdgCAh3kZAIN6JwJ+eTIHeHg8DXJ2RBNtdEsZaHNSH2NxWSRfcGApWm5nLVdtbZBTb
HkyUWyFNE9skjRNbKlZS2uzMklryjFlauYwSGn5L0ho/y5laP8tSGf/LEhn/yxIZ/8s5EoAAM9cAADA
aQAAtNIAAK14AACheAAAl3kAAI15AACGewlAgX4VAH1/JAF4fy8Fc345Cm18QRBoe0kWY3IPH
F54ViFZdl0mVXVkkIF0bC1Oc3YwS3OCMUlzkDFIc6AwRnOxL0VzyC5EcuQtQ3D4LUJv/yxCbv8s
Qm7/K0Ju/ytCbv8r3IAAAMpiAAC9bwAAs3cAAKh8AACdfQAAk30AAIZ+AAB/gQAAeoMQAHaFH
wByhSsDbYQ1CGiDPg1igkUTXYFMGFI/Ux1UfloiUH1hJkx8ailJfHQrRnuALEV7jixDe54sQnuwKk
B7xik/euMoPnj4KT13/yk9dv8pPXX/KT11/yk9df8p2FgAAMVpAAC5dAAAr30AAKSBAACYgQAAjol
AAIKEAAB4hwAAcokJAG6LGgBqjCcBZosxBWGLOgpcikIPV4IJFFOIUBhPh1ccS4ZfIEeFaCNEh
XIIQoV+JkCFjCY/hZ0IPoWuJD2FxSM8hOliOoL3IziA/yQ4fv8IN37/JTd+/yU3fv8l0WAAAMBwAAC1
ewAArIQAAJ+GAACUhgAAiYgAAH2KAABvjgAAaJACAGOSEwBhkyEAXZMsAlmTNQZVkj0KUZ
FFDk2RTBJJkFQWRo9cGUKPZRxAj3AdPo98HjyPix47j5sdOo+tHDIpwxS4julaNYz3HDSK/x0yiP8
eMoj/HzKI/x8yiP8fyWkAALt3AACwggAAp4oAAJqLAACPjAAAg44AAHiQAABrIAAAX5cAAFiZCgB
VmhkAU5sIAFCbLwJNmzgGSZtACUaaSAxCMlAPP5IYET2ZYhM7mW0VOZI5FTeZiBU2mpkVNZ

qrFDOZwhMymOASMJb2FC6U/xUtk/8XLJL/FyyS/xcskv8XwnIAALWAAACsigAAopAAAJWQAAC
JkgAAfZQAAHGXAABlmgAAV54AAE2hAABlow4ARqQbAESkJgBCpTACQKU4BD2kQQY6pEkIO
KRTCjakXQs0pGgMMqV1DTGhAwwpZUMLqWoCy2lvgsrpN0KKqL1Cyif/w0nnv8OJp3/Dyad/w8
mnf8Pu3wAALCJAACokgAAnZYAAI+XAACCMQAAdpwAAGqfAABdogAAUqUAAEapAAA9qwAA
N60OADWuGgA0riUAM68uADGvNwEvr0ECLa9LAyywVQQqsGEFKbBvBSiWfgUmsZAFJbGjBCS
xuQQisNgDla3zBB+r/wYeqv8HHqr/CB6q/wgeqv8ltlCAAKuSAACkmwAAI50AAImfAAB7oQAAbqU
AAGKoAABWqwAASq4AAD+wAAA1swAAK7YAACW4CgAjuRYAlrogACG7KgAguzMAH7w+AB6
8SQAdvVYAHL1kABu+dAAavocAGL6bABa+sQAUvc8AFLvvABO5/wETuP8CErf/AhK3/wlSt/8Crp
EAAKabAACdogAAkKUAAIKoAAB0qwAAZq4AAFiwAABLsgAAQLUAADa4AAAAsuwAAI74AABvC
AAAUxQQADsgMAA3JFwAMyiEAC8orAAvLNwAKzEQACcxTAAjNZAAHzXcABs2MAAXNogADz
LsAAMvcAADK9AAByp8AAcf/AAHH/wABx/8AqZsAAKCKAACUqQAAh60AAHewAABosgAAWbQ
AAEy3AABAUGAANr0AACzBAAAJxAAAGsgAABPLAAALzwAABNIAAADUCgAA1RMAANYdAAD
ZJgAA2jIAANw/AADdUAAA3mIAAN93AADfjgAA36UAAN++AADe3AAA3vEAANz5AADc+QAA3P
kAo6QAAJerAACKrwAAerMAAGq1AABauAAATLwAAD+/AAA0wwAAKscAACHLAAAYzwAAEdIA
AAnWAAAC2QAAANwAAADeAAAA4AUAAOINAADkFwAA5yEAAOKsAADsOgAA7ksAAO9fAAD
wdQAA8I0AAPGkAADxuQAA8NAAAPDeAADw3gAA8N4AmqwAAIyxAAB9tQAAbLkAAFy8AABM
wAAAP8QAADPJAAAozgAAHtMAABbYAAAN2wAABt4AAADgAAAA4wAAAOUAAADnAAAA6gA
AAOWAAADuBQAA8Q4AAPQZAAD3JQAA+jMAAP1FAAD+WwAA/nEAAP+IAAD/nAAA/6wAAP+
1AAD/tQAA/7UAj7IAAIC3AABvuWAAxR8AAE7EAAA/yQAAMs8AACbVAAAc2wAAE98AAArjAAAC
5gAAAOgAADqAAAA7AAAAO4AAADvAAAA8gAAAPQAAAD3AAAA+QAAAPwFAAD/EQAA/x4
AAP8sAAD/PgAA/1QAAP9pAAD/fAAA/4wAAP+VAAD/IQAA/5UA/wAjBf8AHwT/AB0E/wAfA/8AJA
L/ADAB/wA+Af8ASwD/AFkA/wBIAP8AcAD/AHoA/wCCAP8AiQD/AJAA/gCWAP0AnAD8AKEA+w
CnAPoArgD5ALUA+AC/APcAywD1ANsA8gDpAO8A9QDuAP8A7AD/AOoA/wDqAP8A6QD/AOkA
/wDpAP8A/wAgBf8AHAT/ABoD/wAaA/8AlgL/AC0B/wA7Af8ASAD/AFUA/wBiAP8AbQD9AHYA+w
B+APoAhgD4AlwA9wCSAPcAmAD2AJ4A9QCkAPQAqgDyALIA8QC7APAAxgDtANUA6gDIAOcA
8gDIAP4A4wD/AOIA/wDhAP8A4QD/AOEa/wDhAP8A/wAcBf8AGAT/ABYD/wAVAv8AHwL/ACoB/
wA2Af8AQwD/AFEA/gBdAPkAaAD2AHIA9AB6APIAggDxAIgA8ACPAO8AIQDuAJa7QChAOsAp
wDqAK4A6QC3AOcAwgDkANAA4QDhAN0A8ADbAPsA2QD/ANgA/wDWAP8A1QD/ANUA/wDVA
P8A/wAYBP8AFAP/ABED/wATAv8AHAL/ACYB/wAyAf8APwD8AEwA9gBYAPEAYwDuAG0A7AB
2AOoAfQDoAlQA5wCLAOYAkQDkAJcA4wCdAOEApADgAKsA3gCzANwAvgDZAMwA1QDeANI
A7QDPAPkAzQD/AMsA/wDJAP8AyAD/AMgA/wDIAP8A/wIUBP8CDgP/AAoC/wERAF8BGQH/ACI
B/wAtAPwAOQD0AEYA7QBTAOcAXgDIAGgA4gBxAOAAeQDeAIAA3QCGANsAjQDaAJMA2AC
ZANYAoADUAKgA0gCwANAAuwDNAMkAygLcAMgF7QDEB/wAwQn/AL4K/wC9C/8AvAr/ALwK/
wC8Cv8A/wcPA/8HBwL/BwUB/wwwMAf8NFAD/Ch0A+gYnAPEEMwDqAUAA4wFNAN4CWADaB
GMA2AVsANUGdADTB3sA0QiCANAJiQDOCo8AzAqWAMsLNgDJDaYAxw6vAMUPugDDEckAw
BLcALwV7gC4Fv0AtRf/AbMX/wGyF/8BsBj/ArAY/wKwGP8C/wwwJAv8MAAH/EQEA/xYHAP8WDg
D6FBcA7xAhAOUMLADeCzgA2A1GANMQUgDPEI0AzBRmAMkWbwDHGHcAxRI+AMQahADC
G4sAwBySAL8dmgC9HqIAux6rAlkftgC3IMQAtSDXALeI6wGulvoCqyL/A6gj/wSnl/8EpiP/BKYj/wS
ml/8E/xEDAv8RAAD/GgAA/x4AAPYdBQDvGg0A4xYXANoTIwDRGDEAyxxAAMcfTADDIVcBwC
NhAb0lagG7JnlBuiZ5ArgngAK2KlcCtSiOArMplgKyKp4CsCqnA64rsgOsK78DqizSA6cs5wSjLPcG
oC3/B54s/wedLP8HnCV/B5wr/weck/8H/xUAAf8ZAAD/IQAA6yQAAOIAADblgMA1xsLAM4gHAD
GJiAwAwCk6AbwrRwK4LVIDtS5cBLMvZQSxMGwFrzF0Ba0xewWsMolGqjKJBqkzkQanM5oHpjSj

B6Q0rgeiNbsHoDXNCJw14wiZNfQKlZT/C5U0/wuUM/8LkzP/C5Mz/wuTM/8L/xkAAP8gAADuKAA
A4C8AANYyAADPMQAAyysGAMMtFwC8MicAtzQ1Arl2QQSuN00GqzdXB6k4XwinOGcJpTlvCq
Q5dggqiOn0LoDqFC587jQudO5UMmzufDJk8qgyXPLYNITvIDZM73g2QO/IOjTr/D4s6/w+KOf8PiTn
/D4k5/w+JOf8P/xwAAP0mAADIMQAA2TkAAM49AADGPQAAwDgBALs3EgC0OylArj0wAqo+PA
WmP0gloz9SCqA/WwyeP2MNNEBqDppAcQ+YQHgQl0CAEJVAiBGUQJERkkCbEZBBphKOQLM
SjEDDEoIA2hKGQO8ThD/+FII//xOBP/8TgD7/E4A+/xOAPv8T/x8AAO4rAADfOQAA0UEAAMdGA
AC/RgAAuUEAALNADACtRB4AqEYsAqNGOAaeRkMKm0ZNDZhFVg+VRV4Rk0VmEpFFbRSQ
RXQUjkV8FYxFhBaKRY0WiUWXF4dFuxeERa8XgkW/GIBE1Rh9ROwYe0T9GHID/xd4Q/8XeEL/
FnhC/xZ4Qv8W/yMAAOoyAADaQAAAzEgAAMFNAAC5TQAAskkAAK1HCACnSxoAok0oApxNN
QaYTEAKIExJDpFLUhGOSloUi0piFolJaReHSXAZhUI3GoNJgBuCSYkbgEmUHH1Jnxx7SawceUi
8HHZI0R10SOocckj7HHFH/xtwR/8acEb/GnBG/xpwRv8a/ycAAOU5AADURgAAx08AAL1TAACzU
wAARFAAAKdOBACHURYAnFMIAZdTMgWSUjwKjIFGD4pQTxOHT1cWhE5eGYJOZRT/TWwcfU1
0HntNfB95TYUgd02QIHVMnCFzTKkhcEy5IW5MziFsS+cgakv6H2IL/x5pS/8daEr/HGhK/xxoSv8c9
isAAOE+AADQSwAAw1QAALIYAACvWAAAp1YAAKJUAACcVhMAI1kjAZJZLwSNWDoKiVdDD4
VVTBOBVFMXfINbGntSYh14UmkfdlFwIXNQeSJxUlljb1CNJG1QmSRrUKYkaFC2JGZPyyRkt+U
kY0/4lmJP/yFiTv8gYU7/H2FO/x9hTv8f8DAAAN1DAADMUAAAv1kAALZdAACrXAAAO1sAAJ1ZA
ACXWxAak14gAl5eLASJXTcJhFxBdn9bSRN7WVEXeFhYG3VXXx5yVmYhb1Vtl2xUdiVqVH8mZ
1SKJ2VUlidjVKQnYVOzJ15TyCdcU+MmXFL3JfTs/yNbUv8iW1H/IVtR/yFbUf8h7DUAANIHAADIV
AAAvF0AALJhAACnYQAAn18AAJhdAACTYAwAjmldAlIjKgOEYzUlf2E+DXpgRxJ2Xk4Xcl1VG29
bXB9rWmMiaFlrJWVZcydjWHwoYFiHKV5YICpcWKlqWlexKVhXxilWVvUeOVVb2JlVW/yRVVf8jVV
X/IIVV/yJVf8i6ToAANVMAADFWQAAuWEAAK5IAACkZAAAm2QAAJRiAACoZQkAiWcaAlVoJw
KAaDIge2c8DHZIRBFxY0wWbWJTG2lgWh9IX2EjYl5oJl9dcChcXHkqWlyEK1dckStVXKArU1yvKI
FbwypQW98pT1r0J09a/yVPWf8kT1n/I09Z/yNPWf8j5T8AANFRAADCXQAAt2YAAKtoAACgaAAAI
2gAAI9nAACJaQUAhGwXAlBtJQF7bTAFdmw5CnFqQhBsaUkVaGdQGmRmVx5gZV4iXGNIJvlib
ShWYncqU2GCK1FhjytPYZ0rTWGtKkxgwipKYN0pSl/zJ0pe/yZKXv8lSl3/JEpd/yRKXf8k4UUAAM
1WAAC+YgAAtGoAAKdsAACcbAAAk2wAAIprAACEbgEaf3ETAhtylgB2ci0DcXE2CGxwPw5nb0
YTY21NGF5sVBxaa1sgV2piJFNoidQaHQpTWd/KktnjCpKZ5sqSGerKUZnvyhFZtsnRGXyJkRk/y
VEY/8kRGL/JERi/yREYv8k3EoAAMhbAAC7ZwAAsW8AAKRwAACZcQAAj3EAAIRxAAB+dAAAE
HYOAHV4HgBxeCoCbHc0Bmd2PATidUQQXnRLFVlzURIVclgeUXBgIU5waCRLb3EmSG58J0Zuii
hFbpknQ26qJkJuviVAbtokP2zxJD9r/yM/av8jPmn/lj5p/yl+af8i1IEAAMRhAAC3bAAArXMAAKB0AA
CUdQAAi3UAAH53AAB3eQAACXsJAG19GQBqfiYBZX4wBGF9OQhcfEANWHtIEVR6ThZQeVUa
THhdHUI4ZSBGd28iQ3d6l0J3iCNAAd5cjP3eolj13vCA8dtgfO3TxIDpy/yA5cf8gOXD/IDLw/yA5cP8g0
FgAAL9nAAC0cgAAqHgAAJt5AACQeQAAhnoAAHp9AABvfwAAaYIDAGSEEWBhhSEAXoUsAlqF
NQVWhD0JUoNEDU6CSxFKglIVR4FaGESAYxpBgG0cP4B4HT2Ahh08gJYcO4CnGzmAuxo4f9Y
ZNN3wGjV7/xs0ev8bM3n/HDN5/xwzef8cyWAAALtuAACweQAAo30AAJd9AACLfgAAgYAAAHaC
AABohgAAX4kAAFqKDABXjBoAVYwmAFKMLwJOjDgFS4xACUeLRwxEi08PQYpXET6KYBQ8im
oVOop2FjmKhBY3ipQVNoqlFDWKuRMzidUSMofvEzCF/xUvg/8WLoL/Fi6C/xYugv8WwmkAALZ2
AACsgAAAnoIAAJKDAACHhAAAE4YAAHCIAABjIAAAWI8AAFCsAwBMkxIASpQeAEiVKQBFI
CQ5U6BECUQgc9IEoJO5RTCziUXAw2IGcNNJRzDjOUgQ4ylJEOMJSjDS+Utwwtk9MLLJHuDCq
P/w0pjv8PKI3/ECiN/xAojf8QvHIAALF/AACnhwAAmYgAAI2IAACBigAAAdYwAAGmPAABdkgAAUZ
YAAEEZAABAnAcAPJ0UADudHwA6nikAOJ4yATaeOgl0nkMDMp5MBTCeVgYunmEGLJ9uByuff
Qcqn40GKZ+gBieftAUlns8FJJzsBSOA/gcimf8lIZf/CSGX/wkhl/8JtnwAAKyHAACHjQAAIL0AAleOA

AB6kQAAbpMAAGKXAABWmgAAS54AAEChAAA3pAAALqYHACunEwAqpx0AKagnACioMAAnq
TkAJqIDACSpTgEjqVkBliqpnASGqdQEfqocBHqqaARyqrgEaqckAGajpARim/AIYpP8DF6P/BBej/w
QXo/8EsIYAAKiRAACcIAAAj5QAAIGWAABzmQAAZ5wAAAFqgAABPowAARKYAADqpAAAvrAAA
Jq4AAB6wAwAZsg4AGLIYABezlgAWsysAFbQ1ABS0QAATtUwAErVaABG1agAQtnwADraQAAY
1pQAKtb0ACbPeAAmy9gAKsP8ACq//AAqv/wAKr/8Aq5AAAKSaAACWmwAAiJwAAHqgAABsow
AAX6cAAFKqAABHrQAAO68AADGyAAAntAAAHrYABa5AAAPuwAACL4JAAS+EwADvx0AAAsA
mAALAMAABwT0AAMFLAADCWwAAwmwAAMKAAADClgAAwawAAMHHAAC/5gAAvvcAAL7/A
AC+/wAAvv8AppoAAJ6iAACPPAAAgacAAHkrAABkrgAAVbAAAEiyAAA8tQAAMbcAACe6AAAEv
QAAFr8AAA7CAAAGxQAAAMgAADJBwAAyhAAAMwYAADNIgAAzywAANE4AADTRwAA01kA
ANNtAADTgwAA05sAANKyAADSzQAA0eYAANH1AADR9QAA0fUAoKMAAJSoAACGrAAAd68A
AGeyAABXtAAASLcAADy6AAAwwgAAJsEAAB3EAAAUxwAAC8oAAATNAAAA0AAAAANMAAADV
AAAA1wIAANoKAADcEgAA3hwAAOEmAADIMwAA6EMAAOIWAADpbAAA6YQAAOicAADosgAA
58gAAOfeAADn3gAA594Al6oAAImuAAB6sgAAabUAAFi4AABJvAAAO8AAAC/EAAAlxwAAG8sA
ABLPAAAI0wAAANYAAADZAAAA3AAAAAN4AADhAAAA4wAAAOYAAADoAQAA6woAAO4VAA
DylAAA9S0AAPk9AAD6UwAA+2kAAPuAAAD7lwAA+qkAAPq5AAD6uQAA+rkAjlAAAH20AABsu
AAAW7wAAErAAAA8xQAALsoAACPOAAAZ1AAADtkAAAXdAAAA4AAAAOMAAADkAAAA5gAA
AOgAAADrAAAA7QAAAPAAAADzAAAA9QAAAPkBAAD8DAAA/xkAAP8nAAD/OAAA/00AAP9jA
AD/eAAA/4gAAP+XAAD/lwAA/5cA/wAeBP8AGwP/ABoC/wAbAf8AlgH/AC0A/wA7AP8ASQD/AFY
A/wBiAP8AbAD/AHUA/wB9AP8AhAD+AlS/QCRAPwAlgD7AJwA+gCiAPgAqAD2AK4A9AC2APE
AwQDwANAA7wDiAO0A8QDsAP0A6wD/AOoA/wDnAP8A5gD/AOUA/wDIAP8A/wAbA/8AFwP/AB
UC/wAWAf8AHwH/ACoA/wA3AP8ARQD/AFIA/wBeAP0AaAD7AHEA+QB5APgAgAD3AlcA9gCN
APUakwD0AJgA8gCeAPAApADuAKoA6wCyAOkAvQDoAMoA5gDdAOQA7QDjAPoA4QD/AN4A/
wDcAP8A2wD/ANoA/wDaAP8A/wAYA/8AEwL/ABAC/wASAf8AHAD/ACcA/wAzAP8AQQD8AE0A
+ABZAPUAZADzAG0A8QB1APAAfADuAIMA7QCJAOWAjdQqAJQA6ACaAOYAoADjAKcA4QCvA
N8AuADdAMUA2wDWANKA6ADXAPYA0wD/ANEA/wDPAP8AzgD/AM4A/wDOAP8A/wAUA/8AD
gL/AAoB/wAPAf8AGQD/ACMA/wAuAPcAOwDzAEgA8ABUAOWAXwDqAGgA5wBwAOUAdwDkA
H4A4gCEAOEAigDeAJAA3ACWANoAnADXAKMA1QCrANMAAtADrameAzgDRAMwA5ADJAPQ
AxwD+AMUA/wDDAP8AwgD/AMIA/wDCAP8A/wAPAv8ACAH/AAUB/wAMAP8AFQD/AB4A+AAp
AO0ANQDpAEIA5gBOAOIAWQDfAGMA3ABrANoAcgDYAHkA1QCAANMAhgDRAIwAzWCSAM0
AmADLAKAAyQCoAMcAsQDFAL0AwwDNAMAA4QC+APEAuWd9ALkA/wC3AP8AtgD/ALYA/wC
2AP8A/wAIAv8AAQH/AAEA/wIAP8AEAD9ABgA7gAjAOMALwDfADwA2gBIANYAUwDSAF0AzW
BmAM0AbgDLAHUAYQB7AMcAgQDFAYgAwwGOAMICIQDAAP0AvgOmALwErwC7BbsAuAbMA
LYI4ACzC/MAsAz/AK4N/wCrD/8AqhD/AKkP/wCpD/8A/wUCAf8BAAD/CAAA/wsBAP0JCADzBBE
A4gAbANoAJwDUATQAZgNBAMoGTQDHB1cAxApGAMILaADADXAavg53ALwQfQC7EIQAuRG
LALgSkgC2E5sAtBSjALMVrQCxFroArhbLAKwY4ACoGfIAPRr/AKlb/wGhG/8BoBr/AaAa/wGgGv8
B/wkAAf8KAAD/EQAA8BIAAOMQAADeCgUA2QYRANAjHwDJDs0AxBI6AL8WRwC8GFIAuRpb
ALcbZAC1HGSAsx1yALleeQCwH4AAryCHAK0hjwCsIzCAqiKgAKgjqgCml7YApCTGAKIk3AGeJe
8CmyX+Apkl/wOYJP8DlyT/A5Yj/wOWI/8D/w0AAP8SAADvGAAA4h8AANKgAADRHQAazRUIAM
YXGAC+HCcAuSA1ALUjQgCxJU0AryZXAAo0XwGrKWcBqSluAqgqdQKmk3wCpSuDAqMsiwKiL
JMcOC2cA54tpwOcLrIDmi7CA5gu2AOULuwFki38BY8t/waOLP8GjSz/Bo0s/waNLp8G/xEAAPoZ
AADmJAAA2SwAAM4uAADGKwAAwSYCALwkEwC1KCIAsCwwAKwuPAGoL0gCpjBSA6MxWgS
hMmIEnzNpBZ4zcAWcNHcFmzR+Bpk1hgaYNY8GijWYB5Q1oweSNa8HkDW+B4410wiLNekliDT

6CYY0/wmENP8JgzP/CYMz/wmDM/8J/xQAAO4hAADfLgAA0TYAAMY5AAC9NwAAtzIAALMuDQ
CtMx4AqDYrAaQ3OAKgOEMEnTINBpo5VgeYOI4IljplCZU6bAmTOnMKkjp6CpA7ggUOO4sLjTuV
C4s7oAyJO6sMhju6DIQ7zgyBO+YNfjr4DXw6/w17Of8Nejn/DXo4/w16OP8N/xoAAOgqAADYNwA
Ayj4AAL9BAAC2PwAAsDsAAKs4CACmOxkAoT4nAZw/NAOYQD8FIT9JCJI/UQqQP1kLjj9hDIw/a
A2KQG8OiUB2D4dAfg+FQlcQg0CREIFAnBB/QKgRfUC3EXpAyhF4P+MRdT/2EXQ+/xFzPv8Qcj
3/EHI9/xByPf8Q9h8AAOMxAADSPgAAxUYAALIIAACwRgAAqkIAAKVAAwCgQhUAmkUkAJZGM
AOSRjsGjkVFCYtFTQyIRVUOhkVdEIREZBGCRGsSgERyE35EehR8RIMUe0SNFXhEmRV2RKU
VdESzFXFDxxZvQ+AWbUP1FWtC/xRrQv8TakH/E2pB/xNqQf8T8CUAAN83AADNRAAAwEwAAL
RNAACrTAAApEkAAJ9GAACaSRiAIUshAJBMLQOMTDgGiEtBCoRLSg2BSIIQf0pZENxJYBR6S
WcVeEhvFnZldxd0SIAYckiKGXBllhuSKIZa0ixGWIHxBpnR90ZZUfzGGRG/xdjRv8WY0X/FWNF/x
VjRf8V7SsAANo9AADJSQAAvVEAALBSAACmUQAAN08AAJpMAACVTg4AkFEeAltSKgKHUjUG
glE+Cn9QRw57T08ReE5WFHZNXRZzTWQYcUxrGW9MctxsTHwcakyHHGhMkx1mTKAdY0uuH
WFLwR1fStodXkrxG11K/xpcSf8ZXEn/GFxJ/xdcsf8X6TAAANZCAADFTgAAulUAAKxWAACiVQA
Am1QAAJVRAACQUwoAi1YbAldXJwKCVzIFfVY8CnIVRA52VEwRclNTFW9SWHdsUWEaalFoH
GdQcB1IUHkeY1CEH2FPkCBeT50gXE+sH1pPvh9YTtgfV07wHVZO/xxWTf8aVk3/GVZN/xlWTf8
Z5jYAANJGAADCUGAAtlkAAKIZAACfWQAAI1gAAJFWAACLWAcAhloYAIJcJQF9XDAEEVs5CX
RaQg1wWUkRbVhQFWIXVxhmVI4bY1VmHWFUbh9eVHcgXFOBIVpTjSJYU5siVIOqIVRTvCFRU
tUgUVLuH1BR/x1QUf8cUFD/G1BQ/xpQUP8a4jsAAM5LAAC/VwAAs10AAKZdAACbXQAAk1wAA
IxaAACGXQQAQgV8VAH1hlgB5YS0DdGA3CHBfPwxrXkcQZ11OFGRcVRhhW1wbXVpjHltZayBY
WHQhVlh+IINyiyNRWJkT1eolk5XuiJMV9MhS1btIEtV/h5LVf8dS1T/HEtU/xtLVP8b3kAAAMPQAA
C8WwAAr2AAAKJhAACYYQAAj2AAAIdfAACBYgAAfGQRAHhmHwB0ZisCb2U0BmtkPQtmY0QP
YmJME19hUhdbyFkaWF9hHVveaSBsXXehUF18I01diCNMXZcjSl2mlkhcuCFGXNEhRVvsH0Va
/R5FWf8dRVn/HEVZ/xxFWf8c2UUAAMZVAAC5YAAAq2QAAJ9kAACUZQAAi2UAAIJkAAB7ZwA
AdmkNAHJrHABuaygBamsxBWZqOglhaUINXWhJEVlnTxVWZIYZUmVeHE9kZh5MY28gSmN6IU
hjhiJGY5UhRWoIIUnityBBYs8fQGHrHkBg/B1AX/8cQF7/HEBe/xtAXv8b1EoAAMJaAAC2ZQAAq
GgAAJtoAACQaQAAh2kAAHxpAAB1bAAAb24JAGtwGABocSQAZHEuA2BxNwdccD8LWG9GD1
RuTBNQbVMWTWxbGUprYxxHamweRWp3H0NqhB9BapMfQGqjHj5qtR09ac4cPGjqGztm/Bs6Zf
8bOmT/Gjpk/xo6ZP8az1EAAL5gAACzagAApGwAAJdsAACMbQAAGm4AAHZvAABucgAAaHQD
AGR2EwBhdyAAXncrAlp3MwVWdzslUnZCDE91SQ9LdFASSHNYFUVzYBhCcmoaQHJ1Gz5ygh
s9cpEaO3KiGTpytBg4ccwXN3DpFzZu+Xg1bf8YNWv/GDVr/Xg1a/8YyVgAALpmAACubwAAAn3AA
AJNxAACIcQAAfnMAAHJ1AABmeAAAYHoAAft8DQBYfhsAVn4mAFN+LwJPfjcFTH4/CEI9RgtFf
E0OQnxVEUB7XhM9e2cUO3tzFTI7gBU4e48VN3ugFDV7shM0essSM3noEjF3+hMwdf8TL3T/FC
9z/xQvc/8Uw2AAALZtAACpdAAAmnUAAI51AACDdgAAeXgAAG16AABhfgAAWIEAAFKDBgBOh
RQATIYfAEqGKQBHhJlCRIY6BEKGQqC/hUkJPIVRCzqFWg04hWQONoRwDzSfQ8zhY0OMY
WeDTCEsA0uhMkMLYPnCyuA+g0qfv8OKX3/Dyl9/w8pff8PvWgAALF1AACkeQAAIXoAAII6AAB/f
AAAc34AAGiAAABbhAAAUocAAEmLAABDjQsAQI0XAD+OIgA9jisAO48zATmPPAM3j0QENY5M
BjOOVgcxjmAIL45sCC2Peggsj4oIk4+bBymOrgYojsYGGJ43IBSWK+Qcjf8JI4f/CiKH/woih/8Kt3EAA
K19AACefwAAKH8AAIWAAAB5ggAAbYQAAGGHAABWiwAAS44AAEKRAAA5IAAAM5YNADGX
GAawlyIAL5crAC2YMwAsmDwBKphGASmYTwlnmFsCJplnAyWZdQMjmYUCIpmXAIcZqglfmMI
BHZfiARyV+AMbk/8EG5L/BRqR/wUakf8FsnsAAKeEAACYhQAAjIUAH+GAAByiQAAZowAAFqP
AABPkgAARJYAADuZAAAxnAAAKZ4AACKgCwAgoRYAH6EfAB6iKAAdojEAHKI6ABujRQAao1E
AGaNeABijbAAXo30AFaOQABOjpAARo7sAD6LcABCg9QAQnv8AEJ3/ARCd/wEQnf8BrlUAAKG

LAACtiwAAhowAAHmOABrkQAAX5QAAFOYAABImwAAPZ8AADOiAAAqPAAAlacAABmpAAA
SggYADasRAAysGgAMrCMAC6wtAAqtNwAJrUMACK1RAAetYAAFrXEABK2EAAKtmQAARA4AA
KzJAACr6AAAqvkAAKn/AACo/wAAqP8AqJAAAJuSAACOKgAAf5QAAHGXAABkmwAAV58AAEui
AABApGAAAnqkAACyrAAAirgAAGq8AABKxAAAKswAAArUHAAC1EAAAthkAALchAAC3KgAAuD
UAALICAAC5UQAAuWIAALI1AAC6igAAuaAAALm3AAC31QAAAt+0AALX7AAC1/gAAtf4ApJkAAJa
ZAACHmwAAeJ4AAGqiAABcpgAAT6oAAEOuAAA3sAAALLIAACK0AAAZtgAAEbgAAAm6AAB
vAAAAL4AAAC/BQAAwA0AAMIvAADDHgAAxCcAAMYzAADIQAAAYVEAAMIkAADKeQAAYpAA
AMqmAADKvgAAydkAAMjsAADi8QAAYPEAnqEAAI+jAACApGAAcaoAAGOuAABTsQAARbMAA
Di2AAAsuAAAIrsAABi+AAAQwAAAB8IAAADEAAAAxwAAAMoAAADMAAAAazQAAAM8HAADRE
AAA1BkAANYjAADaLwAA3T4AAN5RAADfZQAA4HwAAOGTAADhqQAA4b0AAOHSAADg2QAA
4NkAlKgAAIasAAB3rwAAZrIAAFW1AABGuAAALwAACu/AAAhwgAAF8UAAA3IAAAEywAAAM
4AADRAAAA1AAAANYAAADaAAAA3AAAAAN8AADiAAAA5QYAAOkQAADsHAAA7ykAAPI5A
ADzTgAA82YAAPR9AAD0kwAA9aYAAPW1AAD1ugAA9boAia4AAHqyAABptQAAWLgAAEe8AA
A4wQAAK8UAACDJAAAVzQAAC9IAAAHVAAAA2QAAAN0AADfAAAA4QAAAOMAAADIAAAA6
AAAAOsAAADuAAAA8QAAAPUAAAD4CAAA/BQAAP8iAAD/MgAA/0kAAP9hAAD/eAAA/40AAP+
cAAD/oAAA/6AA/wAaAv8AFwL/ABUB/wAXAP8AHwD/ACsA/wA5AP8ARgD/AFMA/wBeAP8AaAD
/AHAA/wB4AP4AfwD8AIUA+gCLAPgAkAD2AJUA9ACbAPMAoQDyAKgA8QCwAPAAugDvAMgA
7QDbAOwA7ADpAPoA5gD/AOYA/wDIAP8A5QD/AOQA/wDkAP8A/wAWAv8AEwH/ABEB/wARA
P8AHAD/ACgA/wA1AP8AQgD/AE4A/gBaAPsAZAD5AGwA+AB0APYAewDzAIEA8QCHAO8AjAD
tAJIA7ACXAOsAngDpAKUA6ACsAOcAtgDIAMMA4wDUAOEA5wDdAPYA3AD/ANoA/wDZAP8A2
AD/ANcA/wDXAP8A/wASAv8ADQH/AAoB/wAOAP8AGQD/ACQA/wAwAPwAPQD5AEkA9gBVAP
MAXwDwAGgA7gBwAOsAdgDpAHwA5gCCAOQAIADjAI0A4QCTAOAAmgDfAKEA3QCoANsAs
QDZAL0A1gDNANMA4QDQAPEAzgD9AM0A/wDKAP8AyQD/AMkA/wDJAP8A/wANAv8ACAH/A
AQA/wALAP8AFAD/ACAA9gArAPIAOADvAEQA7ABPAOGAWgDIAGMA4gBqAN8AcQDcAHgA2g
B+ANgAgwDWAika1QCPANMAIgDRAJ0AzwCkAM0ArQDLALgAyADHAMUA2gDDAOwAwQD6A
L4A/wC9AP8AvAD/ALwA/wC8AP8A/wAJAf8AAQD/AAEA/wAIAP8AEAD1ABoA7QAmAOgAMgDk
AD4A4ABKANwAVADYAF0A0wBIANAAbADOAHMAzQB5AMsAfwDJAIUAYACLAMYAkGDEAJkA
wwChAMAAqgC+ALQAuWDCALkA1AC3AOgAtQD3ALMA/wCyAP8ArwD/AK4A/wCuAP8A/wACAf
8AAAD/AAAA/wADAPwACgDpABQA4wAgANwALADXADgA0gBDAM0ATgDKAFcAxwBgAMQAZ
wDCAG4AwQB0AL8AewC+AIEAvACHALsAjgC5AJUAtwCeALUApwCzALEAsQC/AK8A0QCsAO
YAqgH2AKcE/wCIBP8ApAX/AKQE/wCkBP8A/wAAAP8AAAD/AAAA9wAAAO4AAQDgAAwA2AAZ
ANAAJQDKADEAxQA9AMEASAC+AFIAvABbALoBYgC4AmoAtgNwALUEdgCzBX0AsgaDALEHi
wCvCJMArQmbAKwKpQCqC7AAqAy/AKYM0gCjDuoAnxH6AJ0R/wCbEv8AmhH/AJoR/wCaEf8A/
wAAAP8CAADzBwAA5QsAANwLAADVBAIAzgAQAMcBHQDABCKAvAg2ALgLQgC1DUwAshBV
ALARXgCuE2UArBRsAKsVcgCqFnkAqBaAAKcXhwCIGI8ApBmYAKIZowCgGq4Anhu8AJsbzwC
YHOYAihz4AJMc/wGSHP8BkRz/AZAb/wGQG/8B/wMAAPoKAADoFQAA2xsAANAcAADHGAAA
wxAFAL4NFQC3EyMAshcwAK4aPACrHEcAqB5RAKYfWQCkIGEAOyFoAKEibgCgl3UAniN8AJ0k
hACbJYwAmiWVAJgmoACWJqsBICa5AZImzAGPJMBjCb2Aoom/wKIJf8DhyX/A4YI/wOGJf8D/w
gAAO4WAADflgAA0igAAMUpAAC9JQAAtyAAALQbDgCuHx4AqSMrAKUmNwCiKEMAnyIMAZ0q
VQGbk10BmSxkApgsagKXLXECIS14ApQtgAKSLogDkC6SA48unAONL6gDiy+1A4gvYASFLuAE
gi7zBYAu/wV+Lf8FfSz/BX0s/wV9LP8F/A8AAOggAADYLAAAYTMAAL0zAAC0MAAArywAAKsnC
QCmKhkAoS0nAJ0wMwCaMT4BlzJIApQzUQOSM1gEkTNgBI80ZgWONG0FjDR0Bos1fAaJNYU

GhzWOBou1mQeDNqUHgTayB341xAd8NdwlEtxXCHc0/wh1M/8ldDL/CHQy/wd0Mv8H8xcAAOI
pAADRNAAAwzsAALY7AACTOQAAPzUAAKQxAWCfMxUAmjYjAJY4LwGSOTOCjzIEBlw5TAWKO
IQHiDpbB4Y6YgiFOmkJgzpwCYE7eAqAO4EKfjuLC3w7lQt6O6lLeDuvC3U6wQxzOtkMcDrvDG45
/wttOP8LbDj/C2w3/wpsN/8K7h8AANwwAADLPAAAvUEAALBBAACnQAAAOtWAAJ06AACyOHE
AID0fAl8/KwGLPzYDiD9ABYU/SAeDP1AJgD9YCN4/Xwt8P2YMez9tDXk/dA53P30OdT+HD3M/kg
9xP58Pbz+sD2w/vRBqPtUQZz7tD2Y9/g9lP8OZDz/DWQ8/w1kPP8N6iYAANc2AADGQgAAuEYA
AKtGAACiRQAAnEIAAJdAAACsQqWAjkMcAlpFKAGFRTMDgkU8Bn5FRQI8RU0LeURUDXdEW
w51RGIPc0NpEHFDcRFvQ3oSbUOEEmtDjXNPQ5wTZkOqE2RDuhNhQtITX0LrEl5B/BFdQf8QX
UD/EF1A/w9dQP8P5isAANI8AADCRwAAteAAKdLAACeSgAAI0gAAJJFAACNRgkAiEkZAIRKJ
QGASzADfEs5BnhKQgl1SUoMc0lRdnBIWBBuSF8SbEhmE2lHbhRnR3cVZUEBFmNHjBZhR5kV
X0enFlxGuBZaRs8WWEbpFvDF+xRXRf8SV0T/EIZE/xFWRP8R4jEAAM5BAAC/TAAAsE4AAKR
PAACaTwAAK00AAI5KAACITAUAg04VAH9QlG7UC0Dd1A3BnNPPwlwTkcMbU5OD2pNVRFnT
FwTZUxjFWJMaxZgS3QXXkt+GFxLiRlaS5cZWEulGVVKthhTSswYUknnF1FJ+hVRSP8UUUj/E1
BH/xNQR/8T3jYAAMpGAAC8UAAARVIAAKBSAACWUGAAj1EAAII0AACDUQIAfIMSAHpVIAB2V
SsCclU0BW5UPQlqU0QMz1JLD2RSUhJhUVkUXIBgFlxQaBhaT3EZV097GIVPhxpTT5QaUU+jG
k9OtBpNTsoZTE3mGEtN+RdLTP8VS0z/FEtL/xRLS/8U2jsAAMdKAAC5VAAAqVUAAJ1WAACTV
gAAi1UAAIRTAAB+VQAAeVcPAHVZHQBxWigBbVoyBGIZOghIWEILYldJD15XUBJbVlcUWVVeF
1ZUZhITVG4aUVN5G09ThBtNU5lbS1OhG0ITshpHU8gaRILkGUZR+BdFUP8WRVD/FUVP/xRFT
/8U1kAAAMNPAAAC1VwAAPlkAAJZAACPWQAAh1kAAH9XAAB4WgAAc1wLAG9eGgBsXyYBaF
8vA2ReOAdgXj8KXV1GDllcTRFWW1QUU1pbFIBZYxhOWWwaS1I2G0IYghtHWJAbRligG0RYsB
pCWMcZQVfjGEBW+BdAVf8WQFT/FUBU/xVAVP8V0UUAAMBTAAcYwWAAolwAAJZdAACMX
QAAG10AAHlcAABzXwAABWEHAGljFgBmZCIAY2UsAl9kNQVbYzwJV2NEDFRiSw9RYVISTmBZ
FUtfYRdlX2oZRI50GkRegBpCXo4aQV6eGT9erxg9XsUXPF3iFztb9xY7Wv8VO1n/FTtZ/xQ7Wf8U
zEsAAL1ZAACuXwAAn2AAAJJhAACIYQAAfmlAAHNhAABsZAAAZ2cDAGJpEgBfah8AXGopAVI
qMQRVajkHUmlBCk9oRw1LaE4QSGdWEkZmXhRDZmcWQWVyFz9lfhc9ZYwXPGWcFjplrhU5Z
cMUOGTgFDZi9hQ2YP8TNI//EzVf/xM1X/8Tx1EAALlEAAcPywAAmmQAAI5IAACEZQAAemYAA
G5oAABlagAAX20AAftvDQBYcBoAVXEIAFJxLgJPcTYETHA9B0lwRApGb0sMQ25TD0BuWxE+
bWUSPG1vEzptfBM5bYoTN22bEjZtrBE0bclQM2zfEDFq9RAwaP8QMGf/ETBm/xEwZv8RwlgAAL
VIAACIZwAAImgAAIppAAB/agAAAdWsAAGItAABfcAAAV3MAAFN1BwBPdxQATXcgAEt4KQBleD
ECRXg5BEN3QAZAd0gIPXdQCjt2WAw5dmINN3ZtDjV2eg4zdogOMnaZDTF2qwwwvdcAMLnXcY
xy9AwrcP8NKM//DSpu/w4qbv8OvV8AALFrAACgbAAAKm0AAIVuAAB7bwAAcXAAAGVzAABadg
AAUXkAAEI8AABFfg0AQ38ZAEF/lwA/gCwAPYA0ATuAOwM5gEMEN39MBjR/VAcyf14IMH9qCS
9/dwktf4YILH+XCct/qQcpf74GJ37cBiZ79Aclef8IJHj/CSN3/wojd/8KuGgAAKpwAACacgAAjXIAAIF
zAAB2dAAAAa3YAAGB5AABUfQAAS4AAAEODAAA7hgQAN4cRADWHGwA0iCQAM4gtADGJNQ
AviT0BLolGAiyJTWMqiVoDKYIIBCeJcwQmiYIDJYmTAyOJpgMhiLsCH4jZAh+F8gMeg/8EHYL/BR
yB/wYcgf8Gs3AAAKR2AACVdwAAiHcAAH14AABxegAAZXwAAFP/AABPgwAARYcAADyKAAAzj
QAAK48GACeQEQAmkRsAJZEKACSRLAAjkjQAPl+ACGSSAAgklMAHPnFAB2TbQAck3wAGpO
OABmToQAXkrYAFJHUABSP8AAUjf8BFiz/AhOM/wlTjP8CrnoAAJ58AACPFAAAg30AAHd+AAABq
gQAAXoQAAFOHAABJiwAAPo8AADWRAAAslAAAJJcAABYzBQAXmg8AFpsYABWbIQAUmykA
E5wyABKcPAARnEgAEJxVAA6dYwANnXMAC52FAAmcmQAHnK4ABZrIAAWZ5wAGmPoABpf/
AAaW/wAGlv8Ap4MAAJeDAACLgWAAfoQAAHCGAABjiQAAV40AAEYQAABBIAAAN5cAAC2aAA
AknQAAHJ8AABWHAANNowQABqQMAAKIFgACpR4AAaYnAACmMQAAPjwAAKZJAACmVwAA

pmcAAKZ5AACmjQAAPqMAAKW5AACk1wAAo/AAAKL8AACi/wAAov8AoIoAAJKJAACFigAAAd4w
AAGmPAABckwAAUJcAAESaAAA5ngAAL6EAACakAAAdpwAAFakAAA2qAAAFrAAAAK0EAACu
DAAArxUAAK8dAACwJQAAsS8AALE7AACySQAAslkAALJsAACygAAAspYAALGsAACxxQAAsO
IAALDyAACv+gAAr/oAmpEAAI2RAAB+kgAAcJYAAGKaAABUngAASKIAADymAAAYqQAAJ6wA
AB6vAAAVsQAADLIAAAS0AAAAAtQAAALcAAAC4AgAAuQkAALoRAAC7GQAAvSIAAL4sAADAQ
QAAwUkAAMJbAADCbWAAwoUAAMKdAADCSwAAwc0AAMHjAADA7gAAwO4AIzgAAIaaAAB4n
QAAaaEAAFumAABNqwAAQa8AADSyAAAotAAAHrYAABS4AAAKugAAArwAAAC+AAAAwAAA
AMEAAADDAAXQAAMcEAADJDAAyxAAM4eAADRKQAA1TYAANZIAADYXAAA2XIAAN
qKAADaoQAA2bUAANnJAADY2QAA2NkAj6IAAICIAABxqgAAY68AAFKyAABCtAAANLcAACe7A
AAAdvQAAE8AAAAAnDAAAAxQAAMcAAADJAAAAyWAAAM4AADQAAAA0wAAANUAAADYAA
AA2wUAAN8OAAAdjGQAA5yUAAOw0AADtSAAA7I0AAO90AADWjAAA8aAAAPGvAADxuQAA8bk
AhqwAAHevAABmsgAAVLUAEE05AAA1vQAAJ8EAABzEAAASyAAAB8wAAADOAAAA0QAAAN
QAAADXAAAA2gAAAN0AADdgAAAA4wAAAOYAAADpAAAA7AAAAO8AADyBgAA9hMAAPoh
AAD9MgAA/kcAAP9dAAD/dAAA/4gAAP+XAAD/oQAA/6EA/wAWAf8AEgH/ABIA/wASAP8AHQD/
ACkA/wA2AP8AQgD/AE8A/wBaAP8AYwD+AGwA+wBzAPkAeQD3AH8A9gCFAPUAIwD0AJAA8
wCWAPIAnADxAKMA7wCrAO4AtADsAMAA6gDRAOcA5gDmAPYA5QD/AOMA/wDiAP8A4gD/A
OEA/wDhAP8A/wASAf8ADQH/AAAsA/wANAP8AGQD/ACYA/wAyAP8APgD/AEOA/ABVAPkAXwD
1AGcA8gBuAO8AdQDuAHsA7QCBAOWAhwDqAlwA6QCSAOgAmADnAJ8A5QCnAOMArwDhAL
sA3gDLANwA4ADaAPEA2AD/ANYA/wDUAP8A0gD/ANAA/wDQAP8A/wANAf8ACAD/AAQA/wAK
AP8AFgD/ACEA/QAtAPkAOQD3AEUA8wBQAO4AWgDqAGIA5wBqAOUAcQDjAHcA4gB9AOAA
ggDfAlgA3QCOANwAlADaAJsA2ACiANUAqwDSALUA0ADEAM4A2ADMAOWAyQD7AMcA/wDE
AP8AwgD/AMAA/wDAAP8A/wAHAf8AAgD/AAAA/wAHAP8AEQD4AB0A8wAoAO8ANADrAEAA5
wBLAOEAVADdAF0A2wBIANgAbADWAHIA1AB4ANIAfgDRAIMAzWcJAM4AjwDLAJYAyQCeAM
cApwDEALEAwgC+AMAA0AC+AOYAuwD2ALgA/wC1AP8AtAD/ALMA/wCzAP8A/wACAP8AAAD
/AAAA/wACAPUACwDuABcA6AAjAOIALgDcADoA1wBFANMATwDPAFgAzABgAMoAZwDIAG0A
xgBzAMUAeQDDAH8AwQCFAMAAiwC+AJIAvACaALoAowC4AK0AtgC5ALQAygCxAN8ArgDwA
KsA/gCqAP8AqQD/AKgA/wCoAP8A/wAAAP8AAAD/AAAA/gAAAOsABQDjABEA2wAdANMAKAD
NADMAyQA/AMYASQDCAFIawABaAL0AYgC7AGgAugBuALgAdAC3AHoAtQCAALQAhwCyAI4
AsACWAK8AnwCtAKkAqwC1AKgAxQCIANoAowDuAKEA+wCgAP8AnwD/AJ4A/wCeAP8A/wAA
AP8AAAD5AAAA6QAAAOEAAADWAAoAzQAWAMYAlgDBAC0AvQA4ALoAQwC3AE0AtABVALI
AXQCwAGMArWbqAK4AcACsAHYAqwB8AKoAgwCoAlSapgCTAKUANQCjAacAoAKzAJ4DwwCc
BdoAmgbtAJCh/QCVB/8AIAf/AJMH/wCTB/8A/wAAAPsAAADqAgAA3wcAANIFAADKAAEAwwAO
AL0AGgC4ACYAtAAyALACPQCtBEcAqwZQAKkIWACnCV8ApgplAKQLbACjDHIAogx5AKANgA
CfDogAnRCRAJsQmwCZEaYAlxGzAJURxACTetsAkBPwAI0U/wCLFP8AihT/AIkT/wCJE/8A/wAA
AO8HAADhEwAA0xgAAMUXAAC9EgAAuQoEALUFEgCwCR8Aqw0rAKcRNwCkE0EAohVLAKA
WUwCeF1oAnBhhAJsZaACZGm4AmBp1AJcbfQCVHIUakxyOAJIdmACQHaQAjh2wAlsewQCJHt
gAhh7uAlMe/gGBHv8BgB7/AX8d/wF/Hf8B+AIAAOgUAADZIAAAyCQAALWjAACzIAAArxsAAKwU
CgCnFxoAohomAJ4dMgCbHz0AmSFHAJciTwCVI1cAkyNeAJIkZACQJWsAjyVyAl4meQCMJoIAii
eLAYgnlQGGJ6EBhCiuAYIovgF/J9QBfCfsAnon/AJ4Jv8DdyX/A3YI/wN2Jf8D8g4AAOEfAADQKgA
AwC0AALMtAACrKgAApiYAAKMhBACfIHUAmiUiAJYnLgCTKTKAkSpCAI4rSwGMLFMBiyxAykt
YQKILWcChi5uAoQudgODLn4DgS+IA38vkgN9L54Dey+rBHkvuwr2L9EEcy7pBHEt+wVvLf8Fbiz/
BW0r/wVtK/8F7RcAANsoAADJMwAAuTUAAK01AACIMwAAAnzAAAJssAACYKxAAky4eAl8wKgC

MMTUBiTI+AOYzRwKEM08DgjRWBIA0XQR/NGQFFtVrBXw1cgZ6NXsGeDWEBnY1jwZ0NZsHcj
WoB281uAdtNM0HajTnB2gz+QdnMv8HZjL/B2Ux/wdlMf8H6B8AANUvAADEOQAAAszsAAKc7AA
CfOgAAmTcAAJU0AACRMwsAjDUaAlg3JgCFOTEBgjk7A385QwR8OksFejpSBng6WQd3OmAld
TpnCHM6bwIxOncJbzqBCm46jAprOpgKaTqmCmc6tQpkOcoLYjnkCmA4+ApfN/8KXjf/CV42/wleN
v8J4yYAAM82AAC+PgAAr0AAAKNAACaPwAAIDwAAI86AACLOgcAhjwXAll+lwB+Py4Bez83A3
g/QAV1P0gHcz9PCHE/VglvP10KbT5kC2s+bAxpPnQMZz5+DWU+iQ1jPpUNYT6jDI8+sg5cPccO
Wj3iDvK89gxYPP8MVzv/C1c6/wtXOv8L3ysAAMs7AAC6QgAAq0QAAJ9FAACWRAAAj0EAAIo/A
ACFPwQAqEETAHxDIAB5RCsBdUQ0A3JEPQVvREUIbURMCWpDUwtoQ1oMZkNhDWRDaQ5i
Q3EPYEJ7EF5ChhBcQpMQWkKhEFdCsBBVQcUQU0HgEFJA9Q5RQP8OUT//DVA+/wxQPv8M
2zEAAMdAAAC2RgAAp0gAAJtIAACSSAAAI0YAAIVEAACARQAAe0YQAHdIHQB0SSgBcEkyA2
xJOgZpSUII20hJCmRIUAxiSFcOX0deD11HZhBbR24RWUZ4EldGgxJVRpATU0afEIFGrhJORcIS
TEXeEUxE9BBLQ/8PS0P/DktC/w5LQv8O1jYAAMRFAACySQAAPesAAJhMAACOSwAAh0oAAIF
IAAB7SQAAdksNAHJNGwBuTiYBa04vA2dONwVkJ8IYU1GCI5NTQ1cTFQOWUxbEFdLYxJVS2
wTUKp1FFBKgRROSo4UTUqdFEtKrBNISsETR0ncEkZI8xFFR/8QRUF/D0VG/w9FRv8P0jsAAMFJ
AACvTQAAoE4AAJRPAACLTwAAg04AAHxLAAB1TgAAcFAJAGxSGABpUyMAZIMtAmJTNQVfU
z0HXFJECIIRSwxWUVEPU1BZEVFQYBJPT2kTTE9zFEpPfhVJT4wVR0+bFEVPqxRDT8TQU7a
E0FN8hFATP8QQEv/EEBK/w9ASv8PzkAAAL5NAACrUAAAnVIAAJFSAACHUgAAfIIAAHZQAAB
wUwAAa1UGAGdWFABkWCAAYFkqAV1YMgRaWDoGV1dBCVRXSAXRvK8OTIVWEETVXhJJV
GcTR1RxFEVUfBVDVloUQVSZFEBUqRM+U70SPFPZEjtS8RE7UP8QO0//EDtP/w87T/8PykUAA
LpRAACoVAAAmVUAAI1WAACDVgAAeIYAAHFVAABqWAAAZVoCAGBCEQBdXR0AW14nAVhe
LwNuxJcFUV0+CE5cRQpLXEwNSVtTD0ZbWxFEWmQSQVpuE0BaehM+WogTPFqXEjtaqBI5W
bwRN1nXEDZX8BA2Vv8PNIX/DzVU/w81VP8PxUsAALZVAACkVwAAIlkAAIpZAAB/WgAAAdloAA
GpaAABkXQAAXI8AAFphDABXYxkAVGQjAFJkLAFPZDQDTGM7BkljQghGYkkKQ2JRDEFhWQ4
+YWIQPGFsEDtgeBE5YIYRN2GWEDZgpw80YLoOM1/VDTFe7w0wXP8OMFv/DjBa/w4wWv8Ow
VEAALFaAACgWwAAklwAAIZdAAB7XgAAcl8AAGVgAABdYwAAV2YAAFJnBwBPaRQATWofAE
pqKABlajACRWO3BENqPwZAakYHPmIOCTtpVgs5aV8MN2hqDTVodg00aIQNMmiUDDFopQsva
LkLLWfUCixl7gorY/4LKmH/DCpg/wwwqYP8MvVcAAKxeAACbYAAAJWEAAIFhAAB3YgAAbWMAA
GFmAABYaQAAUWwAAEpuAQBGcA4ARHEZAEJxIwBACisAPnlzATxyOgM6cUIEOHFkBjVxUgc
zcVwIMXFnCTBxcwkucYIILXGSCCtXowcqcLcGKHDSBidu7QYla/4IJGr/CSRo/wkkaP8JuF8AAKZj
AACWZAAAIWUAAH1mAABYzWAAaWgAAF5rAABTbwAAS3IAAEN1AAA9dwcAOngTADh5HQA
3eSUANXotADR6NQEyej0BMHpFAi56TgMselgEKnpjBCI6cAQoen8EJnqPBCV6oQMjebUDIXnP
AiB37AMfdP0EHnP/BR1x/wYdcf8Gs2cAAKFpAACRaQAAhGoAAHhrAABubAAAY24AAFIxAABN
dQAARXgAAD17AAA1fgAALoAKACyBFQAqgh4AKYImACiCLgAngjYAJom/ACSDSAAJg1MBIoNe
ASCDawEfg3oBHYOLARyDnQAagrEAGILLABeA6gAXfwBFnz/AhZ7/wMWe/8DrG0AAJpuAACM
bwAAf28AAHRwAABpcgAAXXUAAFN4AABIfAAAP38AADaCAAAuhQAAJogAAB+KCgAdihQAHI
sdABuLJQAaiy0AGYw2ABiMPwAWjEoAFYxWABSMZAATjHMAEYyFAA+MmAAAnjKwACovEAAq
J4wAKiPgAC4b/AAuF/wALhf8ApXQAAJR0AACGdAAAe3UAAG92AABieQAAV3wAAEyAAABCh
AAAOlcAAC+KAAAnjQAAH48AABeRAAAQkwgADJQSAAuUGwAKISMACZUsAAiVNQAHIUAAB
pVNAASWWgADImoAAZV7AACVjgAAIaMAAJ55AACT1gAAku8AAJH7AACQ/wAAkP8AnXoAAI5
6AACCEwAAdnsAAGh+AABbgQAAUIUAAEWJAAA7JAAMZAAACeTAAAFIQAAF5gAABCZAAAI
nAIAAZ0KAACdEgAAAnhoAAJ8iAACfKwAAoDUAAKBBAACgTwAAoF8AAKBwAACghAAAn5kAAJ
6uAACeyAAANEUAAJz1AACb/wAAm/8AI4EAAIqBAAB9ggAAb4QAAGGHAABUiwAASI8AAD2TA

AAZlgAAKZoAAB+dAAAXnwAAD6EAAAejAAAApQAAAKYCAACnCQAAqBEAAKkYAAACpIAAAqi
kAAKs0AACsQQAARFEAAKxjAACsdgAArIsAAKuiAACqugAAqdcAAKnsAACo+AAAqPgAkYgAAI
WIAAB2igAAaI4AAFmSAABNIgAAQZsAADWfAAArrogAAaIaUAABioAAAPqgAABqwAAACtAAAAr
wAAALAAAACxAAAAsgYAALMNAAC1FQAAtH0AALcnAAC5MgAAukEAALpTAAC7ZwAAu3wAA
LuTAAC7qgAAu8EAALraAAC57AAAuewAjZAAAH6RAABvIQAAYJkAAFKeAABFowAAOacAAC2r
AAAJrgAAGrEAABCzAAAGtQAAALcAAAC4AAAAuQAAALsAAAC8AAAAvgAAAMAAAADCBwAA
xBAAAMYZAADJlwAAzTAAAM9AAADPVQAA0GoAANCCAADRmgAA0a4AANHCAADR1wAA0d
cAhpKAAHedAABooQAAWqYAAEusAAA/sQAAMLQAACS2AAAZuQAADrsAAAS9AAAAvwAAAM
EAAADDAAAAxQAAAMYAAADJAAAAyWAAAM4AAADQAAAA1AEAAngKAADcFAAA4SAAAOYt
AADoQAAA6VYAAOptAADrhQAA65sAAOusAADrvAAA67wAgKUAAHGqAABjrwAAUblAAEG2A
AAxuQAAI70AABnAAAAANwwAAA8YAAADJAAAAyWAAAM0AAADPAAAA0QAAANQAAADXAAA
A2gAAAN0AAADgAAAA5AAAAOgAAADsAwAA8BAAAPUcAAD6LAAA+0AAAPxXAAD9bgAA/oM
AAP+UAAD/ogAA/6IA/wASAf8ADgD/AAsA/wAOAP8AGgD/ACcA/wAzAP8APwD/AESa/wBVAP0A
XgD6AGYA+ABuAPYAdAD1AHoA9ACAAPMAhgDyAlsA8QCRAPAAIwDvAJ4A7QCmAoOArgDo
ALoA5wDKAOUA4ADjAPMA4gD/AN8A/wDaAP8A1wD/ANYA/wDWAP8A/wANAP8ACAD/AAQA/
wALAP8AFwD/ACMA/wAvAP8AOgD+AEYA+ABQAPMAWgDxAGIA7wBpAO0AcADrAHYA6gB8A
OkAgQDoAlcA5gCNAOUAkWdJAJOA4ACHAN4AqgDcALQA2gDDANgA2ADVAO0A0gD8AM0A/w
DKAP8AyQD/AMkA/wDJAP8A/wAHAP8AAgD/AAAA/wAHAP8AEwD+AB8A+gAqAPcANQDxAEA
A7QBLAOkAVQDmAF0A5ABIAOEAAwDfAHEA3gB3ANwAfQDaAIIA2ACIANYAjgDTAJUA0QCdA
M8ApQDNAK8AywC8AMkAzWDFAOYAwQD2AL8A/wC9AP8AvQD/ALwA/wC8AP8A/wABAP8AA
AD/AAAA/wADAPoADgD0ABoA7wAlAOgAMADkADsA4QBGANwATwDZAFgA1QBgANIAZgDQA
GwAzgByAMwAeADKAH0AyACDAMcAiQDFAJAAwwCYAMEAoQC/AKsAvQC3ALkAxwC2AN0At
ADxALIA/wCxAP8AsAD/ALAA/wCwAP8A/wAAAP8AAAD/AAAA+AAAAPAACQDoABUA3wAgAN
kAKwDUADUA0QBAAM0ASgDJAFIAxQBAMMAYQDBAGcAvwBtAL4AcwC8AHgAuWB+ALkAh
QC4AlwAtgCUALQAnACyAKYArwCyAKwAwQCqANUAqADsAKcA+gCmAP8ApAD/AKMA/wCjAP
8A/wAAAP8AAAD/AAAA7gAAAOQABADaAA4A0AAaAMoAJQDGADAAwQA6AL4ARAC7AE0Au
ABVALYAXAC0AGIASwBoALEAbgCwAHMArWB5AK0AgACsAlcAqgCPAKgAmACmAKIApACuA
KIAvACfANAAngDmAJwA9wCaAP8AmQD/AJgA/wCYAP8A/wAAAP8AAADuAAAA4wAAANgAAA
DLAAgAxAAUAL4AHwC5ACoAtQA0ALIAPgCwAEcArQBPAKsAVwCqAF0AqABjAKYAaQCIAG8
ApAB1AKIAfACHAIMANwCMAJ4AIQCcAKAAmgCrAJcAuQCVAMwAkWdKAJEA9ACPAP8AjgD/AI
0A/wCNAP8A/wAAAPEAAADkAAAA1QMAAMcBAADBAEEAugANALQAGACwACMArAAuAKkA
OACmAEIApABKAKEAUgCgAFkAngFfAJ0CZQCbA2sAmgRyAJkEeQCYBYAAIgaJAJQGkwCSB
54AkAeqAl4luACMCMwAiQnjAlcK9gCFC/8Agwv/AIIL/wCCC/8A+QAAAOkGAADbEAAAYBMAAL
wRAAC0DQAAsAUEAK0AEQCoARwApAQoAKAIMgCdCjwAmwxFAJkNTQCXDIQAlg9bAJQQYg
CTEWgAkRFvAJASdgCOE34AjROHAIsUkQCJfJwAhxWpAIUVuACCFcwAfxblAH0W+AB7Fv8Ae
Rb/AHkV/wB5Ff8A8glAAOITAADOHQAAvh4AALldAACrGgAAphYAAKQOBwCgDhYAmxliAJgVL
QCVFzcAkhHBAJAZSQCOGIAAjBtXAlscXgCKHGGQAIb1rAlcdcgCFHnsAgx+EAlIfjgCAIJoAfSCmA
HsgtQB5IMkAdh/iAHMf9gFyH/8BcB7/AXAd/wFvHf8B7A4AANoeAADGJgAAticAAKsnAACjJQAAN
SEAAJscAACYGRAAKx0eAl8fKQCMITMAiil9AlgjRQCGJE0AhCVUAIImWwCBJmEAfydoAH4nbw
F8KHgBeyiBAXkoiwF3KJcBdCikAXIosgFwKMYBbSfgAmon9QJpJv8CZyX/A2cl/wNnJP8D5xgAA
NMnAAC/LQAARY8AAKQvAACcLQAALyoAAJMmAACQlwsAjCUZAlgoJQCfKi8Agis5AlAsQgF9LU
kBfC1QAXotVwJ4LI4Cdy5IAnUubAJ0LnQDci9+A3AviANuL5QDbC+hA2kvsARnLsMEZC7dBGlt8

wRgLP8EXyv/BF8r/wReKv8E4SAAAM0uAAC5MwAAqjUAAJ41AACWNAAAkTEAAIwuAACJLAc
AhS0VAIEwIQB9MSwAezl1AXgzPgF2M0YCdDNNA3I0VARwNFsEbZrIBW00aQVrNHEFaTR7B
mc0hQZINZEGYzSfBmE0rgZeNMAGXDPaB1oy8QZZMv8GWDH/Blcw/wZXMP8G3CYAAMg1AA
C0OAAApjoAAJo6AACSOQAAjDYAAIY0AACDMwMAfjQSAHo2HgB3OCkAdDkyAXE5OwJvOUI
DbTIKBGs5UAVpOVcGZzleB2U5ZgdjOW4IYTI4CF85ggldOY8JWzmcCVk5qwlXOL4JVDjYCVm3
8AISNv8IUTX/CFE1/whQNf8I2CwAAMM5AACwPAAAoj4AAJY+AACNPQAAhZsAAIE6AAB9OQA
AeToOAHU8GwBxPSYAbj4vAWs+OANpPj8EZj5HBmQ+TQdiPIQIYD5cCV4+YwpcPmsKWj11C1
g9gAtWPYwLVD2aC1I9qQtQPbwLTTzVC0w77gpLO/8KSzr/CUo5/wlKOf8J0jIAAL89AACsPwAAAn
kEAAJNCAACJQQAAGkAAAH0+AAB4PgAAcz8LAG9BGABsQiMAaUMtAWZDNQNjQzwFYENE
BI5DSwhcQIEJWkJZCIhCYAtVQmkMU0JyDVJCfQ1QQY0NTkGYDUtBqA1JQboNR0DTDEY/7Qx
FP/4LRT7/CkU9/wpFPf8KzjcAALtAAACpQwAAm0UAAI9FAACGRQAAfkQAAHhBAABYqgAAbU
QHAGpGFQBmRyEAY0gqAWBIMgNdSDoFW0hBBIHSAhWR08KVEdWC1FGXgxPRmYNTUZw
DktGew5JRocOR0aWDkZFpg5ERbgOQkTRDUBE7AxAQ/0MP0L/Cz9B/ws/Qf8LjysAALdDAACm
RgAAI0gAAIxIAACCSAAAEkgAAHNFAABtRwAAaEKEAGRKEgBhTB4AXk0nAvtNMAJYTTcEVU
0+BINMRQhQTEwKTktTC0xLWw1KS2QOR0ptD0ZKeA9ESoUPQkqUD0BKpA4+SrYOPEnPDTtl
6ww7R/wMOKb/DDpF/ws6Rf8Lx0AAALNHAACiSQAAlEsAAIHLaAB+TAAAdksAAG5JAABoTAA
Yk4AAF5PDgBbURsAWVlkAFZSLQFTUjQDUFI8BU5RQgdLUUkJSVFRC0ZQWAXEUGEOQk9r
DkBPdg8+T4MPPU+SDjtPow45T7UNN07NDDZN6gw1TPwMNUv/DDVK/ws1Sf8Lw0UAALBKAA
CeTQAAKU4AAIVPAAB7TwAAck8AAGhOAABiUQAAXFMAAFhVCgBVVhcAUlchAFBYKgFNWDI
CS1c5BEhXQAZGV0clQ1ZOckFWVgs/VI8MPVpDttVdA05VYENOFWRDTZVoQw0VbMLMITM
CjFT6AowUfsLMFD/Cy9P/wsvT/8LwEoAAktOAACbUQAAjVIAAIFSAAB3UwAAblMAAGJTAABbV
gAAVlgAAFJaBgBOXBMATF0dAEpdJgBHXi4BRV01A0NdPQRAXUQGPI1LCDxcUwk5XFwKN1x
mCzZccgs0XH8LMlyPCjFcoAovW7IJLVvKCCxZ5wgrV/oJKIb/CSpV/woqVP8Ku1AAAKdSAACXV
AAAIvUAAH1WAABzVwAAaVgAAF1aAABWXAAAT18AAEpgAQBGYg4ARGMZAEJkIgBAZCoA
PmQyATxkOQI6ZEAE0GRIBTZkUAY0Y1kHMmNkCDBjcAguY30ILWONBytjngYqY7EGKGLJBS
dg5gUIXvoGJV3/ByRb/wgkW/8ItlQAAKJXAACSWAAAhFkAAHlaAABuWwAAZVwAAFIaAABRYQ
AASmQAAENnAAA+aQgAO2oTADlrHQA4ayUANmstADVsnAEzbDwBMWxEAi9sTAMta1YEK2th
BCprbQQoa3sEJ2uKBCVrnAMka68DImrHAiBp5QlfZvkEH2T/BR5j/wUeY/8Gr1oAAJ1cAACNXQA
AgF4AAHRfAABqYAAAYWEAAfZkAABMZwAARWoAAD1tAAA2cAEAMXEMAC9yFgAtcx8ALHM
nActzLgApdDYAKHQ+ACd0RwEldFEBJHRcASJ0aQEhdHcBH3SHAR50mQEcc6wAGnPEABhx
4wAYb/gBF23/Ahds/wMXbP8DqV8AAJdhAACIYgAAe2MAAHBjAABmZAAAXGYAAFFqAABHbQ
AAP3AAADdzAAAvdgAAKHkEACN7DgAhexcAlHwfAB98JwAefC8AHXw3ABx8QAAafUsAGX1W
ABh9YwAWfXIAFX2CABN9IQARfKgADnu/AA163gANePYADnf/AA51/wEOdf8BomUAAJFmAAC
DZwAAAdmgAAGxpAABhagAAV/m0AAExwAABBdAAAOXcAADF6AAApfQAAIYAAABmCBAAUHA
0AEoUWABGFHgARhSYAEIUuAA6GNwANhkiAC4ZOAAqGWwAJhmoAB4Z7AAWFjQADhaEAA
YS2AAGD0gAAguwAAYD7AAF//wACf/8AmmwAAItsAAB9bQAAcm0AAGdvAABbcQAAUHUAEE
V4AAA7fAAAMn8AACmDAAAhQAAGogAABKKAAALjAYABI0OAAGOFgAAjh4AAI4mAACPLwA
AjzkaAI9FAACPUgAAj2EAAI9yAACPhAAjpkAAI6tAACNxgAAjOMAAIr1AACK/gAAIf8AIHIAAIvy
AAB5cwAAbXQAAGB2AABUegAASX0AAD6CAAA0hQAAKokAACKLAAAajgAAEpAAAAqSAAAD
IAAAAJUIAACWDwAAIxcAAJgeAACyJgAAmS8AAJk6AACaRwAAmIYAAJpnAACaegAAmY8AAJ
mkAACyUwAAI9kAAJbuAACW+QAAlf0AjXkAAIF5AAB0eQAAZnwAAFI/AABNhAAAQYgAADeMA
AAsjwAAIpMAABqVAAASmAAACZoAAAKcAAAANGAAAj8AAACgBwAAoQ0AAKIUAACjHAAApC

QAAKUuAACmOgAApkkAAKZbAACmbgAApYMAAKWaAACIsAAApMoAAKPkAACj8gAAovcAiIA
AAHyAAABtggaAX4YAAFKKAABFjwAAOZMAAC+XAAAKmwAAGp4AABKhAAAJowAAAaUAAA
CmAAAAqAAAAKoAAACrAAAArAMAAK0JAACuEQAAsBkAALEiAACzLAAAtDoAALRLAAC1XgA
AtXMAALWKAAC0owAAs7oAALLTAACy5wAAsu0AhlcAAHWJAABmjQAAWJIAAEuWAAA+mWAA
AMaAAACekAAAcPwAAE6oAAAmTAAArwAAALAAAACyAAAAAtAAALUAAAC2AAAAuAAAAALk
AAAC7AwAAvQsAAL8UAADCHgAAxSoAAMc6AADITQAAYWMAAMI5AADKkQAAYqgAAMq7AA
DJzgAAydgAfZEAG6UAABgmQAAUZ8AAESkAAA3qQAAKq0AACCCxAAAVtAAACrcAAAC5AAA
AugAAALsAAAC9AAAAvWAAAMEAAADDAAAAxQAAAMcAAADKAAAAzQAAANAFADVDgAA2
hoAAOAOAADhOgAA4IAAAONnAADkfwAA5ZUAAOWoAADmtwAA5b0Ad50AAGihAABapwAAS6
0AAD6yAAAutgAAILkAABW8AAAjvwAAAMEAAADEAAAAxgAAAMcAAADJAAAAyWAAAM0AAA
DQAAAA0wAAANYAAADaAAAA3gAAAOIAAADnAAAA7AoAAPEXAAD2JgAA+DkAAPIQAAD6a
AAA+n8AAPqTAAAD6oQAA+6YA/wANAP8ACAD/AAcA/wAMAP8AGAD/ACQA/wAvAP8AOgD/AE
UA/gBQAPsAWQD5AGIA9wBpAPUAbwD0AHUA8wB7APEAgADwAIYA7gCMAOWAkGdQAJkA6
QCgAOcAqQDmALQA5ADDAOIA2ADgAO4A2QD+ANUA/wDUAP8A1AD/ANMA/wDPAP8A/wAH
AP8AAQD/AAAA/wAIAP8AFAD/ACAA/wArAPwANQD5AEEA9gBLAPIAVQDwAF0A7QBkAOoAa
wDoAHEA5gB2AOQAFAdjAIEA4QCHAOAAjQDeAJQA3ACcANoApQDYAK8A1gC8ANIAZwDNA
OYAYQD5AMgA/wDHAP8AxgD/AMYA/wDFAP8A/wABAP8AAAD/AAAA/wAEP8AEAD8ABwA9A
AmAPEAMQDvADwA6wBGAOYAUADiAFgA3gBfANwAZgDaAGwA2ABxANYAdwDUAHwA0gCC
ANEAiADPAI8AzQCXAMsAoADJAKoAxQC2AMEAxgC+AN8AvADzALsA/wC5AP8AuAD/ALgA/w
C4AP8A/wAAAP8AAAD/AAAA/gAAAPcACwDuABcA6AAiAOMALADfADYA2wBBANUASgDSAF
MAzwBaAMwAYQDKAGcAyABsAMYAcgDFAHcAwwB9AMIAgwDAAIoAvQCSALsAmwC4AKUAt
QCwALMAvwCxANYArwDtAK4A/QCtAP8ArAD/AKsA/wCrAP8A/wAAAP8AAAD/AAAA9AAAAOoA
BwDfABEA2AAAdANIAJwDNADEAyga7AMYARQDCAE0AwABVAL0AWwC7AGEAuABnALYAbA
C0AHIAswB4ALEAfgCwAIUArgCNAK0AlgCrAKAAqQCrAKcAugCIAM0ApADmAKIA+AChAP8Ao
AD/AJ8A/wCfAP8A/wAAAP8AAADzAAAA6AAAAANoAAQDQAAwAyAAXAMIAIgC+ACwAuW1AL
gAPwC0AEgAsQBPAK4AVgCsAFwAqwBiAKoAaACoAG0ApwBzAKYAeQCIAIAowCIAKIAkgCg
AJwAngCnAJwAtQCaAMcAmADfAJYA8gCVAP8AkWd/AJIA/wCRAP8A/wAAAPUAAADoAAAA2Q
AAAMwAAADCAAcAuWArALYAHACyACYArgAwAKsAQQC0AEIAPgBKAKQAUQCIAfC0QBdA
J8AYwCeAGkAnQBvAJwAdQCaAHwAmQCFAJcAjgCVAJgAIACkAJEAsQCPAMIAjQDZAI7gC
JAP0AhWd/AIYA/wCGAP8A+wAAAOwAAADcAAAAyQAAAL4AAAC3AAAAAsQALAKwAFgCoACA
ApAAqAKEANACeAD0AnABFAJoATACZAFMAIwBZAJYAXwCUAGUAKwBrAJIAcgCQAHkAjwCB
AI0AiWCLAJYAIQCIAIcArwCFAMAAggDWAIAB7AB+AvoAfQL/AHwC/wB7Af8A8wAAAOmFAADN
DAAAvg0AALMLAACsCAAAqAEDAKQADwCfABoAnAAkAJkALgCWATcAkwNAAJEESACQBU8
AjgZVAI0HWwCLCGEAighoAlkJbwCHCnYAhgp/AIQLiQCCC5QAgAyhAH0MrwB7DMAAEzYAH
YN7wB0Df0Acw3/AHIN/wBxDP8A7QMAANsSAADEFwAAAtBkAAKoYAAACiFgAAAnREAAJsKBgCY
BhMAIAkeAJEMKQCODjIAixA7AIkRQwCHEkoAhhNRAIQUWACDFF4AgRVIAIAVbAB+FnQAFBd
9AHoXhwB4F5IAdhifAHQYrQByF78AbxfYAG0X7wBrF/8AARf/AGkW/wBoFv8A5g4AANAcAAC7I
AAARsIAAKliAACaHwAAIRwAAJIXAACREgwAJBQZAlgXJACFGS4Agho3AIAbPwB+HEcAfR1OA
HseVAB6HIsAeB9iAHcfaQB1IHEAdCB6AHlghABwIZAAbiGdAGshqwBplbwAZiDUAGQg7QBiH/4
BYR//AWAe/wFgHf8B4BgAAMgjAAC1JwAApikAAJwpAACUJwAAjiQAAIohAACIHQcAhB4VAIAgl
AB9lioAeiMzAHgkPAB2JUMAdSZKAHMmUQByJ1gAcCdfAG8nZgFtKG4Bayh3AWkoggFnKI0BZ
SibAWMoqQFgKLoBXifSAVsn6wJaJv0CWSX/Algk/wJYJP8C2iAAAMlpAACvLQAAoS8AAJYvAA

COLQAAiCsAAIMpAACBJglAfSYRAHkoHQB2KScAcyswAHESoABvLEABbS1HAWstTgFqLVUC
aC5cAmYuYwJILmsDYy50A2EufwNfL4sDXS+YA1supwNYLrgDVi3PA1Qt6gRSLPwEUSv/BFEq/
wRRKf8E1CYAAL0uAACrMgAAAnDQAAJI0AACJMwAAgzEAAH4vAAB6LQAAadi0MAHmVgQBvM
CQAbTEtAGoyNQFoMz0BZjNEAmQzSwNiM1IDYTNZBF80YARdNGgFWzRyBVk0fAVXNIgFVTS
WBVM0pQVRM7UFTjLNBUwy6AVLMfoFSjD/BUov/wVKL/8FzywAALgyAACnNgAAmTgAAI44AA
CFNwAAfjUAAHk0AAB1MgAAcDMJAG01FgBpNiEAZzcqAGQ4MgFiODoCYDhBA144SARcOE8
EWjhWBVg4XQZWOGYGVDhvB1I4eQdQOIYHTjiTB0w4owdKOLQHSDfKB0Y25gdFNfkHRDT/B
kQ0/wZEM/8GyzIAALQ2AACjOgAAITsAAIo7AACBOwAAejoAAHQ4AABvNwAAazkGAGc6EwBk
Ox4AYT0nAF49MAFcPTcCWj0+A1g9RQRWPUwFVD1TBII9WwdQPWMITj1sCEw9dwIKPYMJS
D2RCUY9oQIEPLIIQjZICEA75Qg/OvkHPzn/Bz84/wc/N/8HxzYAALE6AACfPQAAKj4AAIY/AAB9P
wAAAdT4AAHA8AABqPAAAZT4CAGE/EABeQBsAXEEIAFICLQFWQjQCVEI7A1JCQgVQQkkGTk
JQB0xBWAhKQWAJSEFqCUZBdQpEQYEKQkGPCkBBnwk+QbAJPEDHCDs/4wg6PvgIOj3/CDk
8/wg5O/8lwzkAAK09AACcQAAAj0EAAINCAAB5QgAAcUEAAGs/AABIQAAAYEIAAFxEDABZRR
gAVkYiAFRHKGfRRzICT0c5A01HQA VKR0cGSEZOB0ZGVQhERI4JQkZnCbGcgO/Rn8KPUaN
CjtFngk5Ra8JN0XFCDZE4gg1QvclNEH/CDRA/wg0QP8lvj0AAKpAAACZQwAAi0QAAIBFAAB2R
QAAbUUAAGZCAABfRQAAWkcAAfZJCQBTShUAUESfAE5MJwBMTc8BSUw2A0dMPQRFTEQ
FQ0tLB0FLUwg/S1sJPUtICTtLcAo5S30KOEuMCTZLnAk0Sq0IMkrEBzBJ4QcvR/YHL0b/CC9F/w
gvRP8lukAAAKZEAACVRgAAiEcAAHxIAABYsQAAaUkAAGBIAABZSgAAVEwAAFBObQBMTxE
ASIAbAEhRJABGUiwBRFizAkJSOGNAUUEEPIFIBTtRUAc5UVkIN1BjCDZQbgg0UHsIMICKCDF
QmgcvUKwHLU/CBitO3wYqTfUGKkv/BylK/wcpSf8HtUQAAKJIAACSSQAAhEsAAHhMAABuTAA
AZU0AAFxNAABTUAAATIIAAEITAABGVQ0AQ1YXAEFXIABAVygAPIgwATxYNwI6Vz4DOFdFB
DZXTQU0V1YGMldgBjBxBAYuV3kGLVeIbitXmQUpVqsFKFbBBCZV3gQIU/UFJFH/BiRQ/wYkT/8
GsEkAAJ1MAACOTQAAgE8AAHRPAABqUAAAYVEAAFdSAABPVQAASFcAAEJZAAA/WwgAO
1wTADpdHAA4XSQANI4sADVeMwEzXjoBMV5CAi9eSgMtXIMDLF5eBCpeaQqOXncEJ16GAyVe
lwMkXakCII2/AiBc3AlfWvQDH1j/BB5W/wQeVv8Fq00AAJhQAACJUQAAe1MAAHBTAAABmVAAA
XVUAAFNXAABKWgAARFwAAD1fAAA2YglAMmMNADBkFwAvZB8ALWUmACxILgArZjUAKWY
9AChmRgEmZk8BJWZaASNmZgEiZnMBIGaDAR5IIAEdZacAG2S8ABlj2gAZYfMBGF//Ahhe/wIX
Xf8DpVIAAJNUAAACEVgAAAd1cAAGtYAABiWQAAWVoAAE9cAABGXwAAP2IAADhIAAAwaAAAK
WsGACZsEAAkbBgAl20gACJtKAAGbS8AH243AB5uQAAdbkoAHG5VABpuYQAZbm8AF25/ABV
ukQATbaQAEWY5AA5r1gAQafEAEGj/ABBM/wEQZf8BnlGAAI5ZAAB/WwAAclsAAGdcAABeXQA
AVF8AAEpiAABAZgAAOWgAADJrAAAqbgAAI3EAABt0BwAYdREAF3UZABZ2IQAVdigAFHYwA
BN2OQASdkMAEHZOAA53WwANdmkAC3Z5AAp2iWAlDZ4ABXSyAARzzAAEcugABHH5AAVv/w
AFb/8AmF4AAIhfAAB5YAAAbmAAAGRhAABZYwAAT2UAAEVpAAA7bAAAM3AAACtzAAAJdgAA
HHgAABV7AAANfQgACX4RAAh+GQAHfyEABn8oAAR/MQADfzsAAAn9GAAB/UwAAf2EAAH9xA
AB/hAAAfpcAAH2rAAB8wwAAe+EAHr0AAB5/QAAef8AkWQAAIFIAAB1ZQAAamYAAF5nAABT
agAASG0AAD9xAAA1dAAALHgAACR7AAAcfgAAFYAAAA2CAAAGhAQAAIYLAACGEwAAhxoA
AlghAACIKQAAiTIAAlk9AACJSgAAiVgAAlpAACJewAAiI8AAlkAAChuwAAhdgAAIXuAAce+gAA
g/8AimoAAHxrAABxawAAZWwAAfhuAABNcgAAQnYAADh6AAAufgAAJIEAABYEAUUhwAADIk
AAAWLAAAajQAAA14FAACPDAAAkBMAAJEZAACSIQAakykAAJQzAACUPwAAIE4AAJReAAC
UcQAAIYYAAJObAACSSQAakcwAAJDnAACQ9AAAJ/wAhHEAAHhxAABscgAAxNQAFF4AABFf
AAAOoEAADCEAAAmiAAAHlwAABSPAAALkQAABJMMAACVAAAAIgAAAjgAAACZBAAAmgoA
AJsRAACcFwAAanh8AAJ8oAACgMwAAoEIAAKBTAAcGzGAAoHoAAKCRAACfpwAAAn8AAAJ7cA

ACd7QAAncfAf3gAAHN4AABlegAAV34AAEqDAAA+iAAAMowAACiQAAAelAAAFJcAAAuaAAAD
nAAAAJ0AAACfAAAAoQAAAKMAAACkAAAApgAAAKcFAACpDAAAqhQAAKwcAACTjGAArjMAA
K9EAAcuVwAArmwAAK6DAACumwAArbEAAK3KAACT4AAArO0Ae38AAGyBAABehQAAUloAA
EOPAAA2IAAAKpkAACCDAAAWoAAADKMAAAKmAaaaqAAAAKkAAACrAAAArQAAAK4AAAC
wAAAAsQAAALMAAAC1AAAAtwcAALkQAAC8GQAAvyUAAMAzAADBRgAAwlsAAMNxAADCiW
AAwaMAAMC5AADaywAAwNwAdIkAAGWMAABXkQAASZcAADycAAAuogAAI6YAAbiqAAAOrg
AABLAAAACzAAAAAtQAAALYAAAC4AAAAuQAAALsAAAC9AAAAvwAAAMEAAADDAAAAxgAAA
MkBAADNCgAA0hYAAncjAADaNAAA20kAAN1gAADedwAA348AAN+jAADgsgAA378AbpQAAF+
ZAABRnwAAQqUAADWrAAAosAAAHLUABK4AAAFuwAAAL0AAAC/AAAAwQAAAMIAADEA
AAAxgAAAMgAADKAAAAzQAAANAAAAADUAAAA2AAAAAN0AADhAAAA5wUAAOwSAADylQA
A8zQAAPRMAAD1ZAAA9nsAAPePAAD3ngAA96gA/wAGAP8AAwD/AAIA/wAJAP8AFQD/ACEA/
wArAP8ANgD/AEEA/gBLAPoAVAD3AFwA9QBkAPMAagDyAHAA8AB1AO8AewDtAIAA7ACGAO
sAjADpAJMA5wCbAOUApADjAK8A4QC8ANsA0ADVAOgA0wD7ANEA/wDRAP8A0AD/AMwA/w
DIAP8A/wAAAP8AAAD/AAAA/wAFAP8AEgD/ABwA/gAnAPwAMgD4ADwA8wBGAO8AUADtAFg
A6gBfAOgAZQDmAGsA5ABxAOIAdgDhAHsA3wCBANwAhwDaAI4A1gCWANMAAnwDQAKkAzQ
C2AMkAxwDHAOAAxQD1AMQA/wDDAP8AwgD/AMIA/wC/AP8A/wAAAP8AAAD/AAAA/wABAP4
ADQD2ABgA8gAjAO4ALQDrADcA5wBBAOIASwDfAFMA3ABaANcAYADTAGYA0ABsAM0AcQD
LAHYAyQB8AMcAggDFAIkAxACRAMIAmgDAAKQAvgCwALsAwAC6ANYAuADvALYA/wC1AP8
AtAD/ALQA/wC0AP8A/wAAAP8AAAD/AAAA+wAAAO8ABwDpABMA4gAeAN0AKADaADIA1gA8
ANAARQDJAE4AxQBVAMIAWwDAAGEAvwBmAL0AbAC8AHEAugB3ALkAfQC4AIQAtgCMALQ
AIQCzAJ8AsQCrAK8AuQCtAM0AqwDoAKkA+wCoAP8ApwD/AKcA/wCmAP8A/wAAAP8AAAD5A
AAA7gAAAOEAAwDYAA4A0AAZAMsAlwDHAC0AwQA2AL0AQAC6AEgAtwBPALUAVgCzAFwA
sQBhALAAZwCvAGwArQByAKwAeACrAH8AqQCHAKgAkACmAJApACmAKMAswChAMYAnw
DgAJ0A9ACcAP8AmgD/AJoA/wCaAP8A/wAAAPoAAADtAAAA3QAAANEAAADGAakAwAAUALo
AHgC1ACgAsgAxALAAOgCtAEIAqwBKAKkAUQCnAFcApgBcAKQAYgCjAGcAogBtAKAAcwCfA
HoAngCCAjwAiwCaAJUAmACHAJcArgCUAL8AkgDXAJEA7gCPAP0AjQD/AI0A/wCMAp8A/QAA
APAAAADeAAAAzGAAAMIAAAC5AAUAswAOAK0AGQCpACMApwAsAKQANQCiAD0AoABFAJ4
ATACcAFIAmgBXAJkAXQCYAGIAIgBoAJUAbwCUAHYAkGB+AJEahwCPAJEAjQCdAIsAqgCJA
LoAhgDPAIQA6ACDAPgAgQD/AIAA/wB/AP8A9QAAAOUAAADPAAAAvwAAALUAAACuAAAAqA
AJAKMAEwCgAB0AnQAmAJALwCYADgAIQBAAJMARwCSAE0AkABTAI8AWQCNAF4AjABkAI
sAawCJAHIAiAB6AIYAgwCEAI4AggCaAIAAqAB+ALcAewDLAHOA4wB3APYAdgD/AHUA/wB0AP
8A7gAAANgCAADEBgAAtAcAAKOHAAACjAwAAAnwADAJsADQCXABcAIAAhAJEAKgCOADMAjAA
7AlOAggCIAEkAhwBPAlUAV/QCEAFsAggBhAIEBaAB/Am8AfgJ3AHwDgQB6A4wAeASYAHYEpg
B0BLUAcQTJAG8E4wBtBfQAbAX/AGsE/wBqBP8A5gMAAM0NAAC5EgAAqxMAAKETAACaEAA
AIQwAAJMFBgCQABEAjAEbAIkDJQCGBS4Agwc2AIEIPgB/CkUAfgpLAHwLUQB7DFgAegxeAH
gNZQB3DW0AdQ52AHMOgABxD4sAbxCYAG0QpgBqELYAaA7MAGUP5gBjEPgAYg//AGEO/wB
hDv8A3w8AAMUWAACyGgAAox0AAJkcAACSGgAAjRcAAIkTAACIDAkAhQwVAIEPIAB+ESkAex
MxAHkUOQB3FUEAdRVHAHQWTgByFIQAcrDbAG8XYgBuGGGoAbBhzAGoZfQBoGYkAZhmWA
GQZpABhGbQAXxnKAFwY5ABbGPgAWHj/AFkX/wBZF/8A1RgAAL0eAACrIQAAniMAAJMjAACLI
QAAhR8AAIEdAAB/GAIAfRYQAHkZHAB2GiUAcxwuAHEdNgBvHT0AbR5EAGwfSwBqH1EAaSB
YAGcgYABmIWgAZCFxAGIhewBgloYAXiKTAFwiogBZlBlAVyHHAFUg4wBTIPcBUh//AVEE/wFR
Hv8Bzh4AALckAACmJwAAmCkAAI4pAACFKAAafYAAHskAAB4IQAAdR8MAHlgGABulilAbCM

qAGkkMwBoJTToAZiZBAGQmSABjJ08AYSdWAF8nXQFeKGUBXChuAVoeeAFYKIQBViiRAVQoo
AFSKLABTyfFAU0n4QFMJvYCSyX/Akok/wJKl/8CySMAALIoAACHLAAAIc0AAIkuAACALQAAeis
AAHUpAABxJwAAbiYIAGsnFABoKR8AZSonAGMrMABhLDcAXyw+AF0tRQFcLUwBW1TAVgtW
gJXLmICVS5rAlMudgJRL0EDTy6PA00ungNKLq4DSC3DA0Ys3wNFK/UDRCr/A0Qp/wNDKf8DxC
cAAK4sAACdMAAAKDEAAIUyAAB8MgAAdTAAAHAAuAABsLQAAaC0EAGUtEQBhLxwAXzAIAF0
xLQBaMjQAWTI7AVcyQgFVM0kCUzNQA1IzVwNQM2ADTjNpBEwzcwRKM38ESDONBEYznAR
EM6wEQjLBBEAX3QQ/MPMEPi//BD4u/wQ+Lf8EwCsAAKowAACaMwAAjDUAaIE2AAB4NQAAC
DUAAGsyAABmMQAAYjIAAF8zDQBcNRkAWTYiAFc3KgBVNzEBUzg4AVE4PwJPOEYDTTthNA
0s4VQRKOF0ESDhmBUY4cQVEOH0FQjILBUA4mgU+N6sFPDe/BT022wQ5NfIEOTP/BDgy/wQ4
Mv8Euy8AAKc0AACXNwAAiTgAAH05AAB0OQAABDgAAGc2AABhNgAAXTcAAfK5CgBWOHya
UzsfAFE8JwBPPC8BTTw2AU9PQJJPUmDSDxKBEY8UgVEPFsFQjxBkA8bgY+PHsGPTyJBjs
8mAU5PKkFNzu+BTQ62gQ0OfEFMzj/BTM3/wUzNv8FtzMAAKM3AACTOgAAHjsAAHo8AABwPA
AAaDwAAGI7AABcOgAAVzwAAFQ9BwBQPxMATkAcAExBJQBKQSwASEEzAUZBOgJEQUED
QkFIBEBBUAU+QVgFPUFiBjtBbAY5QXkGN0GHBjVBlwUzQKgFMUC8BS8/2AQuPvEFLjz/BS47/
wUtOv8FszyAAKA7AACQPAAAgj4AAHY/AABtPwAAZT8AAF4/AABXPwAAUkEAAE5CAwBKRA
8ASEUZAEEZFIgBERikAQkYwAUFGNwl/Rj4CPUZFAztGTQQ5RIYFN0ZfBTVGagYzRncGMkaFB
TBGIQUuRacELEW7BCpE1gMpQ/AEKUH/BChA/wUoP/8FrzoAAJw+AACNQAAAf0EAAHNCAAB
pQgAAYUMAFAIDAABRRAATEYAAEHAAABFSQsAQkoWAEBLHgA+SyYAPUwtADtMNAE5TD
sCN0xCAjVMSgMzTFMEMUtdBDBLaAUuS3UELEuDBCtLIAQpS6UDJ0q5AyVK1AlkSO8DI0b/BC
NF/wQjRP8Eqz4AAJhBAACJQwAAe0QAAG9FAABIRgAAXUYAAFVHAABLSQAARksAAEJNAA
A+TgcAO1ASADIQGwA3USIANIEqADRSMQAzUjgBMVl/AS9SRwltUIACK1JaAypSZgMoUnIDJ1
KBAYVRkgIjUaQCIVG3AR9Q0wEeTu4CHkz+Ax5L/wMdSv8EpklAAJRFAACERwAAAd0gAAGtJAA
BhSgAAWUoAAFFLAABITgAAQIAAADxSAAA3VAIAM1YMADFXFgAvVx4ALlglACxYLAARWDQA
KVg7ACHyRAEmWE0BJVhXASNYyWEiWHABIFh/AR5YjwEdWKIAG1e1ABIW0QAYVO0AGFP9
ARhR/wIXUP8CoUcAAI9JAACASwAAckwAAGdNAABdTQAAVU4AAE1QAABEUwAAPIUAADdX
AAAxWgAAK1wHACdeEQAmXhkAJF8gACNfJwAiXy8AIV82ACBgPwAeYEGAHWBtABtgXwAaY
GwAGGB7ABZfjAAVX58AE16zABBdzgAQW+sAEVr8ABFY/wERV/8Bm0sAAIpNAAB7TwAAblAA
AGNRAABaUgAAUIMAAEIVAABAWAAA0VoAADJdAAArYAAAJGMAAB5ICQAbZhIAGmYaABInI
QAYZykAF2cwABVnOQAUZ0IAE2hNABJoWgAQaGcADmd3AAxniAAKZpsACGauAAZkxgAGY+
MABmL3AAAdg/wAHX/8AIVEAAIRSAAB2VAAAaVUAAF9VAABWVgAATVgAAERaAAA7XQAAM2
EAACxkAAAIzWAAHmkAABdsAQARbgoADW8TAAxvGgALbylACnAqAAIwMgAlcDwAB3BGAAV
wUwAEcGAAAm9wAABvgQAAb5QAAG6oAABtwAAbNwAAGrxAABp/AAAaf8AjIYAAH5YAABw
WQAAZVKAaFxaAABSWwAASF4AAD9hAAA2ZAAALWgAACVrAAAebgAAF3EAABFzAAAJdQY
AA3cNAAB3FQAAeBwAAHgjaAB4KwAAeDQAAHk/AAB5SwAAeVKAHHoAAB4egAAel4AAHeiA
AB2uAAAAddQAAHTsAABz+gAAcv8Ah1wAAHhdAABsXgAAYV4AAfDfAABMYgAAQmUAADhpA
AAvbQAAJnAAAB50AAAXdgAAEHkAAAI7AAABfQIAAH4JAAB/EAAAgyBAAIEdAACCCJAAAgIwA
AIM2AACDQgAAg1AAAINgAACCCgAAgoYAAIKcAACBsQAAGMwAAH/nAAB+9QAAff4AgGMAA
HNjAABoYwAAXWQAAFFnAABFawAAO24AADFyAAAAndgAAH3oAABd9AAAPgAAAB4EAAACE
AAAAhgAAAIcDAACICQAAiQ4AAIsVAACMHAAAjSMAAI4sAACOOAAAJkYAAI5WAACOaAAAJn
wAAI6TAACNqQAAjMIAAlvgAACK8QAAIfsAemkAAG9pAABjagAAVm0AAEpwAAA+dQAANHKA
ACI9AAAFgQAAF4QAAA6IAAAGigAAAIwAAACOOAAAKAAAAJEAAACSAQAAIAYAAJUMAACXE
wAAmBoAAJkjAACbLQAAMzoAAJtKAACbXAAAm3EAAJulaACaoAAAmbcAAJnSAACY6QAAI/U

AdnAAAGtwAABccwAAT3cAAEN7AAA3gQAAK4UAACGJAAAXjQAADpAAAAWTAAAAIQAAAJc
AAACZAAAAmgAAAJwAAACdAAAAAnwAAAKECAACiCQAAPBAAAKYYAACnIQAAqS0AAKk9AA
CpUAAaqWUAAKI7AACpkwAAqasAAKjCAACo2wAAp+wAcncAAGR5AABVfgAASIMAADuIAAAu
jQAAI5IAABmWAAAAOmgAABZ0AAACfAAAAoQAAAKMAAACIAAAApwAAAKgAAACqAAAArAAA
AK4AAACvAAAAsQMAALQKAAC2FAAAuR8AALotAAC7PwAAu1UAALpsAAC6hQAAu50AALuz
AAC6xgAAAutoAa4AAAF2FAABOigAAQZAAADSVAAAnmwAAHJ8AABGjAAAGpwAAAKoAAACs
AAAArgAAAK8AAACxAAAAswAAALUAAAC3AAAAuQAAALsAAAC9AAAAwAAAAMIAAADGBgA
AyxEAANAeAADSLgAA1EIAANVZAADXcAAA1ooAANWhAADUswAA1MQAZYwAAFaRAABIlwA
AO50AAC2jAAAqgQAAFq0AAAqxAAAAAtQAAALgAAAC6AAAAvAAAAAL0AAAC/AAAAwAAAAMIA
AADFAAAAxwAAAMoAAADOAAAAA0QAAANUAAADbAAAA4EAAOYOAADsHQAA7i8AAO9FA
ADwXQAA8nUAAPKLAADzmwAA86gA/wABAP8AAAD/AAAA/wAGAP8AEQD/ABwA/wAnAP8AM
QD/AD0A/QBHAPoAUAD4AFgA9ABfAPAAZQDsAGsA6QBwAOcAdQDIAHsA5ACAAOIAhwDgAI
4A3wCWAN0AnwDaAKoA2AC3ANUAyQDRAOMA0AD4AM4A/wDNAP8AywD/AMUA/wDCAP8A/
wAAAP8AAAD/AAAA/wACAP8ADQD/ABkA/AAjAPkALQD3ADgA8wBCAOsASwDIAFMA4QBaN
8AYADdAGYA2wBrANkAcQDXAHYA1QB8ANMAggDRAIkAzwCRAM0AmgDLAKUAYQCxAMYA
wgDDANsAwQDyAMAA/wC+AP8AvgD/ALsA/wC4AP8A/wAAAP8AAAD/AAAA/wAAAPoACQDzA
BQA7gAfAOsAKQDmADMA3QA9ANkARgDUAE4A0QBVAM8AWwDMAGEAygBmAMkAawDHA
HEAxQB2AMQAFQDCAIMAwACMAL4AIQC8AJ8AugCrALgAuWC1ANEAtADqALIA/gCwAP8AsAD
/AK8A/wCuAP8A/wAAAP8AAAD/AAAA9AAAAOwABQDjABAA3gAaANUAJADPAC4AzAA3AMgA
QADEAEkAwQBQAL8AVgC9AFwAuWbHALkAZgC4AGsAtgBxALUAdwCzAH4AsgCGALAAjwCuA
JoArACmAKoAtACoAMcApwDiAKUA+ACjAP8AowD/AKIA/wCiAP8A/wAAAP8AAAD1AAAA5gAA
ANkAAADQAAsAxwAWAMEAIAC+ACkAuWYALgAOwC1AEMASgBKALAAUQCuAFYArQBcAKs
AYQCqAGYAqQBsaKcAcgCmAHkApACAAKMAigChAJQAnwCgAJ0ArQCbAL8AmgDYAJgA8QC
WAP8AIQD/AJQA/wCTAP8A/wAAAPUAAADkAAAA1QAAAMcAAAC+AAcAtgARALIAGwCvACQ
ArQAtAKsANQCoAD0ApQBFAKMASwCiAFEAOABXAJ8AXACdAGEAnABnAJ0AbQCZAHQAIwB
7AJUAhACUAI8AkgCbAJAAqACOALgAJADPALoA6QCJAPsAhwD/AlcA/wCGAP8A+AAAAOgAAA
DTAAAAxAAAALkAAACvAAIAqgAMAKYAFgCjAB8AoAAoAJ4AMACcADgAmgA/AJcARgCVAEw
AkwBSAJIAVwCRAF0AjwBiAI4AaACMAG8AiWb3AlkAgACIAIsAhgCWAIIQApACCALMAgADIAH4
A4gB9APYAewD/AHsA/wB6AP8A8AAAAANkAAADFAAAAtgAAAKsAAACIAAAAAoAAHAJsAEQCY
ABoAIQAJAJMAKwCQADMAjgA6AlwAQQCKAEcAiQBNAIcAUwCGAFgAhABeAIMAZACCAGsAg
ABzAH8AfAB9AlcAewCTAHkAoAB3AK8AdQDCAHMA3ABxAPEAcAD+AG4A/wBuAP8A5wAAA
M0AAAC5AAAAqwMAAKECAACaAAAAIgABAJIACwCPABUAjAAeAlkAJgCGAC4AhAA2AIAPQ
CAAEMAfwBJAH0ATwB8AFQAewBaAHkAYQB4AGgAdgBwAHUAeQBzAIQAcQCQAG8AngBtA
K0AagC/AGgA2ABmAO8AZQD8AGQA/wBjAP8A2wIAAMIIAACwDAAAog4AAJkNAACRCwAAjAc
AAIoBBQCHAA4AgwAYAIEAIQB+ACkAewAxAHkBOAB3Aj8AdgJFAHQDSwBzBFEEAcgRXAHAF
XgBvBWYAbQZuAGsGeABpB4MAZwePAGUHnQBjB6wAYQe+AF4H2ABcB+0AWwf8AFoH/wBa
B/8A0QsAALoRAACoFgAAmxcAAJEWAACJFQAAgxMAAIAOAAAB/CQgAfAUSAHkHHAB2CSUA
cwotAHELNABwDDsAbg1CAGwOSABrDk4Aag9VAGgQXABmEGQAZRFsAGMRdgBhEolAXxKP
AFwSnQBaeQwAVxLAAFUR2gBTEfEAUHH/AFIQ/wBREP8AyRIAALMYAACiHAAAIr0AAIoAAC
CHQAAfBsAAHgYAAB2FAAAAdRALAHIRFwBuEyAAbBQpAGkVMABnFjcAZhc+AGQXRQBjGEsA
YRISAGAZWQBeGWEAXBpqAFoadABZGn8AVhuMAFQbmwBSGqsATxq+AE0Z2ABMGfAASxn/
AEoY/wBKF/8AwxgAAK0eAACdIQAAkCMAAIUkAAB8lwAAdiEAAHIfAABvHAAAAbRkHAGoZEwB

nGx0AZBwIAGldLQBgHjQAXh87AF0fQgBbIEgAWiBPAFghVgBXIV4AVSFnAFMicQBRIn0ATyKK
AE0imQBLlqkASCG8AEYg1gBFIO8ARB//AUMe/wFDHv8BvR4AAKgjAACZJgAAiycAAIAoAAB3K
AAAcCcAAGwkAABolwAAZiADAGMgEABglhoAXSMiAFskKgBZJTEAvyY4AFYmPwBUJ0YAUyd
MAFEnVABQJ1wBTihIAUwobwFKKHsBSCiIAUYolwFEKKcBQie6AT8m1AE+Je4BPSX+AT0k/wE
9I/8BuCIAAKQnAACVKgAAhysAAHwsAABByLAAAaysAAGYpAABjKAAAYCYAAAFwnCwBZKBYAV
ykfAFUqJwBTKy4AUSw1AFAsPABOLEMBTC1KAUstUQFJLVkBRy1jAkYtbQJELXgCQi2GAkAtI
Ql+LaYCPCy4Ajr0gE4K+wCNyr9Ajcp/wl3KP8CtCYAAKErAACRLQAAgy8AAHgwAABuMAAAZy
8AAGluAABeLAAAWiwAAFYtCABULhMAUS8dAE8wJABNMSwASzEzAEoyOQBIMkABRzJHAU
UyTwJDMlcCQTJgAkAyawl+MnYDPDKEAjoykwI4MqQCNjG3AjMw0AlyL+sCMi79AjEt/wlxLP8Ds
CkAAJ0uAACOMAAAgDIAAHQzAABrMwAAYzMAAF0yAABZMAAAVTEAAFEyBQBOMxAASzQa
AEk1IgBINikARjYwAEQ3NwFDNz0BQTdFAj83TAI+N1UCPDdeAzo3aAM4N3QDNjeCAzQ3kgMy
N6MCMDa1Ai41zwltnOoCLDP8Aiwy/wMsMf8DrS0AAJoxAACKMwAAfDUAAHE2AABnNgAAXzc
AAFk2AABUNAAATzYAAEs3AQBIOA0ARjkXAEQ6HwBCOyYAQTstAD88NAA9PDsBPDxCajo8
Sgl4PFICNjxcAzQ8ZgMzPHIDMTyAAY88kAMtO6ECKzu0Aik6zQloOekCJzj7Aic2/wMnNf8DqTAA
AJc0AACHNwAAeTgAAG05AABkOgAAXDoAAFU6AABPOQAASjsAAEY8AABDPQkAQD4UAD4/
HAA8QCMAO0AqADIBMQA4QTgBNkE/ATRBRwlyQVACMUFZAi9BZAMtQXADK0F+AilBjgloQK
ACJkCyASQ/ywEiPugBljz7Ail7/wliOv8DpTQAAJM3AACDOgAAdTsaAGo8AABgPQAAWD0AAFI
9AABKPgAAREAAAEBBAAA9QgYAOkQQADhEGQA2RSAANUUnADNGLgAyRjUAMEY8AS5G
RAEtRk0BK0ZXAILGYglnRm4CJkZ8ASRGjAEiRp4BIEWxAR5EygAdQ+cBHUH6AR1A/wldP/8Co
TgAAI87AAB/PQAAcJ4AAGY/AABdQAAAVUAAAE5BAABGQgAAP0UAADpGAAA3SAEAM0kLA
DBKFQAvSx0ALUskACxLKgAqTDEAKUw5ACHMQQAmTEoBJUxUASNMXwEhTGwBIEx6AR5Mi
gAcS5wAGkuvABhKyAAXSeYAF0f5ARdG/wEXRf8BnDwAAIs+AAB7QAAAbkIAAGJDAABZQwA
AUUQAAEpFAABCRwAAPEkAADZLAAAwTQAAK08HACHREAAAnURgAJVEfACRSJgAjUi0AIVi0
ACBSPQAFU0YAHVNQABxTXAAaU2gAGVJ3ABdSiAAVUpoAE1GtABFQxgAQT+QAE34ABFM/
wARS/8BI0AAAIZCAAB3RAAAaUUAAF5GAABVRwAATkgAAEdJAAA/SwAAN04AADFQAAArUg
AAJVUBACBXCgAdWBMAHFgaABtZIQAAwSgAGFkwABdZOAAWWUEAFVpMABNaVwARWW
QAEFIzAA1ZhAAMWZYAClipAAAdXwAAHVd0AB1TzAAhT/wAIUv8AkUUAAIBHAABySAAAZUoA
AFpKAABSSwAAS0wAAENNAAA7UAAAM1MAACxWAAAmWAAAH1sAABLeAwATYAwAEWAU
ABbhGwAPYSMADWEqAAxhMgALYTsACmFGAAhhUgAHYV8ABWFuAANgfgABYJEAAF+kAA
BeugAAXdUAAFvuAABb+gAAWv8Ai0oAAHtMAABsTQAAYE4AAFD0AABPTwAARIAAAD5TAA
2VgAALVKAACZcAAAgXwAAGWIAABNkAAALZwcABmgOANpFgACaR0AAWkkAABpLAAAaT
UAAGk/AABpSwAAaVgAAGlnAABpeAAAalsAAGifAABntAAAZc8AAGTPAABj+AAAYv8AhE8AAH
VRAABnUgAAXVIAAFRTAABLVAQAQVYAADhZAAAwXQAAJ2AAACBjAAAZZgAAE2kAAAttrAA
AEbgQAAG8KAABwEQAAcRgAAHEeAABByJQAACi4AAHI4AABByQwAAcIAAAHJfAABBycAAACoQA
AHGZAABwrgAAB8gAAG7IAABt9AAAbP0AfVUAAG9WAABjVwAAWlcAAFBYAABFWwAAO14AA
DJhAAApZQAAlWkAABIsAAASbwAAC3EAAAR0AAAAAdgAAAHcGAAB4DAAAeRIAAHoYAAAB7H
wAAfCYAAH0wAAB9OwAAfUgAAH1XAAB9aAAAfHwAAHySAAB7qAAAEsEAAHnfAAB48gAAAd/s
Ad1sAAGpcAABgXAAAVV0AAElfAAA/YwAANGcAACprAAAhbwAAGXMAABF2AAAJeQAAAnsA
AAB9AAAAfWAAAIBAACBBgAAgwsAAIQRACFGAAAhx8AAIgnAACJMgAAiT8AAII0AACIXwA
AiHMAAliJAACHoQAAh7kAAIXWAACF7QAAhPgAcWIAAGZiAABbYgAAT2UAAENpAAA4bQAAL
XIAACN2AAAAegAAEX4AAAIbAAAAgwAAAIUAAACHAAAAiQAAAIsAAACMAAAAjgMAAI8IAAC
RDgAAkxUAAJQdAACWJgAAIzIAAJdCAACWUwAAImcAAJZ+AACVlwAAIK8AAJTJAAC5AAAK

vMAbWgAAGJoAABVawAASG8AADt0AAAweQAAJX4AABqCAAARhgAAB4oAAACNAAAAjgAA
AJAAAACSAAAAIAAAAJYAAACXAAAAmQAAAJsAAACdBAAAnwsAAKETAAcJHAAApScAAKU
1AACIRwAApVsAAKRyAACkjAAApKQAAKO7AACj1AAAoucAam8AAFTyAABOdgAAQHsAADSB
AAAOhgAAHlsAABKQAAAlkwAAAJYAAACZAAAAmgAAAJwAAACeAAAAoAAAAKIAAACkAAAA
pQAAAKcAAACpAAAAqwAAAK0HAACwEAAAsxsAALUoAAC1OQAAAtU4AALVIAAC1fQAAAtZYAA
LWsAAC1wgAAAtNQAY3kAAFR9AABGgwAAOYkAACuPAAAGlAAAFZkAAAQdAAAAoAAAAKMAA
ACmAAAApWAAAKkAAACrAAAArQAAAK8AAACxAAAAswAAALYAAAC4AAAAuwAAAL0AAADB
AQAAxQwAAMkYAADMKAAAZTsAAM1TAADMbAAAZlYAAAM2dAADNrWAAZb8AXIQAAE2KAAB
AkAAAMpcAACWdAAAZoQAADaYAAAKrAAAAArgAAALEAAAC0AAAAAtAAAAALYAAAC4AAAAugA
AAL0AAAC/AAAAwQAAAMQAAADHAAAAygAAAM4AAADTAAAA2QAAAN8JAADmFwAA6CoA
AOo/AADsVwAA7W8AAO2HAADsnAAA66kA/wAAAP8AAAD/AAAA/wACAP8ADQD/ABkA/wAjAP
8ALgD+ADgA9wBCAPQASwDxAFMA7wBaAOwAYADqAGYA6QBrAOcAcADIAHYA5AB8AOIAgg
DgAlkA3gCRANwAmwDZAKYA1gCzANMAxQDQAOAAzQD3AMsA/wDJAP8AxgD/AL8A/wC8AP
8A/wAAAP8AAAD/AAAA/wAAAP8ACQD+ABUA/AAfAPQAKgDwADQA7AA9AOcARgDkAE4A4Q
BVAN4AXADcAGEA2gBnANgAbADVAHEA0wB2ANIAfQDPAIQAzQCMAMsAIQDJAKAAxgCtAM
QAvgDBANYAvwDwALwA/wC6AP8AuQD/ALUA/wCyAP8A/wAAAP8AAAD/AAAA/gAAAPYABQD
xABEA5wAbAOMAJQDgAC8A3AA4ANYAQQDSAEkAzWBQAMwAVgDKAFwAyABhAMyAZgDE
AGwAwgBxAMEAdwC/AH4AvQCGALsAjwC5AJ0AtwCnALQAtgCyAMwAsADoAK4A/ACsAP8Aq
wD/AKsA/wCoAP8A/wAAAP8AAAD6AAAA8AAAAOYAAgDbAAwA0gAXAM4AIQDLACoAyQAZA
MQAPADBAEQAvGBLALsAUQC5AFYAtwBcALUAYQC0AGYAsGbsALAAcgCuAHgArACAaKoAig
CoAJQApwChAKUARwCjAMIAoQDeAKAA9QCeAP8AnQD/AJwA/wCcAP8A/wAAAPwAAADsAAA
A3gAAANIAAADFAAgAwAASALwAHAC5ACUAtwAuALQANgCxAD4ArQBFAKsASwCoAFEApwB
WAKUAWwCkAGEAowBmAKEAbACgAHMAngB7AJ0AhACbAl8AmQCbAJcAqQCVALoAkWDSA
JIA7QCQAP8AjwD/Al8A/wCOAP8A/QAAAO0AAADcAAAAywAAAL0AAAC1AAQAsAAOAKwAFw
CpACAApWApAKQAMQChADgAnwA/AJ0ARgCbAEwAmQBRAJgAVgCXAFsAIQBhAJQAZwCTA
G4AkQB1Al8AfwCOAlkAjACVAIoAowCIALMAhgDIAIU5QCDApOAggD/AIEA/wCBAP8A8wAAA
N0AADJAAAAuwAAAK8AAACoAAAAowAJAJ8AEwCcABsAmQAKAJcALACVADMAkgA6AJAA
QQCPAEcAjQBMAIwAUQCKAFcAiQBcAlgAYgCGAGkAhQBxAIMAegCBAlQAgACQAH4AngB8A
K0AegDAAHgA3QB3APMAAdQD/AHQa/wB0AP8A5gAAAM4AAAC7AAAArAAAAKMAAACdAAAAI
wAFAJMAADgCQABcAjQAFAlSjWcJAC4AhwA1AIUAPACDAEIAggBHAIAATQB/AFIAfQBYAHwA
XgB7AGUAeQBtAHgAdgB2AIAAdACMAHIAmgBwAKkAbgC6AGwA1ABrAO0AagD8AGkA/wBoA
P8A2gAAAMIAAACvAAAAowAAAJkAAACSAAAAjgAAAlkACQCGABIAgwAaAIEAIGb+ACoAfQAx
AHsANwB5AD0AdwBDAHYSQB0AE4AcwBUAHIAWgBwAGEAbwBpAG0AcgBrAH0AaQCJAGc
AlwBIAKUAYwC3AGEAzgBgAOgAXgD4AF0A/wBcAP8AzWAAALgCAACnBwAAmggAAJAIAACIB
wAAgWMAAIAABAB9AAwAewAVAHgAHgB1ACUAcwAsAHEAMwBvADkAbgA/AGwARQBrAEsA
aQBRAGgAVwBnAF8AZQBnAGMAcABiAhSAYACHAF4AIQBcAKQAWgC0AFgAywBVAOYAVAD
2AFMA/wBTAP8AxQYAAK8MAACfEAAAKxEAAIgSAACAEQAAew4AAHcKAAB2BQYAcwEQA
HAGABuAiEAawMoAGkELwBnBTUAZQY8AGQGQgBjB0gAYQhOAGAIvQBeCVwAXQIIAFsKbg
BZCnoAVwqGAFULIABTCqQAUAq1AE4JzABMCeUASwn3AEsJ/wBKCf8Avg0AAKkUAACZFgAA
jBgAAIEZAAB5GAAAcxYAAG8TAABtEAAAbAsJAGkKEwBmCwxAZA0kAGIOkWBgDzIXhA4AF
0QPwBbEUUAWWhFMAFgSUwBXElOAVRNjAFMTbQBRFHgATxSFAE0UIABKFKQASBO1AEYSz
QBEEugAQxL6AEMS/wBDEf8AuBMAAKQZAACUGwAAhh4AAHweAABzHgAABR0AAGgaAABIG

AAAZBUDAGITDgBfFBgAXBUGAFoWKABYFy8AVxg1AFUYOwBUGUIAUhJAFEaUABPGlgATRt
hAEwbawBKG3YASBuDAEUbkgBDG6IAQRuzAD8aywA9GucAPRn6ADwY/wA8GP8AsxgAAJ8d
AACQIAAAgilAAHcjAABulwAAZyIAAGlgAABfHgAAXRwAAFsbcgBYGxUAVR0dAFMdJQBRHiw
AUB8yAE4fOQBNID8ASyBGAEohTQBIIVUARyFeAEUhaQBDInQAQSKBAD8ikAA9lqAAOyGyA
DggyQA3IOYANh/5ADYe/wA2Hf8Arh0AAJshAACMJAAafiYAAHlnAABpJwAAyiYAAF0IAABalwA
AVyEAAFQhBgBSIhEATyMaAE0klgBLJCkASiUvAEgmNgBHJjwARSZDAEQnSwBCJ1MAQSdcA
D8nZgA9J3IAOyd/ADknjgA3J58ANSewADlmyAAxJeQAMCT4ATAj/wEwlv8BqSEAAJgkAACIJwA
AeikAAG8qAABIKgAAXioAAFkpAABVJwAAUiYAAE4nAwBMJw4ASSgXAEcpHwBfKiYARCstAEI
rMwBBLDoAQCxBAD4sSAA8LFEB0yxaATksZAE3LXABNS19ATMtjAExLJ0BLyyvACwrxgArKuM
AKin3ASoo/wEqJ/8BpiQAAJQnAACEKwAAadywAAGstAABiLgAAWi4AAFUtAABQLAAATSsAAEk
sAABGLQsARC4UAEIvHABALyQAPjAqAD0wMQA8MTcAOjE+ADkxRgA3MU4BNTJYATMyYgEx
Mm4BLzJ7AS0xiwErMZwBKTGuACcwqAIL+IAJS73ASUt/wEILP8BoicAAJErAACBLgAAcy8AAG
gwAABeMQAAVzEAAFEExAABMMAAAARzAAAEQxAABBMgcAPjMRADw0GgA6NSEAOTUoADc2
LgA2NjUANDY8ADM2QwAxNkwBLzdVAS03YAEsN2wBKjd5ASg2iQEmNpoAJDasACI1wwAgNO
AAIDL2ASAx/wEgMP8BnysAAI0uAAB9MQAAcDIAAGQzAABbNAAAUzQAAE00AABINAAAQjUA
AD42AAA7NwQAODgOADY5FgA0Oh4AMzolADE7KwAwOzIALjs5AC07QQAro0kAKjXTASg8Xg
EmPGGoBJDx3ASM7hwAhO5kAHzurABw6wgAbOd8AGzj1ABs2/wEbNf8Bmy4AAIoxAAB6NAAAb
DUAAGE2AABYNwAAUDcAAEo4AABEOAAAPT0AADk7AAA1PAAAMj4KAC8+EwAuPxsALD8h
ACtAKAApQC8AKEA2ACdBPgAIQUYAJEFQACJBWwAgQWcAH0F1AB1AhQAbQJcAGT+qABc/
wAAVPt4AFTz0ABU7/wAWOv8BIzIAAIY1AAB2NwAAADkAAAF06AABUOgAATTsAAEY7AABAPA
AAOT4AADRAAAAvQgAAK0MGACHEDgAmRRcAJUUEACRFJAAiRisAIUYyACBGOgAeRkMAH
UZNABtGWAAaRmUAGEZzABZGgwAURpUAEkWoABBEvgAOQ9sADkLzABBA/wAQP/8AkzUA
AIE5AABYogAAZDwAAFo9AABRPQAAST4AAEM/AAA9QAAANUIAADBEAAAqRgAAJUgBACF
KCgAeSxIAHUaABxMIAAaTCcAGUwuABhMNgAXTD8AFU1JABRNVAASTWEAEEExvAA5MgAA
MTJEACKukAAhKuQAHSdQAB0jtAAhG/QAIRf8Ajj0AAHw9AABtPgAAYD8AAFZAAABNQQAArk
EAAEBCAA5RAAAMkYAACTJAAAMSwAAIE4AABpQBAAWUgwAFFMUABJTGWARUyIAEFMp
AA5TMQANUzoADFNEAApTTwAIU1wAB1NqAAVTegADUowAAVKgAABRtAAAUM4AAE7pAAB
N9wAATP8AiD4AAHdBAABoQgAAXEMAAAFJEAABKRAAAAREUAAD1GAAA1SAAALUsAACdOA
AAhUQAAG1MAABVWAAAOWQcACVoPAAdaFgAGWh0ABVokaANaLAACWjUAAVo+AABaSg
AAWIYAAFpIAABadQAAWYcAAFmbAABYsAAAV8kAAAFbIAABU9gAAVP4AgkMAAHFFAABjRwA
AWEcAAE9IAABISAAAQEKAADhLAAAwTgAAKFEEACFVAAAAbVwAAfVoAAA5cAAAHXwUAAW
AMAABhEgAAYYhkaAGIfAABiJwAAYy8AAGM4AABjQwAAYIAAAGJfAABibwAAYoEAAGGWAAB
hqwAAX8QAAF7hAABd8wAAXP0Ae0kAAGtKAABeSwAAVEwAAE1MAABETQAAO08AADJSAA
AqVQAAIikAABtcAAAUxwAADWEAAAdkAAAAZwIAAGglAABpDgAAahQAAGsaAABslQAAbCgA
AGwxAABsPAAAAbEkAAGxXAABsaAAAa3oAAGuPAABqpgAAab4AAGjdAABn8QAAZvsAdE4AA
GZQAABaUAAAUIAAAEhRAAA+UwAANVYAACxaAAAJXgAAG2EAABRIAAAMaAAABWoAAABt
AAAAbwAAAHADAABxCAAACw4AAHQUAAB1GgAAAdiEAAHgpAAB3NAAAd0AAAHdPAAB3XwA
Ad3IAAHaIAAB2nwAAdbcAAHPVAABz7QAACvkaBvQAAGFVAABYVQAATIYAAEJYAAA4XAAA
LmAAACRkAAACaAAAE2wAAAtvAAAEcQAAAHQAAAB2AAAAeAAAAHoAAAB7AgAAfQcAAH4
MAACAeWAAgRkAAIMhAAACEKwAAHdCAAIRFAACEVgAAg2kAAIN/AAAClWAAgBAAAIDMAAB/6
AAAFvYAAFoAAF5aAABUWwAAR14AADxhAAAwZgAAJmsAABxvAAAUcwAAC3cAAAj6AAAAfA
AAAH4AAACBAAAAggAAAIQAAACGAAAAiAAAAIkFAACLCgAAjREAAI8ZAACRIgAAki0AAJI7A

ACSTAAAKV8AAJF1AACQjgAAj6cAAI7CAACO3wAAjfAAZWAAAFphAABNZAAAQGgAADRtAAA
ocgAAHncAABR8AAAKgAAAAAYMAAACGAAAAhwAAALoAAACMAAAAJgAAAJAAAACSAAAAIAA
AAJYAAACYAAAAmgYAAJwNAACfFgAAoSAAAKluAACiPgAAoVIAAKFoAACggQAAn5sAAJ61A
ACezQAANE MAYWcAAFNqAABGbwAAOXQAACx6AAAhfwAAFoQAAAUJAAABjQAAAJAAAACt
AAAAIAAAAJYAAACYAAAAmgAAAJwAAACeAAAAoAAAAKMAAACIAAAApwAAAKkCAACsCwA
ArxUAALihAACyMQAAskQAALFbAACwdAAAsl8AAK+nAACvvgAAr9IAWnEAAEx2AAA/ewAAMY
IAACWHAAAZjQAADZIAAAKWAAAAmgAAAJ0AAACgAAAAoQAAAKMAAACIAAAApwAAAKkAA
ACrAAAArQAAAK8AAACyAAAAAtAAAAALcAAAC7AAAAvvgAAMMUADFlwAAxTYAAMROADF
ZgAAxn8AAMaXAADGqWAAxrwAU30AAEWDAAA4iQAAKpAAAB6WAAASmwAABqAAAACkAA
AApwAAAKoAAACtAAAAArgAAALEAAACzAAAAAtQAAALcAAAC5AAAAvAAAAAL8AADCAAAAQ
AAAMkAAADNAAAA0gAAANkEAADgEgAA4yQAAOQ5AADIUgAA5GwAAOSGAADkmgAA5akA/
wAAAP8AAAD/AAAA/wAAAP8ACQD/ABUA/wAgAP8AKgD9ADQA+QA+APYARwDzAE8A8ABW
AO4AXADsAGEA6gBnAOgAbADmAHEA5AB3AOMAfQDhAIQA3gCNANwAlgDZAKIA1gCvANMA
wgDQAN4AzAD3AMoA/wDHAP8AwgD/ALsA/wC2AP8A/wAAAP8AAAD/AAAA/wAAAP8ABgD5A
BEA9QAcAPMAJgDxAC8A7AA5AOgAQgDkAEoA4QBRAN4AVwDbAFwA2QBiANUAZwDTAGw
A0ABYAM4AeADLAH8AyACHAMYAkQDDAJwAwACpAL4AuQC8ANIAugDvALkA/wC3AP8AtgD/
ALAA/wCsAP8A/wAAAP8AAAD/AAAA+wAAAPQAAgDoAA0A5AAYAOEAIQDfACsA2wA0ANUA
PADPAEQAYgBLAMcAUQDEAFcAwgBcAMAAyQC+AGcAvQBsaLsAcgC5AHkAtwCBALUAIwCz
AJYAsQCjAK8AsgCtAMgAqWdnAKkA/QCnAP8ApgD/AKQA/wChAP8A/wAAAP8AAAD3AAAA6w
AAANsAAADUAAkAzgATAMsAHQDIACYAxAAuAL4ANwC6AD8AuABGALUATACzAFIAsQBXAL
AAXACuAGEArQBnAKsAbQCqAHQAqAB7AKcAhQCI AJAAowCcAKEAqWcfAL4AnQDcAJsA9QC
ZAP8AmAD/AJcA/wCVAP8A/wAAAPQAAADIAAAA1QAAAMcAAAC/AAUAUGAOALcAGACyACE
AsAApAK4AMQCrADkAqABA AKYARgCkAEwAowBRAKEAVgCgAFwAngBhAJ0AZwCbAG4Amg
B1AJgAfwCWAIkAIACWAJIAPACRALUAjwDPAI0A7QCLAP8AigD/AlkA/wCJAP8A+QAAAOUAAA
DRAAAAvwAAALUAAACvAAEAqgAKAKUA EwCiABwAoAAkAJ8ALACcADMAMgA6AJgAQQCW
AEYAIABMAJMAUQCRAFyAkABcAI8AYQCNAgGajABwAlOAEQCIAlMAhgCQAIQAngCDAK4Ag
QDEAH8A4wB+APkAfAD/AHsA/wB7AP8A6QAAANIAAAC/AAAAAsQAAAKgAAACHAAAAmgAGAJ
cADwCUABgAkgAgAJEAJwCPAC4AjQA1AlOAOwCJAEEAhwBHAIUATACEAFEAggBXAIEAXQB
/AGMAfQBrAHwAdAB6AH4AeACLAHcAmQB1AKgAcwC8AHIA2ABxAPEAbwD/AG8A/wBuAP8A
3QAAAMQAAACwAAAApAAAAJsAAACUAAAAJwACAlsACwCIABMAhgAbAlQAlwCCACoAgAAw
AH4ANwB7ADwAegBCAHgARwB3AE0AdgBSAHQAWABzAF8AcQBnAHAAcABuAHoAbQCGAG
sAIABpAKMAZwC1AGYAzQBkAOoAYwD7AGIA/wBiAP8AzgAAALYAAACnAAAAmgAAAJAAAA
CJAAAAhAAAAIAABwB9AA8AewAXAHkAHgB2ACUAdAAAsAHIAMgBxADgAbwA+AG4AQwBsAE
gAawBOAGoAVABoAFsAZwBjAGUAbABKAHYAYgCCAGAAkABeAJ8AXACwAFsAxgBZAOMAW
AD2AFcA/wBXAP8AwwAAAK0AAACeAgAAkQQAAlcEAAB/AgAAewAAAHcAAgB0AAoAcgATAG
8AGgBtACEAawAoAGkALgBnADQAZgA5AGQAPwBjAEUAYQBKAGAAUQBfAFgAXQBgAFwAa
QBaAHMAWAB/AFYAjQBUAJwAUwCtAFEAwgBPAN0ATgDxAE0A/wBMAP8AuwAAAKYHAACX
CgAAiQwAAH8MAAB3DAAACgkAAG4GAABsAQUAagANAGcAFQBIAB0AYwAkAGEAKgBfADAA
XQA2AFwAOwBaAEEAWQBHAFgBTgBWAIUAVQJdAFMCZwBRA3EATwN+AE0DjABLA5sASQ
OsAEcCwABGAtkARALwAEQC+wBDav8AsgkAAKANAAACREQAAGxMAAHgUAABwEwAAahIAA
GYQAABjDAAAYwgHAGEEEABeBRgAWwYgAFkHJgBXCC0AVggzAFQJOABTCj4AUgpFAFALT
ABPC1MATQxcAEsMZgBJDHEARw1+AEUNjABDDZwAQQytAD8MwQA9C9wAPAvwADwL/gA8

Cv8ArQ4AAJsTAACLFwAAfRgAAHIZAABpGQAAYxgAAF8WAABcFAAAWxEBAFoNCgBXDRQA
VQ4cAFMQIwBRESkATxEvAE4SNQBMEjwASxNCAEkTSgBIFFEARhRaAEQUZABCFW8AQBV
8AD4VwA8FZsAOhSsADcUwgA1E94ANRP0ADUS/wA1Ev8AqBMAAJYXAACGGwAAeR0AAG0
dAABkHgAAXh0AAFkcAABWGGAAVBgAAFMVBQBRFRAAThYYAEwXIABKGcYASBgsAEcZMw
BFGTkARBPAAEIaRwBBG08APxtYAD0bYgA8G20AOht6ADgbiQA1G5kAMxurADEawAAuGdwA
LhnzAC4Y/wAuGP8ApBcAAJlbAACCHwAAAdCAAAGkhAABglgAAWSEAAfQhAABRHwAATh0AA
EwcAQBKHAwASB0VAEUdHQBEHiMAQh8qAEEfMAA/IDYAPiA9ADwgRAA7IU0AOSFWADchY
AA1IWsAMyF4ADEhhwAvIZgALSGpACogvgAoH9sAKB7yACge/wAoHf8AoBsAAI8eAAB+lgAAcS
QAAGUIAABcJQAAVSUAAFAIAABMJAAASSEAAEchAABEIggAQiLSAEAjGgA+JCEAPCQnADsl
LQA5JTQAOCY6ADcmQgA1JkoAMyZTADEnXgAwJ2kALid2ACsnhQApJpYAJyaoACUIvQAjJNk
AlyPxACMj/wAjlV8AnR4AAIsiAAB7JQAAbSYAAGIoAABZKAAAUigAAEwoAABIKAAAARSYAAEE
mAAA+JwUAPCgOADooFwA4KR4ANyokADUqKwA0KzEAMis4ADERPwAvK0gALSxRACwsWwA
qLGcAKCx0ACYshAAkK5UALiunACAquwAdKdgAHSjwAB4n/wAeJv8AmSEAAIcAAB3KAAAaikA
AF4rAABVKwAATiwAAEksAABEKwAAQCsAADwrAAA5LAIANI0LADQuFAAyLhsAMS8iAC8vKAA
uMC8ALDA1ACswPQApMUUAKDFPACYxWQAKMWUAljFyACAxggAeMJMAHDCIABovugAYLtY
AGC3wABks/wAZK/8AliQAAIQoAAB0KwAAZiwAAFsuAABSLgAASy8AAEUvAABBLwAAPC8AAD
cwAAAzMQAAMTIIAC4zEQAsNBgAKzQfACK1JQAoNSwAJJuzACU2OgAkNkMAIjZMACA2VwAf
NmMAHTZwABs2gAAZNZEAFzWkABU0uQATM9QAEzLvABQx/gAUMP8AkigAAIArAABwLgAAY
zAAAFgxAABPMQAASDIAAEIyAAA9MgAAODMAADE1AAAtNgAAKjgEACc5DQAmORUAJDcA
CM6lgAhOikAIDsvAB87NwAdO0AAHDtJABo7VAAZO2AAfztuABU7fgATOO8AETqiAA45twAMO
NEADDfsAA02/QAONf8AjisAAHwvAABsMQAAXzMAAFQ0AABMNQAARTUAAD81AAA6NgAAN
DcAAC05AAApOwAAJD0AAce+CQAePxEAHT8YABxAHwAaQCUAGUAsABhAMwAWQTWAFU
FGABNBUQASQV0AEEBrAA1AewAMQlwACj+fAAg+sgAGPcwABTznAAY7+QAHOv8AiS8AAHc
yAABoNQAAWzYAAFE3AABJOAAAQjgAADw4AAA3OQAAMDsAACo9AAAIpWAAIEEAABtEBAA
XRQwAFUYUABNGGwASRIEAEUYoABBHLwAORzgADEdCAAtHTQAJRIkAB0ZnAAZGdgAERo
gAAkWbAABErgAAQ8YAAELkAABB9AAAQP8AhDQAAHI2AABjOAAAVzoAAE47AABGOwAAPz
sAADo7AAAzPQAALT8AACZCAAahRAAAG0YAABZJAAAQSwcAC00PAAINFgAITR0AB00kAAZ
NKwAETTMAA009AAFNSAAATVQAAE1iAABMcQAATIMAAEUWAABLqWAASSIAAEjfAABH8wA
AR/wAfjkAAG07AABfPAAAUz0AAEo+AABDPgAAPT4AADY/AAAvQQAakeQAACFHAAAcSQAA
FkwAABBOAAAKUQUABFMMAABTEgAAVBkAAfQfAABUJgAAVC4AAfQ3AABUQgAAVE4AAF
RcAABUbAAAu34AAfOSAABSpwAAUb4AAFDcAABP8AAATvwAeD0AAGg/AABaQQAaUEEAA
EdBAABBQQAOKIAADJEAAAqRwAAI0oAABxNAAAWUAAAEFIAAApVAAADWAMAAfKJAABa
DgAAWxQAAfwaAABclQAAXSkAAf0yAABdPAAAXEgAAfXWAABcZgAAXHgAAFuMAABbowA
AWroAAfjYAABX7wAAVvoAcUIAAGJEABWRQAATEUAAEVFAAA9RgAANEgAACxLAAAkTgA
AHVEAABZVAAAQWAAACVoAAAJdAAAAAXwEAAGAGAABiCgAAYxAAAGQVAABIHAAAZiMAA
GcrAABmNQAAZkEAAGZPAABmXwAAZXEAAGWGAABknQAAY7QAAGLRAABh7AAAYPkAakg
AAF1JAABSSQAASKkAAEJKA4TAAALk8AACZTAAAdVwAAFIoAAA5eAAAHYAAAAGMAAAB
mAAAAaAAAAGkBAABrBgAAbAoAAG4QAABvFgAAcBwAAHlkAABylgAAcjoAAHJIAABxWAAAc
WoAAHB/AABvlgAAb64AAG3LAABt6AAAa/gAZE4AAfHOAABQTgAARk4AADxRAAAxVAAAJ1g
AAB5cAAAWYQAADWQAAAzoAAAAagAAAG0AAABWAAAAcgAAAHQAAAB1AAAAdwMAAHkl
AAB7DQAAfBQAAH4cAACAJAAAgDAAAH89AAB/TQAAf2AAAH51AAB9jQAAfKYAAHvBAAB64
AAAEfIAX1QAAfZTAABMVAAAQFYAADVaAAAgXwAAIGQAABZoAAANbAAABHAAAABzAAAA

dgAAAHkAAAB7AAAAfQAAAH8AAACBAAAAgwAAAIUAAACGBgAAiQwAAIsTAACNHAAAjiYAA
l4zAACOQwAAjVYAAI1rAACMgwAAi50AAIq3AACJ1gAAiOwAXFkAAFJaABFXAAAOWEAAC1
mAAAIawAAF3AAAA11AAAEeQAAAH0AAAB/AAAAGgAAAIQAAACGAAAAiAAAAIsAAACNAAA
AjwAAAJEAAACTAAAAIQMAAJgKAACaEgAAnRwAAJ4oAAceOAAAnUoAAJ1gAACceAAAm5lA
AJqtAACZyAAAmOMAWWAAAEtjAAA+aAAAMm0AACVzAAAaeQAAD34AAASDAAAAhwAAAlO
AAACMAAAAJgAAAJAAACTAAAAIQAAAJgAAACaAAAAAnAAAAJ4AAACHAAAAowAAAKYAAA
CpBgAArBAAALAbAACwKgAArzwAAK9RAACuagAArYQAAKyfAACrtwAAqs8AUmoAAERvAAA3
dQAAKnsAAB6BAAAShwAABowAAACRAAAAIAAAAJgAAACZAAAmwAAAJ0AAACfAAAAogA
AAKQAAACmAAAAqQAAAKsAAACuAAAAsAAAAALQAAAC3AAAAvAMAAMEOAADEHAAAxC4A
AMNEAADCXAAAwXcAAMCRAADAqAAAwLoAS3YAAD18AAAwgwAAI4kAABaQAAAKIQAAAJs
AAACeAAAAogAAAKUAAACmAAAAqAAAAKoAAACsAAAAArgAAALEAAACzAAAAtgAAALgAAA
C8AAAAvWAAAMMAAADHAAAAzQAAANMBAADaDgAA2yAAANs1AADbTwAA3GcAAN2AAADe
lgAA3qcA/wAAAP8AAAD/AAAA/wAAAP8ABgD/ABIA/wAcAP8AJgD/ADAA/AA6APgAQgDyAEoA7
gBRAOsAVwDoAF0A5gBiAOUAaADjAG0A4QBzAN8AeQDdAIAA2gCJANgAkwDUAJ4A0QCsAM
4AvwDLANwAyAD3AMUA/wDBAP8AuAD/ALMA/wCwAP8A/wAAAP8AAAD/AAAA/wAAAPwAAg
D4AA0A9gAYAPQAIGdyACsA6gA1AOQAPQDgAEUA3ABMANkAUgDWAfGgA1ABdANIAYwDQA
GgAzQBtAMsAcwDJAHsAxwCDAMUAjQDCAJgAvwCmAL0AtwC6ANAAuADwALUA/wCzAP8Ar
QD/AKgA/wCIAP8A/wAAAP8AAAD/AAAA9gAAAOwAAADnAAkA4wAUAN4AHgDZACcA1QAvAM
8AOADLAEAAyABHAMUATQDCAFIawABYAL4AXQC8AGIAuwBoALkAbgC3AHQAAtQB9ALMAh
wCxAJIArwCfAK0ArwCqAMUAqADmAKYA/gCkAP8AogD/AJ0A/wCbAP8A/wAAAP8AAADyAAAA
4QAAANcAAADQAAUAYqAPAMMAGQDBACIAwAAqALwAMgC4ADoAtQBBALIARwCwAE0Arg
BSAK0AVwCrAFwAqgBiAKgAaACnAG8ApQB2AKMAgAChAlsAoACYAJ0AqACbAlSAmQDaAJc
A9gCWAP8AIQD/AJEA/wCPAP8A/QAAAO4AAADeAAAAyWAAAMEAAAC5AAEAswALALAAAFAC
uAB0ArAAIAKoALQCnADQApAA7AKIAQQCgAEcAngBMAJwAUgCaAFcAmABcAJYAYgCVAGk
AkwBxAJEAegCQAIUAjgCSAlwAoQCLALIAiQDMAIgA7QCHAP8AhgD/AIUA/wCDAP8A8AAAAAN
wAAADEAAAAuAAAAK8AAACnAAAAogAHAJ8AEACdABgAmwAgAJ0AKACXAC8AkW2AJEAP
ACPAEIAjQBHAIwATACKAFEAIQBXAIGAXQCGAGMAhQBrAIMAdACBAH8AgACMAH4AmgB8A
KsAegDBAHkA4gB3APsAdgD/AHYA/wB2AP8A4AAAAAMgAAAC1AAAAqQAAAJ8AAACZAAAAIA
ADAJEACwCOABQAIwAcAlkAlwCIACoAhQAxAIMANwCBADwAgABCAH4ARwB9AEwAewBSA
HoAVwB4AF4AdwBmAHUAbwBzAHkAcgCGAHAAlABvAKUAbQC4AGsA1QBqAPIAaQD/AGkA/
wBoAP8A0AAAAALgAAACoAAAAAnAAAAJIAAACMAAAAhwAAAIQACACAABAAfgAYAHwAHwB7
ACYAeQAsAHYAMgB0ADcAcwA9AHEAQgBwAEcAbgBNAG0AUwBsAFkAagBhAGkAagBnAHQ
AZQCAAGQAjwBiAJ8AYQCxAF8AygBeAOkAXQD8AFwA/wBcAP8AxAAAAK4AAACeAAAAkQA
AAIgAAACBAAAAfAAAAHgABAB0AAwAcgAUAHAAGwBuACEAbQAoAGsALQBpADMAaAA4AG
YAPQBIAEMAYwBIAGIATgBhAFUAXwBdAF4AZgBcAHAAWgB8AFkAigBXAJ0AVgCrAFQAwQB
TAOAAUgD2AFEA/wBRAP8AtwAAAKUAAACVAAAAiAAAAH4AAAB3AAAAcgAAAG4AAABrAAg
AaAAQAGYAFwBkAB0AYwAjAGEAKQBfAC8AXgA0AFwAOQBbAD8AWQBFAFgASwBXAFEAV
QBZAFQAYgBSAG0AUAB4AE8AhgBNAJYATACnAEoAuwBJANcASADvAEcA/gBHAP8ArwAAA
J4BAACOBgAAgQcAAHYIAABuCAAAaAYAAGUDAABjAAMAYAALAF4AEgBcABkAWgAgAFgAJ
QBXACsAVQAwAFMANgBSADsAUQBBAE8ARwBOAE4ATQBWAEsAXwBKAGoASAB2AEYAh
ABEAJMAQwCkAEEAtwBAANAAPwDqAD4A+AA9AP8AqQMAAJcIAACHDAAAeg4AAG8OAAB
mDwAAYA4AAFwLAABaCAAawQQGAFcBDQBVABUAUwAcAFEAlgBPASgATQItAEwDMgBLA

zgASQQ+AEgERQBHBewARQVUAEQFXgBCBmgAQAZ1AD4GgwA8BpIAOwajADkFtgA3BM0A
NgPoADUD9wA0A/8ApAgAAJIOAACCEQAAdBMAAGkUAABgFAAAWhQAAFYTAABTEAAAUQ
0AAFEKCABPBxAATQgYAEoJHgBJCiQARwoqAEYLMABECzYAQww8AEIMQwBADUsAPw1TA
D0NXQA7DmgAOQ51ADcOhAA0DpQAMg6IADANuAAuDNAALQzpAC0L+AAc/8Anw0AAI0TAA
B9FQAAbxcAAGQZAABbGQAAVRkAAAFAYAABNFwAASxQAAEoSawBJEAsARxATAEQRGwBD
EiEAQRInAD8TLQA+EzMAPRQ5ADsUQQA6FEgAOBVRADYVWwA0FWcAMhVzADAVggAtFZI
AKxWkACkUtwAnE9EAJhPsACYS+wAmEv8AmxEAAIkWAAB5GQAAaxsAAGAcAABXHQAUB
0AAEscAABIGwAARRoAAEQXAABCFwcAQBYQAD4XGAA8GB4AOxkkADkZKgA4GTAANho3A
DUaPgAzG0YAMRtPADAbWQAuG2QAKxtxACkbgAAng5EAJRujACMatgAhGdAAIBnrACAY/AA
hF/8AlxUAAIUaAAB1HAAAZx8AAFWgAABTIAAATSAAAEcgAABDHwAAQB4AAD8cAAA8HAMA
Oh0MADgdFQA2HhwANR4iADMfKAAyHy4AMCA0AC8gOwAtIEQAKyBNACogVwAoIWIAJiFvAC
QhfgAiI8AHyChAB0ftQAbHs8AGx7qABsd/AAbHP8AIBgAAIEcAABxHwAAZCEAAFFkjAABQlwAA
SSQAAEQjAABAlwAAPCIAADohAAA3IQAANCIJADIjEgAwlxkALyQfAC0kJQAsJCsaKiUyACkIO
QAnJUEAJiZKACQmVQAIjMAAICZuAB4mfAAcJY0AGiWgABgksAWI80AFSLpABYh+wAWIf8A
kBsAAH4fAABulgAAYCQAAFYmAABNJgAARiYAAEEemAAA8JgAAOCYAADUmAAAxJgAALycG
ACwoDgAqKBYAKSkACcplgAmKikAJSoVACMqNgAiKj8AICtIAB4rUgAdK14AGytsABkrewAXKo
wAFSqeABMpsgARKMwAECfoABEm+gARJf8AjB8AAHoiAABqJQAAXScAAFIoAABKKQAAQykA
AD4pAAA5KQAANSOAADAqAAAsKwAAKSwCACyTcWakLhMAly4ZACEvIAAGLyYAHy8sAB0wN
AAcMDwAGjBFABkwUAAXMFwAFTBpABMweQARL4oADy+dAA0usAAKLccACizjAAor9wALKv8
AiCIAAHYIAABmKAAAWioAAE8rAABHLAAACQwAADssAAA2LAAAMS0AACsuAAAmMAAAIzIA
ACAzBwAeNA8AHDQWABs0HQAaNSMAGDUpABc1MQAWNTkAFDVCABI1TQARNVKAAdjVnAA
w1dgAKNYcACDSZAAyZrAAFMsmABDHfAAQw8gAFL/8AhCYAAHlpAABjKwAAVi0AAEwuAABE
LwAAPS8AADgvAAAzLwAALjAAACgyAAAJNAAAHZyAABs4AwAXOQsAFToSABQ6GQASOx8A
ETsmABA7LQAOOzUADDs/AA7SQAJO1UABztjAAU6cQADOoIAATmVAAA5qQAAOL8AADfbA
AA28AAANfsAfykAAG4sAABfLwAAUzAAAEkxAABBMgAAOzIAADUyAAAxMgAAKzQAACU2AAA
gOAAAGzoAABY8AAARPwcADEAOAAtAFQAJQBwACEAiAAAdAKQAFQDEABEA6AAJARQABQ
FEAAEBEABABQAAQH4AAD+RAAA+pgAAPbwAADzYAAA77gAAOvsAei4AAGkxAABaMwAAT
zQAAEU1AAA+NQAAODUAADM1AAAtNgAAJzgAAACE6AAAcPQAAFz8AABJBAAAMRAUABkYL
AAJGEgAARxgAAEceAABHJQAARy0AAEc2AABHQAAAR0wAAEdZAABGaQAARnoAAEWNAA
BFogAARLgAAEPUAABB7QAAQfkAdDIAAGM1AABWNwAASzGAAEI4AAA8OAAANjgAADA5AA
ApOwAAIz0AAB1AAAAXQgAAEkQAAAxHAAAGSQQAAsJAABMDwAATRQAAE4aAABOIQAA
TigAAE4xAABOOWAATkcAAE5UAABOZAAATXUAAE2JAABMnwAAS7UAAErSAABJ7AAASpKA
bjcAAF45AABSOwAARzsAAEA7AAA6OwAAMzwAACs+AAAKQAAAHkMAABdGAAARSQAAC0s
AAAVOAAAAUAIAAFIHAABTCwAAVBEEAFUWAABWHAAAVyMAAFcrAABXNQAAV0EAAFD0A
ABWXgAAVm8AAFWDAABVmgAAVLEAAFPNAABR6gAAUPkAZzwAAFK+AABNPwAART8AAD
4+AAA3PwAALkEAACZEAAAFrWAAGEoAABFOAAAKUAAABFMMAABWAAAAWAAAAFoDAAB
bBwAAXAwAAF4RAABfFwAAYR0AAGIIAABhLgAAYToAAGFHAABhVwAAYGkAAAF99AABeIAAA
XqwAAF3IAABb5wAAWvgAYUIAAFRDAABKQwAAQ0IAADtDAAAxRQAAKEgAACBMMAAYTWa
AEVMAAAIWAAACWQAAAFwAAABfAAAAAYQAAAGMAAABkAgAAZgcAAGgLAABpEQAAaxcAA
G0eAABtJwAAbTMAAG1AAABsTwAAbGEAAGt2AABqjQAAaaYAAGjBAABn4gAAZvUAW0cAAF
BIAABIRwAAQEcAADVKAAArTQAAIVEAABIWAAAQWgAACF4AAABhAAAAZAAAAGcAAABpA
AAAawAAAG0AAABvAAAAcQAAAHMFAB1CgAADxAAAHkXAAB7IAAAeyoAAHs3AAB6RwAAe

VkAAHltAAB4hAAAd54AAHa4AAB02gAAc/EAVk0AAE5MAABFTQAAOk8AAC5TAAAjWAAAGlw
AABbHAAAHZgAAAGoAAABsAAAAcAAAAHIAAAB1AAAAdwAAAHoAAAB8AAAAfgAAAIAAAAC
CAgAAhAcAAIcOAACJFgAAIyAAAistAACKOwAAik0AAIhAACIeQAAh5IAAIasAACEyQAAg+cAV
FIAAEtSAAA/VQAAMlkAACZfAAAAbZAAAEWkAAAdvAAAAcwAAAHcAAAB6AAAAfQAAAH8AAA
CCAAAAhAAAAIYAAACJAAAAiwAAAi0AAACPAAAAkgAAAJUEAACXDAAAmxYAAJwhAACcMA
AAm0IAAJpWAACZbQAAMlcAAJaiAACVvAAAINkAUUVkAAERcAAA3YAAAKmYAB5sAAATcgA
ACHgAAAB9AAAAgQAAAIQAAACHAAAAiQAAAIIsAAACOOAAAAkAAAAJMAAACVAAAAIwAAAJo
AAACdAAAAAnwAAAKIAAACIAgAAqQsAAK0XAACuJQAARtYAAKxKAACrYgAAqXwAAKiXAACn
sAAAPskASmIAADxoAAAvbQAAInQAABZ7AAAKgQAAAIYAAACLAAAAjwAAAJEAAACTAAAAIg
AAAJkAAACbAAAAngAAAKAAAAACjAAAAPQAAAKgAAACrAAAAArgAAALEAAAC1AAAAugAAAL8
JAADEFgAAwycAAMI7AADAUwAAv20AAL2HAAC8oQAAu7UAQ24AADV1AAAoewAAG4MAAA
+JAAACjwAAAJQAAACaAAAAAnQAAAKAAAAChAAAAowAAAKUAAACoAAAAqgAAAK0AAACw
AAAAsgAAALUAAAC5AAAAvAAAAAMEAAADGAAAAyWAAANEAAADZCQAA3BoAANstAADaRQ
AA2F8AANZ6AADXkgAA2KUA/wAAAP8AAAD/AAAA/wAAAP8AAwD/AA0A/wAZAP8AlwD+ACw
A+gA1APYAPgDzAEYA8ABNAO0AUwDrAFkA6QBeAOcAZADIAGkA4wBvAOEAdQDeAHwA3A
CFANkAjwDWAJsA0gCqAM8AvADLANwAyAD5AMUA/wC7AP8AsQD/AKwA/wCoAP8A/wAAAP8
AAAD/AAAA/wAAAP0AAAD5AAoA8wAUAPAAHgDvACcA6gAxAOUAOQDhAEEA3QBIANoATgD
XAFQA1ABZANIXgDQAGQAzQBpAMsAcADJAHCxwB/AMQAIQDBAJQAvwCjALwAtAC5AM8
AtgDxALQA/wCvAP8ApwD/AKIA/wCfAP8A/wAAAP8AAAD8AAAA8gAAAOsAAADjAAUA3QAQA
NoAGgDZACMA1QArAM8ANADKADsAxgBCAMMASQDAAE4AvABUALoAWQC3AF8AtQBkALM
AagCxAHEAsAB6AK4AgwCsAl8AqgCdAKgArQCmAMQApQDnAKQA/wCiAP8AmwD/AJYA/wCU
AP8A/wAAAPgAAADoAAAA3QAAANIAAADHAAEAwwALAMEAFQC/AB4AvgAmALgALgCyADY
ArwA9AKwAQwCqAEkAqQBOAKcAVACmAFkApABeAKIAZACHAGsAnwBzAJ0AfQCbAlgAmQC
VAJgApgCWALoAIADbAJIA+QCRAP8AjgD/AIoA/wCIAP8A9wAAAOkAAADSAAAAxAAAAALgAAA
CyAAAArWAHAKwAEACpABkApgAiAKQAKQChADEAngA3AJwAPQCaAEMAmABIAJYATQCVA
FMAkwBYAJEAXgCQAGUAjgBtAlwAdgCKAIEAiQCOAlcAngCFALAAhADMAIIA7wCBAP8AgAD/
AH0A/wB7AP8A6AAAAAM0AAAC9AAAAAsQAAAKcAAACHAAAAAnQADAJkADACWABUAIQAdAJQ
AJACRACsAjgAyAlS AOACJAD0AiABCAIYASACEAE0AgwBSAIIAWACAAAF8AfwBnAH0ABwB7A
HoAeQCHAHcAlwB2AKkAdADAAHMA4wByAP0AcQD/AHAA/wBuAP8A1gAAALwAAACtAAAAoA
AAAJgAAACSAAAAjQAAAIkACACGABEAhAAYAIMAHwCCACYAfWAsAH0AMgB7ADgAeQA9A
HgAQgB2AEcAdABNAHMAUgBxAFkAcABhAG4AagBsAHQAawCBAGkAkABoAKIAZgC2AGUA1
QBkAPQAYwD/AGMA/wBiAP8AwgAAAK8AAACgAAAAIAAAAIIsAAACFAAAAFwAAAHsABQB4AA
wAdgAUAHUAGwBzACEAcgAnAG8ALQBtADMAawA4AGkAPQBoAEIAZwBHAGUATQBkAFQA
YgBcAGEAZQBfAG8AXgB7AFwAigBbAJSAWQCuAFkAyABYAOoAVwD/AFYA/wBWAP8AuAAA
AKUAAACVAAAAiQAAAIAAAAB4AAAAcwAAAG8AAQBsAAkAagAQAGgAFwBmAB0AZQAJAGM
AKQBhAC4AYAAZAF4AOABdAD0AWwBDAFoASQBZAE8AVwBXAFYAYABUAGoAUwB2AFEA
hQBQAJUATwCoAE0AvwBMAOAATAD4AEsA/wBLAP8ArgAAAJwAAACMAAAAgAAAAHUAAAB
uAAAAaQAAAGUAAABiAAUAXwAMAF0AEwBcABkAWgAfAFkAJQBxACoAVgAvAFQANABTAD
kAUgA/AFAARQBPAEsATgBTAEwAXABLAGYASQByAEgAgABGAJEARQCjAEQAtwBDANQAQ
gDwAEEA/wBBAP8ApwAAAJQAAACFAAAAdwMAAG0EAABIBAAAXwIAAFwAAABZAAEAwAI
AFUADwBTABUAUQAbAFAAIQBPACYATQArAEwAMABKADUASQA7AEgAQQBGAEGARQBQ
AEQAWQBCAGMAQQBvAD8AfQA+AI0APACeADsAsgA6AMsAOQDdoADgA+gA4AP8AoAAAAI4

DAAB+BwAAcAkAAGYKAABdCwAAWAoAAFMIAABRBQAATwIEAE4ACwBMABEASgAYAEgAH
QBHACMARQAoAEQALQBDADIAQQA4AEAPgA/AEUAPgBNADwAVgA7AGEAOQBtAdcAewA
2AlOANACcADIArgAxAMUAMADhAC8A9AAvAP4AmwMAAIglAAB5DAAAaw4AAGAQAABYEAA
AURAAAE0OABKDKAASAOAAEcHBgBGBA0ARAMUAEIDGgBABCAAPwQIAD4FKgA8BS8AO
wY1ADoGPAA4BkMANwdLADUHVQAzB2AAMQhsADAlEgAuClkALAEbACoGrQAoBslAJwTdAC
YD8gAmA/0AlggAAIQNAAB0EAAAZhMAAFsUAABSFQAATBUAAEcUAABEEwAAQhEAAEAOA
QBACwgAPwoPAD0KFgA7CxwAOQwiADgMKAA2DC0ANQ0zADMNOgAyDkIAMA5LAC4OVQA
sD2AAKg9tACgPewAmD4sAJA6dAClOrwAgDcYAHgzgAB4M8gAeC/0AkQ0AAH8RAABvFAAAY
hYAAFcYAABOGQAASBgAAEMYAAA/FwAAPBYAADsUAAA6EgMAOBEKADcREgA1EhkAMxMf
ADITJQAwEysALhQxAC0UOAArFUAAKhVJACgVUgAmFV4AJBvRACIWeQAgFYoAHhWcABwUr
wAZE8YAGBPjABgS9QAZEf8AjRAAAHsUAABrFwAAXhoAAFMbAABLHAAARBwAAD8bAAA7G
wAAOBoAADYZAAA0FwAAMhchADAXDwAuGBYALRgcACsZlgAqGSgAKBouACcaNQAIgJ0AJ
BtGACIbUAAgG1wAHhtpABwbDwAaG4gAGBqaABYarQAUGcUAExjiABMX9gATFv8AihMAAHgX
AABoGgAAWx0AAFAeAABIHwAAQR8AADwfAAA4HgAANB4AADEdAAAvHAAALB0EACodDAA
pHhMAJx4aACUeIAAkHyYAlx8sACEfMwAgIDsAHiBEABwgTgAbIFoAGSBnABcgdgAVIIYAEEx+Z
ABEfrAAOHsMADR3fAA0c9AAOG/8AhhYAAHQaAABkHQAAVx8AAE0hAABFIQAAPiEAADkhAA
A0IQAAMCEAAC0hAAApIQA AJyIAACUiCAAjlxAAISMxACakHQAeJCMAHSQpABwkMAAAJTgA
GSVBABcITAAVJVgAEyVIABEIdAAPJYUADSSWAAsjqQAJI74ABYHaAAch7wAIIP0AghkAAHAd
AABhIAAAVCIAAEojAABCJAAAOyQAADYkAAAxJAAALSQAACkkAAAIJQA AIScAAB8nBQAdKA
wAGykUABopGgAYKSAAFyknABYqLgAUKjYAEyo/ABEqSQAPKIUADSpiAAsqcAAJKoEABymTA
AUopgADJ7sAAibVAAlI7QACJfoAfhwAAGwgAABdlwAAUSUA AEcmAAA/JgAAOScAADMnAAAU
JwAAKicAACyOAAAhKQAAHSsAABosAQAXLgkAFC8QABMvFwASLx0AEC8kAA8vKwANLzIAD
C87AAovRgAIL1EABi9eAAQvbAACL30AAC6PAAAtowAALLgAACvSAAAq7AAAKfkAeiAAAGgjA
ABaJgAATSgAAEQpAAA8KQAANikaADApAAAsKQAAKCoAACQrAAAFIQAAGi8AABYwAAASM
wYADTQNAA s1FAAKNRoACDUhAAc1JwAGNS8ABDU3AAM1QQABNU0AADVaAAA0aQAANH
kAADSMAAAZoAAAMrUAADHQAAAaw6QAAL/gAdSQAAGQnAABWKgAASisAAEEsAAA5LAAAM
ywAAC4sAAAqLAAAJSoAACAvAAAbMQAAFzMAABI1AAANNwUABzoLAAQ6EQABOhcAADsdA
AA7JAAAOysAADszAAA7PQAAO0kAADpWAAA6ZQAAOnUAADmJAAA5ngAAOLMAADbOAAA
16QAANfcAcCgAAF8rAABRLQAARi4AAD4vAAA3LwAAMS8AACwuAAAnLwAAIjEAABwzAAAXN
gAAEjgAAA06AAAIQAQAAj4JAAA/DgAAQBQAAEEZAABBIAAAQScAAEEvAABBOAAAQUQAA
EFRAABBYAAAQHEAAECFAAA/mwAAPrEAAD3LAAA86AAAO/cAaiwAAFovAABNMQAAQzIAA
DsyAAA0MgAALzEAACkyAAAjNAAAHTYAABg5AAASOWAADT4AAAdAAAACQgIAAEQHAABF
CwAARhAAAEcWAABJGwAASSIAAEkqAABJNAAAST8AAEHMAABIWwAASG0AAEEeAAABGlwA
ARa4AAETJAABD5wAAQvgAZDIAAFU0AABJNQA AQDUAADk1AAAzNAAALDUAACU3AAAFoQ
AAGDwAABI/AAAMQgAAB0QAAAFHAAAASQAAAEsEAABMCAAATgwAAE8RAABRFwAAUh0A
AFIIAABSLgAAUjkaAFFGAABRVgAAUGcAAFB7AABPkgAATqoAAE3FAABL5QAAS/cAXTcAAF
A4AABFOQAAPTkaADc4AAAwOQAAKDsAACa9AAAZQAAAEkQAAAxGAAAFSQAAAEWAAAB
PAAAAUQAAAFMAABVBAAAVggAAFgMAABaEgAAWxgAAF0fAABcKAAAXDMAAFxAAABbT
wAAW2EAAFP1AABZjAAAWKUAAAFfAAABW4gAAVPYAVzwAAEs9AABCPQAAPDwAADQ8AAA
rPwAAIkIAABpFAAASSQAAC0wAAARPAAAAUgAAAFUAAABYAAA AWgAAAFwAAABeAAAA YA
MAAGIHAABkDAAAZhIAAGgZAABPlgAAaSwAAGg4AABoSAAAZ1kAAGZtAABlhAAAZJ0AAGO4
AABh2gAAYPMAUkEAAEHCAABBQQA AOUEAAC5DAAAKRwAAG0sAABNPAAAKUwAAA1YAA

ABaAAAAXQAAAGAAAABjAAAAZgAAAGgAAABqAAAAbAAAAG4AAABwBQAACgsAAHUSAAB
3GgAAeCQAAHcwAAB2PwAADIAAAHRkAABzewAAcpUAAHCvAABvzwAAbu0ATkYAAEZGAAA
+RgAAM0gAACdMAAAAdUQAAFFYAAApbAAABXwAAAGMAAAABmAAAAagAAAG0AAABvAAAA
cgAAAHQAAAB3AAAAeQAAAHsAAAB9AAAAgAMAAIIJAACFEQAAiBsAAIgnAACHNQAAhkCAA
IVaAACDcQAAgosAAICIAAB/wQAAfulATEsAAERLAAA4TgAAK1MAACBYAAAVXQAACmMAAA
BoAAAAbQAAAHAAAAB0AAAAdwAAAHoAAAB9AAAAfWAAAIIAAAACEAAAAhwAAAIkAAACMAA
AAjgAAAJEAAACUCAAAmBEAAJocAACZKgAAmToAAJhOAAcXYwAAIX0AAJOXAACSsQAAkc
0ASVEAAD1UAAAwWQAAI18AABdIAAAMbAAAAAXIAAAB3AAAAfAAAAH8AAACCAAAAhQAAAI
gAAACKAAAAjQAAAJAAAACSAAAAIQAAAJcAAACaAAAAAnQAAAKAAAACkAAAAqAUAAKwQA
ACtHgAArC4AAKtBAACqWAAAqHEAAKaLAAClpQAAo70AQIsAADVgAAAoZgAAG20AAA51AAA
DewAAAIeAAACGAAAAiWAAAlwAAACpAAAAkgAAAJQAAACXAAAAmgAAAJwAAACfAAAAoQ
AAAKQAAACnAAAAqWAAAK4AAACyAAAAtwAAALwFAADCEgAAwyIAAME1AAC/TAAAvWUAA
Lt/AAC5mgAAAt7AAO2cAAC5uAAAgdQAAFH0AAAEeAAAAigAAAJAAAACUAAAAmAAAAJkAAA
CcAAAAAnwAAAKEAAACkAAAApWAAAKoAAACtAAAArwAAALIAAAC2AAAAugAAAL8AAADFAA
AAyWAAANIAAADaAwAA4BMAAN4mAADcPAAA2VYAANZwAADUiQAA06AA/wAAAP8AAAD/A
AAA/wAAAP8AAAgD/AAoA/wAVAP8AHwD/ACgA/QAyAPkAOgD2AEIA8wBJAO8AUADrAFYA5wB
cAOQAYQDhAGcA3wBtANwAcwDaAHoA1gCDANQAjgDRAJoAzQCcoAMoAvADIAN8XgD8AME
A/wC1AP8ArAD/AKYA/wChAP8A/wAAAP8AAAD+AAAA/QAAAP0AAAD2AAUA9AARAPIAGgDy
ACQA7AAtAOQANgDcAD4A2ABFANQASwDRAFEazwBXAM0AXADKAGEAyABnAMYAbQDEA
HQAwQB9AL8AhwC8AJMAuQCIALcAswC0ANEAsgD0ALAA/wCmAP8AnwD/AJsA/wCYAP8A/w
AAAPoAAAD2AAAA8QAAAOUAAADgAAEA3QAMANoAFgDUAB8AzWAOAMkAMQDEADgAwQA
/AL4ARQC7AEsAuABRALYAVgC0AFsAsgBhALAAZwCuAG4ArAB3AKoAgACoAlwApgCbAKQAr
ACiAMMAoQDqAKAA/wCZAP8AkWd/AI4A/wCmAP8A+AAAAPAAAADIAAAA1gAAAMsAAADFA
AAAwQAHALoAEgC5ABsAuAAjALMAKwCvADIAqWAA5AKkAPwCnAEUApQBLAKMAUACiAFUAo
ABbAJ4AYQCdAGgAmwBwAJkAeQCXAIUIQCTAJMApACRALgAjwDdAI4A/QCLAP8AhgD/AIIA
/wCAAP8A7gAAAN0AAADMAAAAaVAAAALQAAACuAAAAqAAEAkuADQCjABYAogAeAKAAJQC
dAC0AmgAzAJcAOQCVAD8AkWBEAJEASQCPAE8AjQBUIsAWgCJAGEAhwBpAIYAcgCEAH4
AgwCLAIEAnACAAK8AfgDMAH0A8wB8AP8AeAD/AHUA/wB0AP8A3gAAAMYAAAC2AAAAqQA
AAKIAAACbAAAAIgAAAJMACACQABEAjwAZAI4AIACLACcAhwAuAIQAMwCCADkAgQA+AH8A
QwB9AEgAfABOHsAVAB5AFsAdwBiAHUAawB0AHcAcgCEAHEAIABvAKcAbQC/AGwA5gBsA
P8AawD/AGkA/wBnAP8AyAAAAALQAAACkAAAAmgAAAJIAAACKAAAAhQAAAIIBQCAAAwAfQ
AUAHsAHAB6ACIAeAAoAHUALgB0ADMAcgA4AHAAPQBvAEMAbQBIAGwATgBqAFUAaQBdA
GcAZQBIAHAAZAB9AGIAjABhAJ8AXwC1AF4A2ABdAPkAXQD/AF0A/wBbAP8AugAAAKcAAAC
YAAAAjAAAAIMAAAB8AAAAdwAAAHMAAQBWAAKAbgAQAG0AFwBsAB0AagAjAGgAKQBmAC
4AZAAzAGMAOABhAD0AYABDAF4ASQBdAE8AWwBXAFoAYABYAGoAVwB3AFUAhgBUAJgA
UwCtAFIAyQBRAO4AUQD/AFEA/wBQAP8ArgAAAJwAAACNAAAAgAAAAHcAAABvAAAAagAA
AGYAAABjAAUAYQAMAGAAEwBfABkAXgAfAFwAJABaACkAWAAuAFcAMwBWADgAVAA+AF
MARABRAEOAUABSAE8AWwBNAGUATABxAEOAgABJAJEASACmAEcAvgBGAOIARgD8AEYA
/wBGAP8ApgAAAJMAAAACEAAAAAdgAAAGwAAABIAAAAYAAAAFsAAABZAAIAVgAIAFUADwBU
ABUAUwAbAFIAIABQACUATgAqAE0ALwBLADQASgA5AEkAPwBHAIEYARgBOAEQAVgBDAG
EAQgBtAEEAewA/AlwAPgCfAD0AtQA9ANQAPADzADwA/wA8AP8AnQAAAIwAAAB8AAAAbgAA
AGQAAABcAAAAVgAAAFIAAABQAAAATQAFAEwACwBKABEASQAXAEgAHABHACIARQAmA

EQAKwBCADAAQQA1AEAAOwA/AEIAPQBKADwAUwA7AF0AOQBpADgAdwA2AlgANQCaADQ
ArgAzAMkAMwDpADIA/QAyAP8AlwAAAIUAAAB1AgAAZwUAAF0GAABUBwAATgYAAEoEAAB
HAgAARgACAEQACABDAA0AQQATAEAAGQA/AB4APgAjADwAKAA7AC0AOgAyADgAOAA3A
D8ANgBGADQATwAzAFoAMQBmADAAcwAuAIQALQCWACwAqQArAMEAKgDgACoA9gApAP8
AkQAAAH8DAABvBwAAygoAAFcMAABPDAAASAwAAEQLAABBCQAAPwcAAD4EBAA8AQoA
OwAQADoAFgA4ABsANwAgADUUAJQA0ACoAMwAvADEANQAwADwALwFEAC0BTQAsAVcAKg
FkACgBcQAnAYEAJQCTACQApGjALsAlgDYACIA7wAhAP0AjAMAAH0IAABqCwAAXQ4AAFIQ
AABKEQAAQxEAAD8QAAA7DwAAOQ0AADcLAQA2CQYANQYMADQFEgAyBRgAMQYdAC8GI
gAuBycALQctACsHMwAqCD0AKAhCACcITAAICVcAlwljACEJcQAgCYAAHgiSAB0lpAAbBrkAGg
XSAB0E6gAZA/kAiAcAAHYLAABmDwAAWRIAAE4TAABGFAAAPxQAAD0UAAA2EwAAMxIAAD
ERAAAwdwIAMA0HAC4MDQAtdBQAKwwaACoNIAAoDSUAJw4rACUOMgAkDjkAlg9CACAPTA
AfEFcAHRBkABsQcwAYEIMAFg+VABUOqAATDbsAEgzVABEL6wARC/gAhAoAAHEPAABiEwA
AVRUAAEsWAABCFwAAPBcAADcXAAAZFwAALxYAAC0VAAArFAAAKhMDACgTCQAnEhEAJ
RMXACQTHQAIcFMAIRQpACAULwAeFTcAHBVAABsVSgAZFVUAFxViABUVcQATFYEAERWT
AA4UpwAMFL0AChLUAAoS7AALEfoAgA0AAG4SAABeFgAAUhgAAEgZAAA/GgAAORoAADMa
AAAvGgAAKxkAAckZAAAmGAAAJRgAACMYBgAhGA0AHxkUAB4ZGgAdGSAAGxkmABoaLQA
YGjUAFxo9ABUaSAATGIMAERpgAA8abwAMGn4AChqQAakZowAHGbYABRjPAAUW6QAFFvc
AfBEAAGoVAABbGAAAThsAAEUcAAA9HAAANh0AADAdAAAsHAAAKBwAACUcAAAJHAAAIbW
AAB0dAwAbHQoAGh4RABgeFwAXHx4AFh8kABQfKwATHzIAER87AA8fRQANH1AACyBdAAkfa
wAHH30ABR+MAAMeoAABHbQAABzMAAAb5wAAGvYAeBQAAGcYAABYGwAASx0AAEIEAAA
6HwAAMx8AAC4fAAApHwAAJh8AACMfAAAFHwAAHCAAABgiAAAWlgcAFCMNABIkFAARJBsA
DyQhAA0kKAAMJC8ACiQ3AAkkQQAHEwABSRZAAmKZwABJHcAACSJAAAJnQAAIrEAACHKA
AAg5QAAHvYAdBcAAGMbAABUHgAASCAAAD8hAAA3lQAAMCIAACshAAAnlQAAIyEAACAHA
AADlgAAGSQAABUIAAASJwUADSkLAAspEgAJKRgACCKeAAcpJAAFKSsABCk0AAIpPQAAKUk
AACIVAAApZAAAKXQAACiHAAAomwAAJ7AAACXJAAAK5QAAJPQAcBsAAF8eAABRIQAARSM
AADwkAAA0JAAALiQAACKkAAAIJAAAliMAAB4kAAAaJQAAFicAABlpAAANKwQACS0JAAUuDg
ACLxUAAC8bAAAvlQAALygAAC8wAAAvOgAAL0UAAC9SAAAvYAAALnEAAC6EAAAtmQAALK
4AACvIAAAq5QAAKfUAax8AAFOiAABNJAAAQiYAADkmAAAxJwAAKyYAACcmAAAKJgAAICYA
ABsnAAAXKQAAEisAAA0tAAAJMAQABDIIAAAZDQAANBIAADUYAAA1HgAANSUAADUsAAA1
NgAANUEAADVOAAA1XQAANG0AADSBAAAZlgAAMq0AADHHAAAw5AAALvYAZiMAAFYmAA
BJKAAAPikAADYpAAAvKQAAKikAACyOAAAhKAAAHCoAABcsAAATLgAADTEAAAKzAAAENQI
AADcHAAA4CwAAOg8AADsUAAA8GgAAPCEAADwpAAA8MgAAPD0AADxKAAA7WAAAO2oAA
Dp9AAA5kwAAOKoAADfFAAA25AAANfcAYCcAAFEqAABFKwAAOywAADMsAAAtLAAAKSsAA
CMrAAAEIQAAGC8AABMxAAANNAACDcAAAM5AAAAOwEAAD0EAAA/CAAAQAwAAEIRAAB
DFgAARB0AAEQkAABELQAAQzgAAENFAABDVAAAQmUAAEJ4AABBJwAAQKcAAD/CAAA+4w
AAPPCAWiwAAEWuAABBLwAAODAAADEvAAAsLgAAJi4AAB8wAAAZMwAAEzUAAA04AAAH0
wAAAJ0AAABAAAAAQgAAAEQBAABGBAAASAgAAEoMAABLEgAATRgAAE4fAABNKAAATTIA
AE0/AABMTgAAS18AAEpzAABJigAASKMAAEe+AABG4QAARfcAVDEAAEcZAAA9MwAANjMAA
DAyAAApMgAAIjQAAB03AAAUOgAADT0AAAZAAAAAQwAAAEYAAABIAAAASwAAAE0AAABP
AAAAUQQAAFMIAABVDAAAVxMAAFkZAABZlgAAWCwAAFG4AABXRwAAVikAAFFvtAABUgwA
AU50AAFG4AABQ3AAAT/UATjYAAEM3AAA7NwAANTYAAC02AAAKOAAAHDSAAABQ/AAAMQg
AABkYAAABJAAAATAAAAE8AAABSAAAAVAAAAFCAAABZAAAAWwAAAF0DAABfBwAAAYgwA

AGQTAABmHAAAZSUAAGUxAABkQAAAY1EAAGJIAABgfAAAX5YAAF2wAABb0gAAWvEASTw
AAEA7AAA6OgAAMjoAACg9AAAEQAAAFUQAAAxJAAAFATAAAFAAAAABUAAAAVwAAAFoAAA
BdAAAAAYAAAAGIAAABkAAAAZwAAAGkAAABrAQAAbgYAAHEMAABzFAAAdB4AAHQqAABzO
AAAcUkAAHBdAABvcwAAbYwAAGynAABqxQAAaegARkAAAD8/AAA3PwAALEIAACFGAAAXS
wAADU8AAARUAAAAWAAAAFwAAABgAAAAZAAAAAGcAAABqAAAAbQAAAHAAAAABYAAAAAdQ
AAAHcAAAB6AAAAfQAAAIIEAACDCwAAhhQAAIYgAACFLgAAhD4AAIJSAAACaAAAf4EAAH2
dAAB6uAAAedoARUUAAD1EAAAxRwAAJUwAABIRAAAOVwAABF0AAABiAAAAZgAAAGsAAA
BvAAAAcgAAAHUAAAB4AAAAewAAAH4AAACBAAAAgWAAAIYAAACIAAAAIwAAAI4AAACSAg
AAIgsAAJkWAACYJAAAIzQAAJVHAACTXQAAkXYAAI+RAACNqwAAjMcAQkoAADZNAAApUgA
AHVgAABFfAAAFZgAAAGwAAABYAAAAAdQAAAHoAAAB9AAAAgAAAAIMAAACHAAAAigAAAI0
AAACPAAAAkgAAAJQAAACXAAAAmwAAAJ4AAACiAAAApgEAAKsLAACtGAAARccAAKs5AAC
pTgAaP2YAAKWAAACjmgAAobIAO1QAAC5ZAAAhYAAAFGcAAAHuAAAAAdQAAAHsAAACBAA
AAhAAAAIkAAACMAAAAJwAAAJEAAACVAAAAmAAAAJsAAACeAAAAoQAAAKMAAACnAAAAq
gAAAK4AAACyAAAAuAAAAAL0AAADECwAAxRsAAMMtAADBQgAAvlsAALt0AAC5jgAAAt6UANG
AAACZnAAAZbgAADHYAAAB+AAAAhQAAAIIsAAACQAAAAkgAAAJYAAACZAAAAmwAAAJ0AA
AChAAAApAAAAKcAAACqAAAArQAAALAAAAACzAAAAuAAAAAL0AAADDAAAAyQAAANEAAADa
AAAA4g8AAOEhAADfNgAA3E4AANlnAADVgAAA05cA/wAAP8AAAD+AAAA/gAAP8AAAD/AA
gA/wASAP8AGwD/ACYA+wAvAPcAOADzAEAA8ABHAO0ATgDqAFQA5wBZAOQAXwDhAGQA
3wBqANwAcQDaAHkA1gCCANMAjQDQAJkAzACoAMkAvgDGAOIawwD/ALgA/wCrAP8AowD/A
J0A/wCZAP8A/wAAPsAAAD3AAAA9gAAPgAAAD4AAEA9gAMAO8AGADvACEA6QAqAOIAM
wDcADsA2ABCANMASADQAE4AzgBUAMsAWQDJAF8AxgBkAMQAawDCAHIAvwB7AL0AhgC6
AJIAtwChALUAtAcYANQArwD3AKkA/wCeAP8AlwD/AJIA/wCPAP8A+QAAAPIAAADuAAAA6wAA
AOUAAADgAAAA1gAJANQAEwDRABwAzQAIAMcALQDCADUAvwA8ALsAQgC4AEgAtQBNALI
AUwCwAFgArgBeAKsAZACpAGwApwB0AKUAFgCjAlOAOQCaaKAAqWCEAMUAnQDtAJsA/wCR
AP8AigD/AIYA/wCEAP8A7wAAAOgAAADeAAAA0QAAAMgAAAC+AAAAugAEALYADgC1ABcAt
QAgAK8AKACpAC8ApgA2AKMAPACHAEIAnwBHAIJ4ATACcAFIamgBXAJkAXgCXAGUAIQBtAJ
MAdwCRAIMAJwCRAI0AowCLALkAiQDgAlgA/wCDAP8AfQD/AHoA/wB4AP8A5AAAAANcAAADC
AAAAuAAAAK4AAACoAAAAowAAAKEACQCeABIAMwAbAJJoAlgCWACKAIAAvAJEANQCPADs
AjQBBAIsARgCKAEsAiABRAIYAVwCEAF4AggBmAIEAcAB/AHsAfQCJAHsAmgB6AK4AeADOA
HcA9gBzAP8AbwD/AG0A/wBsAP8A0wAAAL4AAACuAAAApAAAAJsAAACUAAAAkAAAAIsABQ
CJAA0AiAAVAIcAHQCFAcMAggAqAH8ALwB9ADUAewA6AHoAPwB4AEUAdwBKAHUUAUABzA
FcAcgBfAHAAaQBuAHQAAbACBAGoAkQBpAKUAaAC/AGcA6gBmAP8AYgD/AGAA/wBfAP8AwA
AAAKsAAACeAAAAkwAAAloAAACDAAAAfgAAAHoAAQB4AAkAdgAQAHUAFwB1AB4AcgAkAH
AAKgBuAC8AbAA0AGoAOQBoAD4AZwBEAGUASgBjAFEAYgBZAGAAyGBeAG0AXQB6AFwAi
gBaAJ0AWQCzAFgA2gBXAP4AVgD/AFUA/wBUAP8AsQAAAJ8AAACRAAAAhAAAAHoAAAB0A
AAAbgAAAGsAAABoAAUAZwAMAGYAEwBIABkAYwAfAGEAJQBfACoAXQAvAFwAMwBaADkA
WQA+AFcARABWAEsAVABTAFMAXABRAGcAUABzAE4AgwBNAJUATACqAEsAygBLAPQASw
D/AEOA/wBKAP8ApQAAAJQAAACEAAAAeAAAAAG4AAABmAAAAAYQAAAF0AAABbAAEAwQAI
AFgADgBXABUAVgAaAFUAIABTACUUAUQAqAFAALgBOADMATQA5AEwAPwBKAEUASQBNA
EcAVwBGAGEARABuAEMAfABCAI4AQQCjAEAAvQBAAOCaAQAD/AEAA/wBAAP8AnAAAAIsAA
AB7AAAAbgAAAGMAAABbAAAAVgAAAFIAAABQAAAAATgAEAEwACgBMABEASwAWAEoAHA
BIACEARwAIAEUAKgBEAC8AQgA0AEEAOgBAAEEAPwBIAD0AUgA8AFwAOwBoADkAdwA4AI

gANwCdADYAtQA2ANkANgD5ADYA/wA2AP8AIQAAAIIAAABByAAAAZQAAAFoAAABTAAAATQA
AAEkAAABGAAAARAACAEMABwBCAA0AQQASAEAGAA/AB0APgAhADwAJgA7ACsAOgAwA
DgANgA3ADwANgBEADQATQAzAFgAMgBkADAACgAvAIMALgCXAC0ArQAtAMsALQDvAC0A/
wAtAP8AJgAAAHsAAABrAAAAAXgEAAFFQCAABMAwAARgIAAEEBAAA+AAAAPAAAADsABAA6A
AoAOQAPADgAFAA3ABkANgAeADQAIwAzACcAMgAsADAAMgAvADgALgBAAC0ASQArAFQA
KgBgACkAbgAnAH4AJgCSACUApwAIAMEAJQDkACQA+wAIAP8AiAAAAHUAAABmAwAAWAY
AAE4HAABGCAAAQAgAADsHAAA4BgAANQQAADQCAgAzAAcAMgAMADEAEQAwABYALwAb
AC0AHwAsACQAKwApACkALwAoADUAIJwA9ACYARgAkAFEAIwBdACIAagAgAHsAHwCOAB4
AogAeALkAHQDZAB0A8wAdAP8AgwAAAHADAABhBwAAVAoAAEoLAABBDAAAOWwAADYMA
AAyCwAALwoAAC0IAAAsBgQALAQJACsCDQApAhMAKAIYACcBHQAmASEAJAIInACMCLAAiAj
MAIQI7ACACRAAEAk4AHQJaABsCaAAaAngAGAGKABcBngAWALMAFgDOABYA6gAVAPwAfw
IAAGwHAABdCgAAUA0AAEYPAAA9EAAANxAAADIQAAAuDwAAK4AAACgNAAANDAIAJgoGA
CUICGAkBxAAlwcVACEHGgAgCB8AHwgkAB4IKgAdCDEAGwk5ABoJQwAYCU0AFwlaABUJaA
ATCXcAEgmJABAIInAAPB7AADgXIAA4E5AAOAvYAegUAAGgKAABZDgAATBEAAEISAAA6Ew
AAMxMAAC4TAAaqEwAAJxIAACQSAAAiEQEAIRAEACAOBwAfDQsAHg0SABwNGAAbDh0AG
g4jABgOKgAXDjEAFQ86ABMPRAASEE8ADxBcAA0QaQALEHgAChCJAAgPnAAHDrAABg3HAA
UM4gAECvMAdgKAAGQNAABVEQAASRMAAD8VAAA3FQAAMBYAACsWAAANFQAAIxUAACE
VAAAeFAAAHBQCABsTBAAZEwgAGBMOABcUFQAVFBsAFBQhABIUJwARFS8ADxU3AA0VQ
QALFUsACRVXAAcVZQAFFXQABBWGAAMUmQABE60AABLFAAAR4AAAEPMACgwAAGERA
ABSFAAARhYAADwXAAA0GAAALRgAACgYAAAKGAAAIrGAAAB4XAAAbFwAAGRcBABcXAwA
UGAUAEhkLABEZEgAPGhgADRoEAAwaJAAKGisACRozAAcaPAAFGkcAAxpTAAIaYQAAGnEA
ABmDAAAZIwAAGKwAABfEAAAW4AAAFfIAbg8AAf0TAABPFwAAQxgAADoaAAAxGgAAKxoAA
CYaAAAIgGAAHhoAABwaAAAZGQAAfXoBABQbAgAQHAQADR4JAAoeDwAIHhUABx4bAAYeI
QAEHygAAx8wAAEfOQAAH0QAAB9QAAAfXgAAHm8AAB6BAAAdlgAAHKsAABvDAAAa4QAAG
fMAahIAAFkWAABLGQAAQBsAADccAAAvHQAAKB0AACMcAAAgHAAAHrwAABobAAAYGwA
AFRwAABEEAgANIAQACSEIAAYjDQADIXIAASMYAAAKHgAAJCUAACQtAAAKNgAAJEEAACRN
AAAJXAAAI2wAACN/AAAIiAAAlaoAACDDAAAf4QAAHvQAZhYAAFYaaABIHAAAPR4AADQfAAA
sHwAAJh8AACIeAAAEHgAAGx0AABkdAAAVHgAAEiAAAA0iAQAKJAQABiYHAAEnCwAAKBAA
ACkVAAApGwAAKSIAACkpAAApMgAAKT0AACIKAAApWAAAKGkAACh8AAAnkgAAJqkAACXD
AAAK4gAAI/UAYRoAAFEdAABEHwAAOIeAADEhAAApIQAAJCEAACAgAAAdIAAGh8AABYhA
AASlgAADSQAAAKmAAAFKAIAASoGAAAsCQAALg0AAC8SAAAwGAAAMB8AADAmAAAwLwA
AMDkAAC9GAAAvVQAALmYAAC55AAAtkAAALKcAACvCAAaQ4wAAKPcAXB4AAE0hAABBIwA
ANiQAAC4kAAANJAAAIyMAAB8iAAAcIlgAAfYMAABIIAAANJwAACSkAAAUAAAAALgEAADAEA
AAyBwAANAoAADYOAAA3FAAANxsAADciAAA3KwAANzUAADdCAA2UQAANmIAADV1AAA0j
AAAM6UAADHAAAAw4wAAL/gAVyIAAEglAAA9JgAAMycAACsnAAAmJgAALIUAAB4IAAAYJgAA
EygAAA4qAAAJLQAABDAAAAAaYAAAAANAADcBAAA5BAAAOwcAAD0LAAA/EQAAQBcAAEA
eAAA/JwAAPzEAAD89AAA+TAAAPV0AAD1xAAA8iAAAOqEAADm9AAA44QAAN/kAUScAAEQp
AAA5KgAAMCoAACopAAAIKAAAIcGAABoqAAAUlAAADi4AAAGxAAADNAAAADcAAAA5AAAA
PAAAAD4AAABAAQAAQgQAAEQIAABHDAAASRIAAEoaAABJlgAASSwAAEg4AABIRwAAR1gA
AEVrAABEGwAAQ50AAEK4AABA3QAAP/cASyWAAD8tAAA2LQAALI0AACkrAAAJKwAAHC0AAB
UwAAAOMwAACDYAAAE5AAAAAPAAAAD8AAABCAAAAARAAAAEcAAABJAAAAswAAAE0EAAB
QCAAAUg0AAFUUAABVHQAAVCYAAFQyAABTQAAAUIEAAFFkAABPewAATpYAAEyxAABK1

QAASfQARjEAADsxAAAzMQAALS8AACcvAAAeMgAAFjUAAA44AAHPAAAAD8AAABDAAAAR
gAAAEkAAABMAAAATwAAAFEAAABUAAAAVgAAAFkAAABbAgAAXgcAAGENAABjFQAAYx8A
AGlqAABhOAAAYEkAAAF5dAABcdAAAWo4AAFiqAABXywAAVe8AQTYAADg1AAAYNAAAKzQA
ACE2AAAYOgAAED4AAAdCAAAARgAAAEoAAABOAAAAUQAAAFUAAABYAAAAWwAAAF0AA
ABgAAAAAYgAAAGUAAABoAAAAawEAAG4HAABxDgAAcxgAAHljAABxMAAAb0EAAG5UAABra
gAAaYQAAGegAABlvgAAY+QAPjoAADg5AAAwoQA AJTsAABtAAAAARRAAAB0oAAABOAAAAU
wAAAFcAAABbAAAAXgAAAGIAAABIAAAAaAAAAAGsAAABuAAAAcAAAAHMAAAB2AAAAeQAA
AHwAAACABgAAgw8AAIqBAAACDKAAAgTgAAH9LAAB9YQAAe3kAAHmUAAB3sAAAddEAPj4A
ADY+AAAqQQA AHkYAABNLAAAIUQAAAFcAAABcAAAAAYQAAAGUAAABpAAAAbQAAAHEAAA
B1AAAAeAAAAHsAAAB+AAAAgAAAAIMAAACGAAAAiQAAAI0AAACRAAAAIQUAAJkQAACYH
QAAliwAAJQ/AACSVQAAkG0AAI2HAACKowAAiL8AO0QAAC9HAAAiTAAAFIIAAApZAAAAXwA
AAGYAAABrAAAAcAAAAHUAAAB5AAAAfAAAAIAAAACEAAAAHwAAAIoAAACNAAAAkAAAAJ
MAAACWAAAAmQAAAJ0AAACHAAAApQAAAKoGAACuEgAArCIAAKo0AACoSQAApWEAAKJ6
AACglQAAAnq4ANE0AACdSAAAaWQAADWEAAABoAAAAcAAAAHYAAAB7AAAAgAAAAIQAAA
CIAAAAIwAAAI8AAACTAAAAIgAAAJkAAACcAAAAoAAAAKMAAACmAAAAqgAAAK4AAACzAA
AAuQAAAMAAAADIBAAAyxQAAMglAADEOQAAwVEAAL1pAAC6gwAAAt5sALFkAAB9gAAASaA
AABXAAAAAB5AAAAgAAAAIYAAACLA AAAkAAAAJQAAACXAAAAmgAAAJ0AAACHAAAApAAAA
KcAAACqAAAArgAAALEAAAC1AAAAugAAAL8AAADFAAAAazQAAANYAAADfAAAA6AcAAOcaA
ADkLgAA4UUAAN1eAADYdwAA1I8A/wAAPsAAAD4AAAA+AAAApKAAAD9AAYA/wAQAP8AG
QD/ACMA/QAtAPkANQD1AD0A8gBFAO8ASwDqAFE5gBXAOMAXQDfAGMA3ABpANgAcADV
AHcA0QCAAM4AjADLAJkAxwCpAMUAwQDDAOcAwQD/AK8A/wCjAP8AmgD/AJUA/wCRAP8A+
gAAPQAAADwAAAA7wAAAPEAAAD1AAAA8wAKAPAAFQDvAB4A6gAnAOAMADXDgA0gA
/AM4ARgDLAEwAyABRAMYAVwDEAFwAwQBiAL8AaQC9AHEAugB6ALgAhQC1AJIASgCiAK8A
tQCtANoAqwD8AJ8A/wCWAP8AjwD/AIoA/wCGAP8A8QAAAOoAAADmAAAA5QAAAOAAAADb
AAAA1AAAFANIAEADMABkAxwAiAMEAKgC9ADIAuQA5ALYAPwCzAEUAsQBLAK8AUACtAFYA
qwBcAKkAYwCnAGoApQBzAKMAfQCgAlOAngCaAJwArACZAMgAmADzAJEA/wCHAP8AgQD/A
H0A/wB7AP8A5gAAAN0AAADYAAAAzwAAAMIAAAC6AAAAtwAAALAACwCvABQArwAcAKsAJ
ACmACwAowAyAKEAOACfAD4AnQBEAJsASQCZAE8AlwBVAJUAWwCUAGMAkgBrAl8AdQCN
AllIiwCRAIkAowCGALsAhQDIAIIA/wB6AP8AdAD/AHAA/wBuAP8A2AAAAMwAAAC/AAAAsQAA
AKkAAACjAAAAAnAAAAJoABQCZAA4AmAXAJcAHgCTACUAKAAsAl0AMgCKADcAiAA9AIYAQ
gCEAEgAggBOAIAAVAB/AFsAfQBkAHsAbgB5AHoAdwCJAHUAmgB0ALAAcgcDRAHIA+wBsAP8
AZwD/AGQA/wBjAP8AyAAAALYAAACqAAAAAnQAAAJUAAACNAAAAiAAAAIUAAQCEAAkAgwA
RAIIAGQB/ACAAfAAmAHkALAB3ADEAdQA2AHQAOWByAEEAcQBHAG8ATQBtAFQAawBcAGo
AZgBoAHIAZgCAAGQAKQBiAKYAYQDCAGEA7wBeAP8AWwD/AFgA/wBXAP8AtQAAAKUAAA
CWAAAAIwAAAAIMAAAB7AAAAAdgAAAHMAAABxAAQAbwAMAG8AEwBuABoAbAAgAGkAJgBnA
CsAZQAwAGMANQBIAADoAYABAAF8ARgBdAE0AWwBVAFoAXwBYAGoAVgB4AFUAIQBT AJ0A
UgC2AFE4ABRAP8ATwD/AE0A/wBNAP8AqQAAAJgAAACIAAAAfAAAAHEAAABrAAAAZgAAA
GMAAABgAAAAAXwAHAF4ADgBeABUAXQAbAFoAIABYACUAVwAqAFUALwBUADQAUGA5AFE
APwBPAECATgBPAEWAWABLAGMASQBxAEgAgQBGAJUARQCsAEUAzgcBEAPgARAD/AEMA/
wBDAP8AngAAAI sAAAB8AAAAbwAAAGQAAABdAAAAWAAAAFUAAABTAAAAUQADAF AACQ
BQABAAUAAWAE4AGwBMACAASgAIAEkAKgBHAC8ARgA0AEQAOGBD AEEAQgBJAEAAUGA/
AF0APgBqADwAegA7AI0AOgCjADoAwAA5AO0AOQD/ADkA/wA6AP8AIAAAAIIAAABYAAAAZAA

AAFoAAABSAAAATAAAAEkAAABHAAAAARQAAAEUABgBEAAAsARAARAEMAFwBBABwAPwAh
AD4AJQA9ACoAOwAvADoANQA5ADwANwBEADYATQA1AFgAMwBIADIAdAAxAlYAMACcADA
AtQAvAN8ALwD/ADAA/wAwAP8AjAAAAHkAAABqAAAAAXAAAFIAAABKAAAAARAAAAEAAAAA
9AAAAPAAAADsAAwA6AAgAOQANADkAEwA4ABgANgAdADUAIQAzACYAMgArADEEMAAvA
DcALgA/AC0ASAArAFMAKgbgACkAbwAoAIAAJwCVACcArQAmAM8AJgD3ACcA/wAoAP8AhAA
AAHIAABiAAAAVQAAAEsAAABDAAAAPQAAADgAAAA1AAAMwAAADIAAQAxAAUAMAAKA
DAADwAvABQALgAZACwAHQArACIAKgAnACkALQAoADMAJgA7ACUARAakAE8AlwBbACIAa
gAhAHsAIACPAB8ApgAfAMQAHwDsAB8A/wAfAP8AfgAAAGwAAABcAAAAUAIAAEYDAAA9BA
AANwQAADIEAAAuAwAALAIAACoAAAApAAMAKAAHACgADAAnABEAJgAVACUAGgAkAB8AI
wAkACIAKQAhADAIAA3AB8AQAAAdAEsAHABXABsAZgAaAHYAGQCKABgAoAAYALsAGADh
ABgA/AAYAP8AeQAAAGcAAABYAwAASwYAAEEHAAA5CAAAMggAAC0IAAApBwAAJgcAACQ
GAAAJBAIAIgIFACEBCQAhAA0AIAASAB8AFwAeABwAHQAhABwAJgAbAC0AGgA0ABkAPQAY
AEgAFgBUABUAYgAUAHMAEwCGABIANAAARALMAEQDTABEA8gARAP8AdQAAAGMCAABUB
wAARwkAAD0LAAA1CwAALgsAACKLAAAIcwAAIgsAAB8KAAAdCQEAHAgEABwHBwAbBQsAG
gUQABkFFAAYBRkAFwUeABYEJAAVBCoAFAQxABMEOGASBEUAEQRSAA8DYAAOA28ADA
OCAAsClwAKAK0ACgDHAAoA6AAKAPwAcAEAAF8GAABQCgAARAwAADoNAAAYDgAAKw4A
ACYOAAAhDgAAHg4AABwNAAAZDQEAGAwEABcMBgAWCwkAFQoMABQKEgATChYAEgocA
BEJIQAQCSgADwowAA0KOQAMCkQACgpQAAgKXgAGCm4ABQqAAAMKkwADCKgAAgbAAA
MF3gADAvUAbAQAAFsjAABNDAAAQQ4AADcQAAAuEQAAKBEAACMRAAAfEQAAAGxEAAABkR
AAWEAEAFRAEABMQBgASDwgAEA4KAA4ODgANDxQADA8aAAsPIAAKECYACRAuAAcQN
wAGEEEABBB0AAMQXAABEGwAAA9+AAAOkgAADqcAAA2+AAAL2wAACvAAaAcAAFcMAAB
JDwAAPhEADQTAAAsEwAAJRQAACAUAACeAwAAGRMAABYTAAAUeGlAEHIFABESBwAOE
ggADBmJAAoUDAAIFBIABxQXAAYUHQAFFCMABBQrAAIVNAABFT4AABVLAAAUWQAAFGkA
ABR8AAATkQAAEQcAABG/AAAQ3QAADvIAZQoAAAFQPAABGEgAAOxQAADEVAAApFgAAIxYA
AB4WAAAaFQAAFxUAABUVAAATFAMAERQGAA8UBwAMFQcACRYIAAYYCwADGQ8AAhkVA
AEZGgAAGSEAABkoAAAZMQAAGTsAABIIAAAZVwAAGWcAABh6AAAXkAAAF6cAABa/AAAV3
gAAE/QAYQ0AAFASAABDFQAAOBcAAC4XAAAmGAAAIRgAABwYAAAYFwAAFhcAABQWAgA
SFQUAEByFAAwXBQAJGAUABhoHAAlcCQAAHgwAAB4SAAAEGAAAHx4AAB4IAAAeLgAAHjk
AAB5FAAAeVAAAHWUAAB14AAAcjgAAG6YAABrAAAAZ4QAAGPYAXBAAE0VAABAFwAANR
kAACsaAAAKGgAAHxoAABoZAAAXGQAAFRgBABMXAwAQGAIDBkCAAKbAgAGHAMAAh4FA
AAgBwAAIgoAACQPAAAKFQAAJBsAACQjAAAKKwAAJDUAACRCAAAjUQAAI2IAACJ2AAAhjAA
AIKUAAAB/AAAAe4gAAHfgAWBUAAEKYAAA8GgAAMRwAACgcAAAIHAAAHrWAAABkbAAAXGgA
AFBkBABEaAAANGwAACR0AAAUFaAACIQEAACMDAAAmBQAAKAgAACoMAAArEgAAKxgAA
CsgAAArKAAAKjIAACo/AAApTgAAKV8AAChzAAAniQAAJqMAACW/AAAK4wAAI/oAUxkAAEQcA
AA4HgAALh8AACyFAAAgHgAAHB0AABKcAAAWHAAAEh0AAA0eAAAJIAAABSIAAAEIAAAAJw
AAACkBAAAsAwAALgYAADAJAAAZDgAAMxUAADMCAAAZJAAAMi4AADi7AAAXSgAAMFsAAC
9uAAAUhQAALaAAACy8AAAq4wAAKfsATR0AAEAAGAAA0IQAAKyIAACQhAAAFIAAAHB8AABge
AAATIAAADSIAAAKKAAAEJgAAACKAAAArAAAAALQAAADAAAAAAZAAAAANQMAADgGAAA6CwA
APBEAADwYAAA7IQAAOyoAADs2AAA6RAAAOVYAADhqAAA2gQAANZwAADS4AAAY4AAAMf
sASCIADskAAAXJQAAKCAACIKAAAFIgAAGiEAABQjAAAOJQAACSgAAAMrAAAAALQAAADA
AAAAZAAAAANGAADgAAAA7AAAAPQAAAEACAABDBgAARgwAAEcTAABGGwAARiUAAEUwA
ABEPwAAQ1AAAEfKAABAewAAPpYAAD2ZAAA72wAAOfkAQiYAADcoAAAtKAAAJycAACIIAA

dJQAAFiAAA8pAAAJLAAAAjAAAAAzAAAAANwAAADkAAAA8AAAAAPwAAAEIAAAABEAAAAARwA
AAEkAAABMAGaATwcAAFINAABSFgAAUiiAAAFerAABQOQAAT0oAAE1dAABLdAAASY8AAEe
sAABF0QAARPUAPSSAADMSAAARkWAJyKAACEpAAAYKwAAES4AAAKyAAACNgAAADoAAA
A9AAAAQAAAAEMAAABGAAAAASQAAAEwAAABPAAAAUgAAAFQAAABXAAAAWglAAAF4IAABh
EAAAYBkAAAF8kAABeMgAAxEIAAFtWAABYbAAAVoYAAFSjAABSwwAAUOWAOTAAADEwAAA
rLgAAJS0AABswAAASNAAACTgAAAE8AAAAQAAAAEQAAABIAAAATAAAAE8AAABSAAAAVQ
AAAFgAAABbAAAAAXgAAAGEAAABkAAAAZwAAAGsDAABuCQAAcRMAAHAEaABvKgAAbjkAA
GxMAABpYgAAZnwAAGSYAABhtgAAX+AAANjQAADeZAAApMgAAHzUAABU6AAAKPgAAUUMA
AABIAAAATQAAAFEAAABWAAAAWgAAAF0AAABhAAAAZAAAAAGgAAABrAAAAbgAAAHAAAA
B0AAAAAdwAAAHsAAAB/AQAAgwkAAIQUAACDIQAAGTAAAH9DAAB8WAAAEeXEAHhMAABZq
QAACcoANjgAAC44AAAJOWAAGD8AAAXFAAABTAAAAFEAAABWAAAAXAAAAAGAAAAABIAAAA
aQAAAG0AAABxAAAAAdAAAAHcAAAB6AAAAfQAAAIAAAAACEAAAAHwAAAIIsAAACPAAAAIAAA
AJgKAACYFwAAAliYAAAJM4AACQTgAAjWYAAIqAAACImwAAhbYAND0AACHBAAAbrgAAEEwA
AANTAAAAWgAAAGAAAAABMAAAAwAAAHAAAAAB0AAAAeQAAAH0AAACBAAAAhQAAAIgAA
ACLAAAAjwAAAJIAAACVAAAAmQAAAJ0AAACiAAAApwAAAKwAAACyCwAAArxsAAKwsAACpQ
QAAPVgAAKJxAACfjAAANkYALUcAACBMAAATUwAABIsAAABiAAAAagAAAHAAAAAB3AAAAfA
AAAEIAAACFAAAAiQAAAI0AAACRAAAAIQAAAJkAAACcAAAAAnwAAAKIAAACmAAAAqgAAAK4
AAACzAAAAuQAAAMEAAADJAQAAZRAAAMkhAADFNQAawEwAALxIAAC4fgAAtJcAJVMAABh
aAAALYgAAAGsAAABzAAAAewAAAIIAAACCHAAAAjAAAAJAAACUAAAAmAAAAJ0AAACHAAA
ApQAAAKkAAACsAAAAsAAAAALMAAAC4AAAAvQAAAMQAAADMAAAA1AAAN8AADoAAAA
8AAAAO8SAADsJgAA6DsAAONUAADfbAAA2oQAAAAAAAAAAAAAAAAAAAAAAAAAEDBAUGCA
kKcW0ODxESExQWFxgaGxwdHyAhliQIJigpKistLi8wMjM0Njc4OTs8PT5AQUJERUZHSUPLTU5P
UFJTUVFVXWFlbXF1eYGFiy2VmZ2lqa2xub3Bxc3R1d3h5enx9foCBgoOFhoeliouMjo+QkZOUIZaY
mZqcnZ6foaKjpKanqKqrrK2vsLGztLW2uLm6u72+v8HCw8TGx8jJy8zNz9DR0tTV1tfZ2tvd3t/g4uP
k5ufo6evs7e7w8fL09fb3+fr7/P7////////////////////////////////////wAAAAAAAAAAAAAAAAAAAA
ABAwQFBggJCgsNDg8REhMUFhcYGhscHR8glSlkJSYOKSorLS4vMDIzNDY3ODk7PD0+QEFC
REVGR0IKS01OT1BSU1RVV1hZW1xdXmBhYmNiZmdpamtsbm9wcXN0dXd4eXp8fX6AgYKDhY
aHilqLjI6PkJGTIJWWmJmanJ2en6Gio6Smp6iqq6ytr7Cxs7S1tri5uru9vr/BwsPEXsflycvMzc/Q0dLU
1dbX2drb3d7f4OLj5Obn6Onr7O3u8PHy9PX29/n6+/z+////////////////////////////////////8AAAAA
AAAAAAAAAAAAAAAAAAAAQMEBQYICQoLDQ4PERITFBYXGBobHB0fICEiJCUmKCkqKy0uLzAyM
zQ2Nzg5Ozw9PkBBQkRFRkdJSktNTk9QUINUVVdYWVtcXV5gYWJjZWZnaWprbG5vcHFzdHV3
eHl6fH1+glGCg4WGh4iKi4yOj5CRk5SVlpiZmpydnP+hoqOkpqeoqqusra+wsbO0tba4ubq7vb6/wcL
DxMbHyMnLzM3P0NHS1NXW19na293e3+Di4+Tm5+jp6+zt7vDx8vT19vf5+vv8/v////////////////////////////////////
////////////////////////////////////AAECAwQFBgcICQoLDA0ODxAREhMUFRYXGBkaGxwdHh8glSljJCUmJygpKi
ssLS4vMDEyMzQ1Njc4OT07PD0+P0BBQkNERUZHSEIKS0xNTk9QUVJTUVFVWV1hZWltcXV5fY
GFiy2RIZmdoaWprbG1ub3BxcnN0dXZ3eHl6e3x9fn+AgYKDhIWGh4iJiouMjY6PkJGSK5SVlpeYm
ZqbnJ2en6ChoqOkpaanqKmqq6ytrq+wsbKztLW2t7i5uru8vb6/wMHCw8TFXsflycrLzM3Oz9DR0tP
U1dbX2Nna29zd3t/g4eLj5OXm5+jp6uvs7e7v8PHy8/T19vf4+fr7/P3+/21mdEAAAAAAwQhAAAB
AAAAAAAAAAAAAAAAAAAAQAAAAAAAAAAAAAAAAAAAAAAEAAAABAgMEBQYHCAkKCwwN
Dg8QERITFBWUFXgZGhscHR4fICEilyQIJicoKSorLC0uLzAxMjM0NTY3ODk6Ozw9Pj9AQUJDRE
VGR0hJSktMTU5PUFFSU1RVVldYWVpbXF1eX2BhYmNkZWZnaGlqa2xtbm9wcXJzdHV2d3h5e
nt8fX5/glGCg4SFhoeliYqLjI2Oj5CRkpOUiZaXmJmam5ydnp+goaKjpKWmp6ipqqusra6vsLGys7S1

tre4ubq7vL2+v8DBwsPExcbHyMnKy8zNzs/Q0dLT1NXW19jZ2tvc3d7f4OHi4+TI5ufo6err7O3u7/D
x8vP09fb3+Pn6+/z9/v8AAQECAGMDBAQFBgYHBwgICQkKCwsMDA0NDg8PEBARERITExQUF
RYWFxcYGRkaGhschB0eHh8glCEiliMkJCUmJicoKSkqKywtLS4vMDEyMjM0NTY3ODk6Ozw9Pj
9AQkNERUZISUpMTU9QUINVV1haXF5gYmRmaGptb3F0dnl8foGDhomLjpCSiZeZm52foaOlP6iq
rK2vsLKztba3ubq7vL2/wMHCw8TFxsflycrLzM3Nzs/Q0dLS09TV1tbX2NnZ2tvb3N3d3t/f4OHh4uPj
5OXI5ubn6Ojp6err6+zs7e7u7+/w8PHy8vPz9PT19vb39/j4+fn6+/v8/P39/v7/AAEBAGIDAwQEBQY
GBwclCAkJCgsLDAwNDQ4PDxAQERESExMUFBUWFhcXGBkZGhobHBwdHh4fICAhliljJCQIJiY
nKCKpKissLS0uLzAxMjlzNDU2Nzg5Ojs8PT4/QEJDREVGSEIKTE1PUFJTVVdYWIxeYGGJkZmhqb
W9xdHZ5fH6Bg4aJi46QkpWXmZudn6Gjpaeoqqytr7Cys7W2t7m6u7y9v8DBwsPExcbHyMnKy8zN
zc7P0NHS0tPU1dbW19jZ2drb29zd3d7f3+Dh4eLj4+TI5ebm5+jo6enq6+vs7O3u7u/v8PDx8vLz8/T
09fb29/f4+Pn5+vv7/Pz9/f7+//6wGAD9sicD/rM0DP+yPhv/skgt/71PP/vEVIX4yF1s9MljgvHKaJfpym
6p4sVyt9y+dsPVuHvP0LGA2smrh+fBp5n4waeZ+MGnmfjBp5n4waeZ+MGnmfjBp5n4waeZ+MGn
mfjBp5n4waeZ+MGnmfjBp5n4waeZ+MGnmfjBp5n4waeZ+P6wGAD9sicD/rM0DP+yPhv/skgt/71P
P/vEVIX4yF1s9MljgvHKaJfpym6p4sVyt9y+dsPVuHvP0LGA2smrh+fBp5n4waeZ+MGnmfjBp5n4w
aeZ+MGnmfjBp5n4waeZ+MGnmfjBp5n4waeZ+MGnmfjBp5n4waeZ+MGnmfjBp5n4waeZ+P6wGA
D9sicD/rM0DP+yPhv/skgt/71PP/vEVIX4yF1s9MljgvHKaJfpym6p4sVyt9y+dsPVuHvP0LGA2smrh+
fBp5n4waeZ+MGnmfjBp5n4waeZ+MGnmfjBp5n4waeZ+MGnmfjBp5n4waeZ+MGnmfjBp5n4waeZ
+MGnmfjBp5n4waeZ+P6wGAD9sicD/rM0DP+yPhv/skgt/71PP/vEVIX4yF1s9MljgvHKaJfpym6p4s
Vyt9y+dsPVuHvP0LGA2smrh+fBp5n4waeZ+MGnmfjBp5n4waeZ+MGnmfjBp5n4waeZ+MGnmfjB
p5n4waeZ+MGnmfjBp5n4waeZ+MGnmfjBp5n4waeZ+P6wGAD9sicD/rM0DP+yPhv/skgt/71PP/vE
VIX4yF1s9MljgvHKaJfpym6p4sVyt9y+dsPVuHvP0LGA2smrh+fBp5n4waeZ+MGnmfjBp5n4waeZ+
MGnmfjBp5n4waeZ+MGnmfjBp5n4waeZ+MGnmfjBp5n4waeZ+MGnmfjBp5n4waeZ+P6wGAD9sic
D/rM0DP+yPhv/skgt/71PP/vEVIX4yF1s9MljgvHKaJfpym6p4sVyt9y+dsPVuHvP0LGA2smrh+fBp5
n4waeZ+MGnmfjBp5n4waeZ+MGnmfjBp5n4waeZ+MGnmfjBp5n4waeZ+MGnmfjBp5n4waeZ+MG
nmfjBp5n4waeZ+P6wGAD9sicD/rM0DP+yPhv/skgt/71PP/vEVIX4yF1s9MljgvHKaJfpym6p4sVyt9y
+dsPVuHvP0LGA2smrh+fBp5n4waeZ+MGnmfjBp5n4waeZ+MGnmfjBp5n4waeZ+MGnmfjBp5n4w
aeZ+MGnmfjBp5n4waeZ+MGnmfjBp5n4waeZ+P6wGAD9sicD/rM0DP+yPhv/skgt/71PP/vEVIX4yF
1s9MljgvHKaJfpym6p4sVyt9y+dsPVuHvP0LGA2smrh+fBp5n4waeZ+MGnmfjBp5n4waeZ+MGnmf
jBp5n4waeZ+MGnmfjBp5n4waeZ+MGnmfjBp5n4waeZ+MGnmfjBp5n4waeZ+P6wGAD9sicD/rM0
DP+yPhv/skgt/71PP/vEVIX4yF1s9MljgvHKaJfpym6p4sVyt9y+dsPVuHvP0LGA2smrh+fBp5n4wae
Z+MGnmfjBp5n4waeZ+MGnmfjBp5n4waeZ+MGnmfjBp5n4waeZ+MGnmfjBp5n4waeZ+MGnmfjB
p5n4waeZ+P6wGAD9sicD/rM0DP+yPhv/skgt/71PP/vEVIX4yF1s9MljgvHKaJfpym6p4sVyt9y+dsP
VuHvP0LGA2smrh+fBp5n4waeZ+MGnmfjBp5n4waeZ+MGnmfjBp5n4waeZ+MGnmfjBp5n4waeZ+
MGnmfjBp5n4waeZ+MGnmfjBp5n4waeZ+P6wGAD9sicD/rM0DP+yPhv/skgt/71PP/vEVIX4yF1s9
MljgvHKaJfpym6p4sVyt9y+dsPVuHvP0LGA2smrh+fBp5n4waeZ+MGnmfjBp5n4waeZ+MGnmfjBp
5n4waeZ+MGnmfjBp5n4waeZ+MGnmfjBp5n4waeZ+MGnmfjBp5n4waeZ+P6wGAD9sicD/rM0DP+
yPhv/skgt/71PP/vEVIX4yF1s9MljgvHKaJfpym6p4sVyt9y+dsPVuHvP0LGA2smrh+fBp5n4waeZ+M
GnmfjBp5n4waeZ+MGnmfjBp5n4waeZ+MGnmfjBp5n4waeZ+MGnmfjBp5n4waeZ+MGnmfjBp5n4
waeZ+P6wGAD9sicD/rM0DP+yPhv/skgt/71PP/vEVIX4yF1s9MljgvHKaJfpym6p4sVyt9y+dsPVuHv
P0LGA2smrh+fBp5n4waeZ+MGnmfjBp5n4waeZ+MGnmfjBp5n4waeZ+MGnmfjBp5n4waeZ+MGn
mfjBp5n4waeZ+MGnmfjBp5n4waeZ+P6wGAD9sicD/rM0DP+yPhv/skgt/71PP/vEVIX4yF1s9Mljgv
HKaJfpym6p4sVyt9y+dsPVuHvP0LGA2smrh+fBp5n4waeZ+MGnmfjBp5n4waeZ+MGnmfjBp5n4w

aeZ+MGnmfjBp5n4waeZ+MGnmfjBp5n4waeZ+MGnmfjBp5n4waeZ+P6wGAD9sicD/rM0DP+yPhv
/skgt/71PP/vEVIX4yF1s9MljgvHKAJfpym6p4sVyt9y+dsPVuHvP0LGA2smrh+fBp5n4waeZ+MGnmf
jBp5n4waeZ+MGnmfjBp5n4waeZ+MGnmfjBp5n4waeZ+MGnmfjBp5n4waeZ+MGnmfjBp5n4waeZ
+P6wGAD9sicD/rM0DP+yPhv/skgt/71PP/vEVIX4yF1s9MljgvHKAJfpym6p4sVyt9y+dsPVuHvP0LG
A2smrh+fBp5n4waeZ+MGnmfjBp5n4waeZ+MGnmfjBp5n4waeZ+MGnmfjBp5n4waeZ+MGnmfjBp
5n4waeZ+MGnmfjBp5n4waeZ+P6wGAD9sicD/rM0DP+yPhv/skgt/71PP/vEVIX4yF1s9MljgvHKAJf
pym6p4sVyt9y+dsPVuHvP0LGA2smrh+fBp5n4waeZ+MGnmfjBp5n4waeZ+MGnmfjBp5n4waeZ+
MGnmfjBp5n4waeZ+MGnmfjBp5n4waeZ+MGnmfjBp5n4waeZ+P6wGAD9sycD/rQ0DP+yPhv/tEcs
/b9OPvjIVVTzzVxs79Fhg+rUZZjl1mmq3tBtudnLcMbRxXTRyr943L+5f+essor0rLKK9KyyivSssor0r
LKK9KyyivSssor0rLKK9KyyivSssor0rLKK9KyyivSssor0rLKK9KyyivSssor0rLKK9P6xFwD8sycD/b
Q0DP+zPhv/tkcs+sJOPfXNVFPv01ps6dpeg+TfYZnc2WSr09NpuczMbMTEx3DPu8F02K+8eOOdt
X/unbV/7p21f+6dtX/unbV/7p21f+6dtX/unbV/7p21f+6dtX/unbV/7p21f+6dtX/unbV/7p21f+6dtX/unbV
/7v2xFwD8tCcD/bU0DP+0Phr/uEcr98VNPfHSU1Pq21hr5ONbg9zjX5nS3GOqytRnt8HOacK4yWz
Lr8Nw1aO+c96SuXnokrl56JK5eeiSuXnokrl56JK5eeiSuXnokrl56JK5eeiSuXnokrl56JK5eeiSuXnok
rl56JK5eeiSuXnokrl56P2yFgD7tCcD/LU0DP+0Phr8u0Yq88INPO3ZUVLI41Zr4OtYg9TIXpjK3Wko
wNZltLfQZ7+uy2nlpcZs0ZnBb9mKvHXiirx14oq8deKKvHXiirx14oq8deKKvHXiirx14oq8deKKvHXiir
x14oq8deKKvHXiirx14oq8deKKvHXiirx14vyyFgD7tSYD+7YzC/61Phr5vkYp781MOujfUFHg6lRq2+
9Xgs3nXZbC32CluNhisa/SZLumzWbEnchpzJHEbdSEwHLbhMBy24TactuEwHLbhMBy24TactuE
wHLbhMBy24TactuEwHLbhMBy24TactuEwHLbhMBy24TactuEwHLbhMBy2/yzFQD6tiYC+rczC/2
2Phn1wUYn6tNMOeHIT0/c8FNp0/FWgMXpW5S64F6jsNpgrqfUYbedz2O/IMtmx4rHas9/w3DVf8N
w1X/DcNV/w3DVf8Nw1X/DcNV/w3DVf8Nw1X/DcNV/w3DVf8Nw1X/DcNV/w3DVf8Nw1X/DcNV/w
3DVf8Nw1fuzFQD5tyYC+bgzCvy4PhnxxkUI5NpLntvqT0/U81JpzPRVf77qWpGy4lyfqNxeqp/WX7
OW0mC6jc5jwoPKZ8h6x23OesdtznrHbc56x23OesdtznrHbc56x23OesdtznrHbc56x23OesdtznrHb
c56x23OesdtznrHbc56x23Oesdtzvq0FAD4uCUC+LkzCvu5PhjszEUj3eFKNNXuTk/N9lFoxPZUfbb
sWI6q5FqboN5cppfaXa6O1V61htJhu33OZcF1y2vGdctrxnXLa8Z1y2vGdctrxnXLa8Z1y2vGdctrxn
XLa8Z1y2vGdctrxnXLa8Z1y2vGdctrxm1EwD3uSUC97szCfa+PRbl00Qg1+dK
NM7zTk/F+1BmvPdSeq7uVoqj51iXmOFaoY/dW6iH2l2uf9ZftHfTY7lw0Wm9cNFpvXDRab1w0Wm
9cNFpvXDRab1w0Wm9cNFpvXDRab1w0Wm9cNFpvXDRab1w0Wm9cNFpvXDRab1w0Wm9cNF
pvfi3EQD1uyQB9b0yCe/FPBPa3kEc0O1KNcb5TU68/05ks/IPdqbxU4Sa6laQkOVYmofhWqF/3lum
eNteq3HZYa9r1meza9Zns2vWZ7Nr1meza9Zns2vWZ7Nr1meza9Zns2vWZ7Nr1meza9Zns2vWZ7
Nr1meza9Zns2vWZ7Nr1meza9Zns/W5DgDzvSIB8sAxCOTROQ7S50lfx/VJNr3/S02y/0xgqP1NcJ
z1UX6R71SJh+pWkX/nWZd45FuccuFdoGzfYKRm3mSnZt5kp2beZKdm3mSnZt5kp2beZKdm3m
SnZt5kp2beZKdm3mSnZt5kp2beZKdm3mSnZt5kp2beZKdm3mSnZt5kp+y8CwDvwSAA7cQwBtb
gNQzJ8Elhvv5GNrL/R0mn/0hanf9KaZL6TnWH9FJ/fvBVhnjtWlty61qPbOldk2jnYJZj5mSYY+Zkm
GPmZJhj5mSYY+ZkmGPmZJhj5mSYY+ZkmGPmZJhj5mSYY+ZkmGPmZJhj5mSYY+ZkmGPmZ
Jhj5mSYY+ZkmNnABgDqxhsA3tQmAsvsOBC/+0Ais/9CNkf/Q0Wb/0VTkf9HYYf/TGt9+1BzdvdUe
XD1V31s81mBaPJcg2TwX4Zg72OIYO9jiGDvY4hg72OIYO9jiGDvY4hg72OIYO9jiGDvY4hg72OIY
O9jiGDvY4hg72OIYO9jiGDvY4hg72OIYO9jiNbDBgDjzhMAz+YnBMH4OBK0/zwip/88MZr/Pj+P/0F
MhP9FV3z/SGB0/05nbv9Sa2r9VW9m/FhxY/tbc2D6XnVd+WJ3Xflid135Yndd+WJ3Xflid135Yndd+
WJ3Xflid135Yndd+WJ3Xflid135Yndd+WJ3Xflid135Yndd+WJ3Xflid9PHBgDT2wkAw/QsB7X/NBSO
/zUgm/83LI7/OjiD/z5Def9CTXH/RIRr/0pZZv9PXWP/U19g/1ZiXv9ZZFz/XWVa/2BnWv9gZ1r/YGda
/2BnWv9gZ1r/YGda/2BnWv9gZ1r/YGda/2BnWv9gZ1r/YGda/2BnWv9gZ1r/YGda/2BnWv9gZ8/LB

wDF8BkBt/8rCar/LhOc/y4djv8xKIP/NjJ4/zs9b/9ARmj/RExi/0lRX/9OVVz/U1da/1ZZWP9aW1f/XVx
V/2FeVf9hXIX/YV5V/2FeVf9hXIX/YV5V/2FeVf9hXIX/YV5V/2FeVf9hXIX/YV5V/2FeVf9hXIX/YV5V/
2FeVf9hXv+oGAD/qSYE/6kxDf+mPBr/pkUp/69NOP+zV0z/tGBh/7JodviuclrvqXic6KKAquKaiLbdk
o/A2YuWydWFntDSgarVzn+52L52uNm+drjZvna42b52uNm+drjZvna42b52uNm+drjZvna42b52uN
m+drjZvna42b52uNm+drjZvna42f+oGAD/qSYE/6kxDf+mPBr/pkUp/69NOP+zV0z/tGBh/7Jodviuclr
vqXic6KKAquKaiLbdko/A2YuWydWFntDSgarVzn+52L52uNm+drjZvna42b52uNm+drjZvna42b52u
Nm+drjZvna42b52uNm+drjZvna42b52uNm+drjZvna42f+oGAD/qSYE/6kxDf+mPBr/pkUp/69NOP+
zV0z/tGBh/7JodviuclrvqXic6KKAquKaiLbdko/A2YuWydWFntDSgarVzn+52L52uNm+drjZvna42b5
2uNm+drjZvna42b52uNm+drjZvna42b52uNm+drjZvna42b52uNm+drjZvna42f+oGAD/qSYE/6kxDf
+mPBr/pkUp/69NOP+zV0z/tGBh/7JodviuclrvqXic6KKAquKaiLbdko/A2YuWydWFntDSgarVzn+52
L52uNm+drjZvna42b52uNm+drjZvna42b52uNm+drjZvna42b52uNm+drjZvna42b52uNm+drjZvna4
2f+oGAD/qSYE/6kxDf+mPBr/pkUp/69NOP+zV0z/tGBh/7JodviuclrvqXic6KKAquKaiLbdko/A2YuW
ydWFntDSgarVzn+52L52uNm+drjZvna42b52uNm+drjZvna42b52uNm+drjZvna42b52uNm+drjZvn
a42b52uNm+drjZvna42f+oGAD/qSYE/6kxDf+mPBr/pkUp/69NOP+zV0z/tGBh/7JodviuclrvqXic6KK
AquKaiLbdko/A2YuWydWFntDSgarVzn+52L52uNm+drjZvna42b52uNm+drjZvna42b52uNm+drjZv
na42b52uNm+drjZvna42b52uNm+drjZvna42f+oGAD/qSYE/6kxDf+mPBr/pkUp/69NOP+zV0z/tGB
h/7JodviuclrvqXic6KKAquKaiLbdko/A2YuWydWFntDSgarVzn+52L52uNm+drjZvna42b52uNm+drj
Zvna42b52uNm+drjZvna42b52uNm+drjZvna42b52uNm+drjZvna42f+oGAD/qSYE/6kxDf+mPBr/pk
Up/69NOP+zV0z/tGBh/7JodviuclrvqXic6KKAquKaiLbdko/A2YuWydWFntDSgarVzn+52L52uNm+
drjZvna42b52uNm+drjZvna42b52uNm+drjZvna42b52uNm+drjZvna42b52uNm+drjZvna42f+oGAD/
qSYE/6kxDf+mPBr/pkUp/69NOP+zV0z/tGBh/7JodviuclrvqXic6KKAquKaiLbdko/A2YuWydWFntD
SgarVzn+52L52uNm+drjZvna42b52uNm+drjZvna42b52uNm+drjZvna42b52uNm+drjZvna42b52u
Nm+drjZvna42f+oGAD/qSYE/6kxDf+mPBr/pkUp/69NOP+zV0z/tGBh/7JodviuclrvqXic6KKAquKaiL
bdko/A2YuWydWFntDSgarVzn+52L52uNm+drjZvna42b52uNm+drjZvna42b52uNm+drjZvna42b5
2uNm+drjZvna42b52uNm+drjZvna42f+oGAD/qSYE/6kxDf+mPBr/pkUp/69NOP+zV0z/tGBh/7Jodvi
uclrvqXic6KKAquKaiLbdko/A2YuWydWFntDSgarVzn+52L52uNm+drjZvna42b52uNm+drjZvna42b
52uNm+drjZvna42b52uNm+drjZvna42b52uNm+drjZvna42f+oGAD/qSYE/6kxDf+mPBr/pkUp/69N
OP+zV0z/tGBh/7JodviuclrvqXic6KKAquKaiLbdko/A2YuWydWFntDSgarVzn+52L52uNm+drjZvna4
2b52uNm+drjZvna42b52uNm+drjZvna42b52uNm+drjZvna42b52uNm+drjZvna42f+oGAD/qSYE/6k
xDf+mPBr/pkUp/69NOP+zV0z/tGBh/7JodviuclrvqXic6KKAquKaiLbdko/A2YuWydWFntDSgarVzn+
52L52uNm+drjZvna42b52uNm+drjZvna42b52uNm+drjZvna42b52uNm+drjZvna42b52uNm+drjZvn
a42f+oGAD/qSYE/6kxDP+nOxr/qEUo/7FNOP+3Vkv/uV5h/7hmdve1bovusHWd56p9rOGihLnbmo
vE1pOSztKLmtfOhafdwn+04LR6td+0erXftHq137R6td+0erXftHq137R6td+0erXftHq137R6td+0erXf
tHq137R6td+0erXftHq13/+oFwD/qiYE/6oxDP+nOxn/qkUo/7RNN/+7VER+vV1g/r5kd/e8a4zuuHKe5
rN5rt+qgLzZo4b11JuN0s+Sld3KiqTmt4Kw56uBs+KrgbPiq4Gz4quBs+KrgbPiq4Gz4quBs+KrgbPiq
4Gz4quBs+KrgbPiq4Gz4quBs+KrgbPiq4Gz4v+pFwD/qiUD/6oxDP+oOxn/rEUUn/7ZMNv6+U0r7wlt
g+sRid/fDalztwW+g5b11sN60e77YrYHL0qWH1s2fkeDEmaLoro2u6qGlseahiLHmoYix5qGlseahiL
HmoYix5qGlseahiLHmoYix5qGlseahiLHmoYix5qGlseahiLHmoYix5v+pFwD/qyUD/6sxC/+pOxn/r
kQm/7hMNfvCUkn4x1pg9clgd/LLZo3ty2uh5Mlxst3AdcHWu3zN0reF1s21kt+1opppJus7ZeRr+mX
ka/pl5Gv6ZeRr+mXka/pl5Gv6ZeRr+mXka/pl5Gv6ZeRr+mXka/pl5Gv6ZeRr+mXka/pl5Gv6f+pFwD/
qyUD/6sxC/+pOxn/r0Ql/btLNfnGUUn0zFlf8dBed+zTY43n1mei4NJss9nOccHRyHjMysOA1cC+id6
qrZDomKak7o6druqOna7qjp2u6o6druqOna7qjp2u6o6druqOna7qjp2u6o6druqOna7qjp2u6o6druq

Ona7qjp2u6v+qFgD/qyUD/6sxC/+pOxj/sUQI+71LNPbJUJx0Vdf7Nhcd+bdYI7f3WOj1tZns83PbcD
EyXLLu8R41LC+ft2huYfnkLaf7liurumlrq7piK6u6Yiurumlrq7piK6u6Yiurumlrq7piK6u6Yiurumlrq7pi
K6u6Yiurumlrq7piK6u6f+qFgD/rCUD/6wxC/+qOxj/s0Qk+MBLM/LNUEft2FZe5uBaduHnXI7W4GGi
zNhlsLQab+4yWzKr8Rw06O/c9ySunrlhLiP6IG5p+eBuafngbm54G5p+eBuafngbm54G5p+eBu
afngbm54G5p+eBuafngbm54G5p+eBuafngbm5/+qFgD/rCUD/60xC/+rOxj/tkMj9cNKM7ST0b
o31Rd4elXdtrpW43O4mCgw9lkr7nRZryuy2jHpcZr0JjBbdmHvHHie7qD5Xe7mOR3u5jkd7uY5He7
mOR3u5jkd7uY5He7mOR3u5jkd7uY5He7mOR3u5jkd7uY5He7mOR3u5jkd7uY5P+rFQD/rSQD/6
0xC/+rOxj8uEMi8cdKMONYTtKi51Jc3e9VddPsWovG41+eu9tirLDY7mlzWXDnMhozJDEa9SBw
G/bdr593nO+jt5zvo7ec76O3nO+jt5zvo7ec76O3nO+jt5zvo7ec76O3nO+jt5zvo7ec76O3nO+jt5zvo
7ec76O3v+rFQD/riQD/64wCv+tOxf5vEMg7cxJL+TeTkPd7FFb2fRUdMvuWYm+5F2bstxfqajVYbW
d0GK/k8tlx4jHaM99w27Vc8J52G/Bh9hvwYfYb8GH2G/Bh9hvwYfYb8GH2G/Bh9hvwYfYb8GH2G/
Bh9hvwYfYb8GH2G/Bh9hvwYfYb8GH2P+sFAD/riQC/68wCv+wOhX1wEie59JILN3ITUDW8FFb0
PZTc8PwWle25luYqt5dpp/YXrCV02C5i89jwYHLZsh4yGzNcMZ10GzGgNFsxoDRbMaA0WzGgN
FsxodrBmaA0WzGgNFsxoDRbMaA0WzGgNFsxoDRbMaA0WzGgNFsxoDRbMaA0f+tEwD/ryMC
/7AwCf60OhPvxkEc4NtHKdbqTUHO9VBaxvpScbrxVoSt6FmUoeFboZbcXKuM1l6zg9NguXvPZL9
yzWnEbMtyx2jLe8hoy3vlaMt7yGjLe8hoy3vlaMt7yGjLe8hoy3vlaMt7yGjLe8hoy3vlaMt7yGjLe8hoy
3vlaMt7yP+uEgD/sSMC/7lwCfi6ORHnz0AY2ORHKM7wTEHF+k9ZvP5QbrD0U3+k61aOmOVYm
43fWqSE21yre9hesHTVYrVu02e5aNFvvGXRdr1l0Xa9ZdF2vWXRdr1l0Xa9ZdF2vWXRdr1l0Xa9Z
dF2vWXRdr1l0Xa9ZdF2vWXRdr1l0Xa9ZdF2vf+vEQD9silB/bQvCPDCOA3b3DwTz+xHKsX4S0G
7/0xXsf9Oaqf3UHqa71SHjulXkoTkWZt84VuhdN5dpm7cYKpo2mWtZNIsr2HYcrFh2HKxYdhysWH
YcrFh2HKxYdhysWHYcrFh2HKxYdhysWHYcrFh2HKxYdhysWHYcrFh2HKxYdhysWHYcrFh2HKxYdhysfuxDgD6tSAB+
rcvB+TPMwjR5j0WxvRGK7v/SECw/0ITpv9LZJz7TXOQ9FJ/he9ViHzrWJB16FqVb+ZdmWnkYJxk
4mSfYOFpoV3gbaJd4G2iXeBtol3gbaJd4G2iXeBtol3gbaJd4G2iXeBtol3gbaJd4G2iXeBtol3gbaJd4
G2iXeBtovC0CgD3uB4A78lqA9TgLGfH8T4YvP5DLLD/RD6l/0Vomv9HXZD/SmqF+k90fPZTfHTy
V4Ju8FqHau5dimXtYl1h62OPXupokVzqa5Jc6muSXOprklzqa5Jc6muSXOprklzqa5Jc6muSXOprkl
zqa5Jc6muSXOprklzqa5Jc6muSXOprkt65BQDyvRoA2tgWAMnuMgq8/DwasP8/KqT/PzmY/0FIj9
EVYT/R2B7/01pc/1Sb237VnRo+Vl3ZfdceWH2X3tf9WJ9XPRmf1r0aoBa9GqAWvRqgFr0aoBa9G
qAWvRqgFr0aoBa9GqAWvRqgFr0aoBa9GqAWvRqgFr0aoBa9GqAWvRqgNi8BADnyAwAy+kfAr
76Mw2x/zcapP84J5j/OjSM/z1Bgf9BTHj/RVvx/0lcav9PYWb/U2Ri/1dnX/9aaV3/XWpb/2FsWf5kbVf
+Z25X/mduV/5nblf+Z25X/mduV/5nblf+Z25X/mduV/5nblf+Z25X/mduV/5nblf+Z25X/mduV/5nbtTAB
ADN1QgAv/cmBLL/Lw6l/zEZmP8yJlv/NS+A/zk5df89Qm3/Qklm/0ZPYf9LUI7/T1Vb/1NXWf9XWFj/
WlpW/11bVP9gXFP/Y11T/2NdU/9jXVP/Y11T/2NdU/9jXVP/Y11T/2NdU/9jXVP/Y11T/2NdU/9jXVP/
Y11T/2NdU/9jXc7GBQDB9REAtP8IBqb/KQ6Y/ykXi/8rIH7/MClz/zUyav86OWP/P0Bd/0NEWf9ISfb
/TEpU/1BMU/9UTVH/V05Q/1tPT/9eUE7/YVFO/2FRTv9hUU7/YVFO/2FRTv9hUU7/YVFO/2FRTv
9hUU7/YVFO/2FRTv9hUU7/YVFO/2FRTv9hUf+gFwD/oSQE/6AvDP+cORn/nEMm/6RMM/+nVUT
/p2BX/6Vrav6fdH32l32O74+GnOqHj6flgJew4nuftuB4p7rfdq+93nW7v9h0x8HlbMbCxGrGwsRqxsL
EasbCxGrGwsRqxsLEasbCxGrGwsRqxsLEasbCxGrGwsRqxsLEasbCxGrGwv+gFwD/oSQE/6Av
DP+cORn/nEMm/6RMM/+nVUT/p2BX/6Vrav6fdH32l32O74+GnOqHj6flgJew4nuftuB4p7rfdq+93n
W7v9h0x8HlbMbCxGrGwsRqxsLEasbCxGrGwsRqxsLEasbCxGrGwsRqxsLEasbCxGrGwsRqxsL
EasbCxGrGwv+gFwD/oSQE/6AvDP+cORn/nEMm/6RMM/+nVUT/p2BX/6Vrav6fdH32l32O74+Gn
OqHj6flgJew4nuftuB4p7rfdq+93nW7v9h0x8HlbMbCxGrGwsRqxsLEasbCxGrGwsRqxsLEasbCxG
rGwsRqxsLEasbCxGrGwsRqxsLEasbCxGrGwv+gFwD/oSQE/6AvDP+cORn/nEMm/6RMM/+nVU

T/p2BX/6Vrav6fdH32l32O74+GnOqHj6flgJew4nuftuB4p7rfdq+93nW7v9h0x8HlbMbCxGrGwsRqx
sLEasbCxGrGwsRqxsLEasbCxGrGwsRqxsLEasbCxGrGwsRqxsLEasbCxGrGwv+gFwD/oSQE/6
AvDP+cORn/nEMm/6RMM/+nVUT/p2BX/6Vrav6fdH32l32O74+GnOqHj6flgJew4nuftuB4p7rfdq+9
3nW7v9h0x8HlbMbCxGrGwsRqxsLEasbCxGrGwsRqxsLEasbCxGrGwsRqxsLEasbCxGrGwsRqx
sLEasbCxGrGwv+gFwD/oSQE/6AvDP+cORn/nEMm/6RMM/+nVUT/p2BX/6Vrav6fdH32l32O74+G
nOqHj6flgJew4nuftuB4p7rfdq+93nW7v9h0x8HlbMbCxGrGwsRqxsLEasbCxGrGwsRqxsLEasbCx
GrGwsRqxsLEasbCxGrGwsRqxsLEasbCxGrGwv+gFwD/oSQE/6AvDP+cORn/nEMm/6RMM/+nV
UT/p2BX/6Vrav6fdH32l32O74+GnOqHj6flgJew4nuftuB4p7rfdq+93nW7v9h0x8HlbMbCxGrGwsRq
xsLEasbCxGrGwsRqxsLEasbCxGrGwsRqxsLEasbCxGrGwsRqxsLEasbCxGrGwv+gFwD/oSQE/6
AvDP+cORn/nEMm/6RMM/+nVUT/p2BX/6Vrav6fdH32l32O74+GnOqHj6flgJew4nuftuB4p7rfdq+9
3nW7v9h0x8HlbMbCxGrGwsRqxsLEasbCxGrGwsRqxsLEasbCxGrGwsRqxsLEasbCxGrGwsRqx
sLEasbCxGrGwv+gFwD/oSQE/6AvDP+cORn/nEMm/6RMM/+nVUT/p2BX/6Vrav6fdH32l32O74+G
nOqHj6flgJew4nuftuB4p7rfdq+93nW7v9h0x8HlbMbCxGrGwsRqxsLEasbCxGrGwsRqxsLEasbCx
GrGwsRqxsLEasbCxGrGwsRqxsLEasbCxGrGwv+gFwD/oSQE/6AvDP+cORn/nEMm/6RMM/+nV
UT/p2BX/6Vrav6fdH32l32O74+GnOqHj6flgJew4nuftuB4p7rfdq+93nW7v9h0x8HlbMbCxGrGwsRq
xsLEasbCxGrGwsRqxsLEasbCxGrGwsRqxsLEasbCxGrGwsRqxsLEasbCxGrGwv+gFwD/oSME/6
AvDP+cORj/nkMI/6ZLMv+qVEP/q15X/6loa/6kcn31nXuP7pWEnuiNjKrhZW04H+cu916pcDbeK7E
2ni7xtBzw8jAbMLJvW7Exr1uxMa9bsTGvW7Exr1uxMa9bsTGvW7Exr1uxMa9bsTGvW7Exr1uxMa
9bsTGvW7Exv+hFgD/oiME/6EuDP+dORj/oUlk/6ILMf+uU0L/r11W/65ma/6qbn70pXeR7Z6BoeaVi
a7hjJG53lSYwtl/ocnWfKzN1Xu70MRzvdG3csDNtXPBy7VzwcU1c8HLtXPBy7VzwcU1c8HLtXPBy7
VzwcU1c8HLtXPBy7VzwcU1c8HLtXPBy/+hFgD/oiME/6EuDP+eORj/o0lj/6tKMP+yUkH/tFtV/7Nka/
2xbH/0rHSS7KZ9o+WdhbHfll292YuVyNWEntDSf6nVzn252Lp1udevd73RrXi/zq14v86teL/OrXi/zq
14v86teL/OrXi/zq14v86teL/OrXi/zq14v86teL/OrXi/zv+iFgD/oyME/6luC/+fOBj/pUEi/65KL/+1UUH/u
FpV/7hiav23aX/zs3GT6615peOmgbTdnljB1pORzdKMm9XQiqrZxoa327J8uNqofLvUpn290qZ9vd
Kmfb3Spn290qZ9vdKmfb3Spn290qZ9vdKmfb3Spn290qZ9vdKmfb3Spn290v+iFgD/oyMD/6luC/+f
OBf/p0Eh/7BJLv+4UED+vFIU/L5gavq9Z4Dzum6U6rZ1puKufbbcpoXE1qGQzdCam9XEkaTbvY61
3aqDtt2ggbrXn4K71J+Cu9SfgrvUn4K71J+Cu9SfgrvUn4K71J+Cu9SfgrvUn4K71J+Cu9SfgrvUn4K
71P+iFQD/pCMD/6MuC/+gOBf/qEEh/7JLv67UD/7wFdU+MNeavfDZYDywmUv6cByp+K5ebfctIT
D1K2PzceiltW5l53br5Kr36KLtd+Yhrja4a615eGuteXhrrXI4a615eGuteXhrrXI4a615eGuteXhrrXI4a6
15eGuteXhrrXI4a61/+jFQD/pCMD/6MuC/+gOBf/qkEg/7RJLf+Tz74xFTZ9chdavLKYodWyiW6ctu
qOLJeLbZwYHCzLWKzL6qkdWvnpfco5ej4ZqUs+GQjLfcj4u42Y+LuNmPi7jZj4u42Y+LuNmPi7jZj4
u42Y+LuNmPi7jZj4u42Y+LuNmPi7jZj4u42f+iFQD/pCMD/6QuC/+hOBf/q0Ef/7ZILPnAtz71yVVT8c
5bae7SYIDp1WWW5NhsqNzTdbXRyn7BxL2FzLeyjNSnppHbmZ2b4ZKes+KHk7beh5K324eSt9u
Hkrfbh5K324eSt9uHkrfbh5K324eSt9uHkrfbh5K324eSt9uHkrfbh5K32/+jFQD/pSMD/6QuC/+hOBb/
rUEf/bhIK/bDTj3xzI RS7dVaaenbXoDj4WKW29xpp9LVcbXlz3nAvcaAy7C7h9OgrovbKkWV4Yinre
KAnLXegJq33lCat9yAmrfcgJq33lCat9yAmrfcgJq33lCat9yAmrfcgJq33lCat9yAmrfcgJq33P+kFQD/
pSID/6UuCv+jOBb/r0Ae+rtIKvPHTjzu1FJR6N1YaOPIWoDc5WCv0d5mpsJWbbS9z3PAs8l6yqnFg
dKaulbaibCQ34KyqeB7p7bde6S323ukt9t7pLfbe6S323ukt9t7pLfbe6S323ukt9t7pLfbe6S323ukt9t7
pLfbe6S32/+kFAD/pilD/6UuCv+IOBX/sUAd975HKe/LTTrp21FP4+VVZ9/tWH/T516UyN9jpb3WaL
Oy0G2+qMpyyZ7Fd9GRwn7YhL+M3Xy+pN13trfbdrG42XaxuNI2sbjZdrG42XaxuNI2sbjZdrG42Xa
xuNI2sbjZdrG42XaxuNI2sbjZdrG42f+kFAD/pilD/6YuCv+nNxT/tEAb9MJHJ+rRTDjj4IBN3u1TZtnxV
n3K6VySvuBgpLPYY7Gn0Wa9nMtpx5HGbs+Ew3TVecGA2XPBltpxwa/Yc8K51nPCudZzwrnWc8

K51nPCudZzwrnWc8K51nPCudZzwrnWc8K51nPCudZzwrnWc8K51v+IEwD/pylC/6cuCf+qNxL7u
D8a78dGJeTZSzbC6E9M1vJSZdD0VXzB61qQtOFdoKnaX66c02G6kM1jxITIZsx5xWvSb8N21Wv
DiNVpxJ/UasWn0mrFp9JqxafSasWn0mrFp9JqxafSasWn0mrFp9JqxafSasWn0mrFp9JqxafSasW
n0v+mEwD/qCEC/6gtCf+tNxH2vT4X6M9FItziSJLU7U9MzfZSZMb3U3m47ViMq+RbnJ/dXKmT1I6
zh9FhvH3NZMNzymnlbMhyzGflgM1kyJPMZMiZzGTImcxkyJnMZMiZzGTImcxkyJnMZMiZzGTImc
xkyJnMZMiZzGTImcxkyJnMZMiZzP+nEgD/qSEC/6ktCP+yNg7vxDOU39IDHtToSjPM805MxPxQY
7z5UXau71aloedYI5XgWqKJ21ysf9Zfs3bTYrlv0Ge9aM9vwGTOesJhzorCYM6PwmDOj8Jgzo/CY
M6PwmDOj8Jgzo/CYM6PwmDOj8Jgzo/CYM6PwmDOj8Jgzo/CYM6Pwv+oEQD/qiAC/6stCPm4N
QvmzjsQ1uNCHczvSjTD+k1Luf9OYLH7T3Kk81OCI+tWklvIWJqB4Fujd9xdqXDaYa5q2GaxZdZst
GHVdbVe1YK2XdWGtl3VhrZd1Ya2XdWGtl3VhrZd1Ya2XdWGtl3VhrZd1Ya2XdWGtl3VhrZd1Ya2
XdWGtv+pDwD/rB8B/60sB/DCMgjZ3TUMzexDH8P4STW5/0pJrv9LXKX/TG2Z91B7jfBUh4LqV5B
55lqYceNdnWvhYKFI32SkYd5ppl3dcaha3XqpWd1+qVndfqlZ3X6pWd1+qVndfqlZ3X6pWd1+qVnd
fqlZ3X6pWd1+qVndfqlZ3X6pWd1+qfysDAD/rx4B+rYpBOHRKQPP6DgOw/VClbn/RjSu/0ZGo/9HV
5n/SWWO+05yg/ZSfHrxVoRy7ImKa+tdj2bqYJj6GSUX+dollvmbphY5XWZV+V4mVfleJIX5XiZV+
V4mVfleJIX5XiZV+V4mVfleJIX5XiZV+V4mVfleJIX5XiZV+V4mfGvCQD8shsA7MMfANHjJQPE8zk
Ruf8/lq3/QTki/0FBI/9EUI3/R12D/0toef1QcHL5VXZr9lh7ZvRcfmPzX4Ff8mODXfFnhVrwbLZX73KH
Vu90iFbvdIhW73SIVu90iFbvdIhW73SIVu90iFbvdIhW73SIVu90iFbvdIhW73SIVu90iN+zAwD4txcA
1NgJAMbwLAa5/jcTrf86laH/Oy+V/z08iv9ASID/RFN4/0hccP9OY2r/U2hl/1drYf5bbI79XnBc/GJxWv
tlc1j6aXRV+m51VfpwdVX6cHVV+nB1VfpwdVX6cHVV+nB1VfpwdVX6cHVV+nB1VfpwdVX6cHVV
+nB1Vfpwddi3AgDUxwYAx+4XALr8LQmt/zIUof80H5T/NSqJ/zg1fv88QHT/QEIs/0VPZv9JVWH/T1
he/1RbW/9YXVn/W19Y/19gVv9iYVT/ZmJS/2pjUv9rY1L/a2NS/2tjUv9rY1L/a2NS/2tjUv9rY1L/a2NS
/2tjUv9rY1L/a2NS/2tjUv9rY9S7AwDI0wYAu/ofAq7/KQqh/ysTIP8tHlf/LyZ8/zMvvcv84N2n/PT5i/0FD
Xf9GR1j/SkpW/05MVP9STIL/Vk9R/1IQUP9cUE//YFFN/2RSTf9IUk3/ZVJN/2VSTf9IUk3/ZVJN/2VS
Tf9IUk3/ZVJN/2VSTf9IUk3/ZVJN/2VSTf9IUUsnFBQC88AkAr/8fBKL/lgqU/yQRh/8mGXr/KiFv/y4pZv
80L1//ODRZ/z05Vf9CPFH/Rj5P/0pATf9OQkz/UUNL/1RDSv9XREn/W0VI/19GR/9gRkf/YEZH/2BG
R/9gRkf/YEZH/2BGR/9gRkf/YEZH/2BGR/9gRkf/YEZH/2BGR/9gRv+ZFQH/mSIE/5csDP+SNhj/kk
Aj/5IJLv+dUz3/nF9O/5lqYP+TdHH9jH6A9oSljfF9kpfud5yf63OkpOlwrKfob7Sq6G+9q+dvyazebNCt
0mfSrspk0q/KZNVymTSr8pk0q/KZNVymTSr8pk0q/KZNVymTSr8pk0q/KZNVymTSr/+ZFQH/
mSIE/5csDP+SNhj/kkAj/5IJLv+dUz3/nF9O/5lqYP+TdHH9jH6A9oSljfF9kpfud5yf63OkpOlwrKfob7S
q6G+9q+dvyazebNCt0mfSrspk0q/KZNVymTSr8pk0q/KZNVymTSr8pk0q/KZNVymTSr8pk0q/K
ZNVymTSr/+ZFQH/mSIE/5csDP+SNhj/kkAj/5IJLv+dUz3/nF9O/5lqYP+TdHH9jH6A9oSljfF9kpfud
5yf63OkpOlwrKfob7Sq6G+9q+dvyazebNCt0mfSrspk0q/KZNVymTSr8pk0q/KZNVymTSr8pk0q/
KZNVymTSr8pk0q/KZNVymTSr/+ZFQH/mSIE/5csDP+SNhj/kkAj/5IJLv+dUz3/nF9O/5lqYP+TdH
H9jH6A9oSljfF9kpfud5yf63OkpOlwrKfob7Sq6G+9q+dvyazebNCt0mfSrspk0q/KZNVymTSr8pk0q/
KZNVymTSr8pk0q/KZNVymTSr8pk0q/KZNVymTSr/+ZFQH/mSIE/5csDP+SNhj/kkAj/5IJLv+dU
z3/nF9O/5lqYP+TdHH9jH6A9oSljfF9kpfud5yf63OkpOlwrKfob7Sq6G+9q+dvyazebNCt0mfSrspk0q
/KZNVymTSr8pk0q/KZNVymTSr8pk0q/KZNVymTSr8pk0q/KZNVymTSr/+ZFQH/mSIE/5csDP
+SNhj/kkAj/5IJLv+dUz3/nF9O/5lqYP+TdHH9jH6A9oSljfF9kpfud5yf63OkpOlwrKfob7Sq6G+9q+dv
yazebNCt0mfSrspk0q/KZNVymTSr8pk0q/KZNVymTSr8pk0q/KZNVymTSr8pk0q/KZNVymTSr
/+ZFQH/mSIE/5csDP+SNhj/kkAj/5IJLv+dUz3/nF9O/5lqYP+TdHH9jH6A9oSljfF9kpfud5yf63OkpOl
wrKfob7Sq6G+9q+dvyazebNCt0mfSrspk0q/KZNVymTSr8pk0q/KZNVymTSr8pk0q/KZNVymTS
r8pk0q/KZNVymTSr/+ZFQH/mSIE/5csDP+SNhj/kkAj/5IJLv+dUz3/nF9O/5lqYP+TdHH9jH6A9oSlj

fF9kpfud5yf63OkpOlwrKfob7Sq6G+9q+dvYazebNct0mfSrspk0q/KZnKvymTSr8pk0q/KZnKvymT
Sr8pk0q/KZnKvymTSr8pk0q/KZnKvymTSr/+ZFQH/miEE/5gsDP+TNhf/IT8i/5xJLf+hUzz/oF5N/55
oYP+ZcnH8kXyB9YqGkO+Cj5vre5mk6HahquZzqq7lcbKw5HG8suNxYrPXbc61ymfOtsNnz7PDZ8
+zw2fPs8Nnz7PDZ8+zw2fPs8Nnz7PDZ8+zw2fPs8Nnz7PDZ8+zw2fPs/+aFQD/miEE/5gsDP+UN
hf/mD8h/6BILP+IUDv/pVtM/6RmX/+gcHL7mXmD85GCK+2Ji6DogZWq5HqdseJ2prfgdLC63307v
NtyyL7Ma8i/wGrKvLtszbe7bM23u2zNt7tszbe7bM23u2zNt7tszbe7bM23u2zNt7tszbe7bM23u2zNt/
+aFQD/myEE/5ksC/+VNhb/mz8g/6NHK/+pTzr/qllM/6ljX/+mbHL7oXaE8pp/leuRiKPmiJGv4YCUuN
56o7/cd67D2na7xtByw8jAa8Lt2/HwbNxyryzccq8s3HKvLNxyryzccq8s3HKvLNxyryzccq8s3HKvLN
xyryzccq8s3HKvP+bFAD/nCEE/5osC/+WNRb/nT8f/6VHKv+sTzn/r1hL/65hX/2sanL6qHKF8aF8l+q
ZhKbkk12z3oaVvtp+n8fYe6zL132+zMd3wMy4ccDMr3TFxax1yL+sdci/rHXlv6x1yL+sdci/rHXlv6x1y
L+sdci/rHXlv6x1yL+sdci/rHXlv/+bFAD/nCEE/5osC/+WNRb/nz4e/6hGKf+vTjj/s1dK/bNfXvqxZ3P3r
m+G8al4memiganiml33l+TwteJn8nQharOzlS5z79+vs+wd7/PqHjDyKV6xsOlesbDpXrGw6V6xs
OlesbDpXrGw6V6xsOlesbDpXrGw6V6xsOlesbDpXrGw/+cFAD/nCED/5srC/+XNRb/oT4d/6pGKP
+yTjf+tIVJ+7hdXvi3ZXP0tG2H8LB1muiqfavho4e32JqRws2QmsrFiqXPwImx0beGvdKpfb3RoXzBy
59+xMaffsTGn37Exp9+xMaffsTGn37Exp9+xMaffsTGn37Exp9+xMaffsTGn37Exv+cFAD/nSAD/5s
rCv+YNRb/oz4c/6xGJ/+0TTb9ulRI+bxcXfa8Y3Lxu2qH7Lhxm+WzeqvbqoW30KCNwcWWlsq7kJ/
QtY2r07GMu9Sig7zTm4G/zpiBwsiYgcLImIHcyJiBwsiYgcLImIHcyJiBwsiYgcLImIHcyJiBwsiYgcL
mIHcyP+cFAD/nSAD/5wrCv+ZNRX/pD0b/65GJv+2TDX7vINI98FbXfPCYXLvwmel6MFum+C6eKr
VsYK2yaaKwb2dksqzLzVrQ5GI1KeQtdabibrVIIW+0JKFwcqShcHKkoXByPKFwcqShcHKkoXByPK
FwcqShcHKkoXByPKFwcqShcHKkoXByv+dEwD/niAD/5wrCv+aNRT/pj0b/7BFJfy5TDT4wlJH9MZ
ZXPJHX3Lsy2WI5ctsmvDdanPt361wqyGwLajjsqsm5bRopWf1ZyUrdeUkLnXjlq80ouJwM2LicDni
4nAzYuJwM2LicDni4nAzYuJwM2LicDni4nAzYuJwM2LicDni4nAzf+dEwD/niAD/50rCv+cNRT/qD
0a/7JFJPq8TDP1xIFG8MxYW+zRXXLo1WKH4NRqmtXLcqjJvnu1vLOCv7CqicmlopHRmpuZ1pK
Zp9iMmLnYhI+71ISOv86Ejr/Ohl6/zoSOv86Ejr/Ohl6/zoSOv86Ejr/Ohl6/zoSOv86Ejr/Ohl6/zv+dEw
D/niAD/50rCv+eNRP/qj0Z/rREI/e/SzLxy1BE7NRWWufbWnHj4l+G2t1nmc/Sb6fCxae0trp+v6qxhMi
eqYvQkaOT1oigodmEo7jZfJe71X2Uvs99IL7PfZS+z32Uvs99IL7PfZS+z32Uvs99IL7PfZS+z32Uvs
99IL7PfZS+z/+eEwD/nyAD/54rCv+fNBL/rD0Y/LdElvPDSjDt0E9D591UWeLmV3Dd6VyF0uVkl8jba
6a7z3Ozr8N5vaO6f8eWsobPiqyO1YCqnNd8rbTXdqK71XedvtB3nb7Qd52+0HedvtB3nb7Qd52+0
HedvtB3nb7Qd52+0HedvtB3nb7Qd52+0P+eEgD/nyAD/58rCf+iNBH/rjwX+LtEIO/ISi7n105B4udR
V97uVW7V7lqEyedhlr7fZ6Wz2G2xqM5zvJvFecaOvn/Ng7mJ0nq4mdV3u7HVcbC803Gov85xqL/O
cai/znGov85xqL/Ocai/znGov85xqL/Ocai/znGov85xqL/Ocai/zv+fEgD/oB8C/58rCf+kNBD/sjwV9L9D
HunPSSzg4E0+2+tRVdbyVG3M8liCv+hdlbPgYqSo2WewnNJrupHNccOFyXjLe8eDz3TGk9Fwxqj
QbMK+z2y3wctst8HLbLfBy2y3wctst8HLbLfBy2y3wctst8HLbLfBy2y3wctst8HLbLfBy/+fEgD/oR8C/
6ErCf+nMw77tjsT7sVCG+LYRyjZ5k080u9QVcz2U2zC9FaAtOlakqjhXKKc2l+uj9NjuIPPZ8F4y27H
cMI3ymrJhcxnyZnLZcmvymfLw8dny8PHZ8vDx2fLw8dny8PHZ8vDx2fLw8dny8PHZ8vDx2fLw8dny
8PHZ8vDx/+gEQD/oh8C/6lqCP+rMwz2uzoQ585AGNnhRiXR7E09yfVQVMH8UWq49VN9quxXjp7
kWpyR3VuohNdesnjSYbpuz2fAZ81vw2PNe8RgzYvEXc2fw17OrsFezq7BXs6uwV7OrsFezq7BXs
6uwV7OrsFezq7BXs6uwV7OrsFezq7BXs6uwf+hEAD/ox4C/6MqCP+xMQruwzgN3Ns9E9HoRyfJ8
0w9wPxOU7f/T2et+FF4oO9ViJTov5WH4VqgfN1cqXLZYK9q1ma0ZNRttmDUdrhd04O4W9SUuFr
UoLda1KC3WtSgt1rUoLda1KC3WtSgt1rUoLda1KC3WtSgt1rUoLda1KC3WtSgt/+jDgD/pR0B/6Yp
B/i4LwbjzzMI0+Q9FMnwRyi/+ko9tf9LUaz/TGOj+05zlvNSgYrsVo1/51mWdeNcnm3gYKNm3mSnY
d1qqV3ccqta23yrWNUKrFbIKtX25SrV9uUq1fbIKtX25SrV9uUq1fbIKtX25SrV9uUq1fbIKtX25SrV9u

[illegible]

ddCyqHXQsqh10LKoddCyqHXQsqh10LKoddCyqHXQsv+UEwD/IB4E/5IpC/+QMhT/mzsZ/6NDI/+
qSy/+r1NA+q9cUveuZGbzm67qZ1jeiffp7glYes2l2Rtc+Emr3lgKXBxH6yw8F/wsS2esXDq3TGw6
Z3yruieM61onjOtaJ4zrWieM61onjOtaJ4zrWieM61onjOtaJ4zrWieM61onjOtf+VEgD/lR4E/5MpC/+
RMhP/nTsZ/6VDlv+sSi78s1E/+LRaUvSzYmbwsWp66q1yjuOme57anYWwR0JOOtcaKlr2+haDDuY
OsxraDu8avgMTGpHnExp97yb6cfM23nHzNt5x8zbecfM23nHzNt5x8zbecfM23nHzNt5x8zbecfM23
nHzNt/+VEgD/lR4E/5MoCv+TMhP/nzoY/6hClf+vSi36tIA99rIYUfG5YGXsuGd65rVvjt6teJ3TPlGqyJ
qKtb6Rkr62ipvEslemyKyHtMmohsLJnX/CyJh/x8GWf8u6ln/LupZ/y7qWf8u6ln/LupZ/y7qWf8u6ln/Lu
pZ/y7qWf8u6ln/Luv+VEgD/lh4D/5QoCv+VMhL/oDoX/6pCH/6ySSz4ulA89L5XUO/AXmXpwGR64r
xsjdi0dpzNqn6pwqCGtLeXjr2ukJbFp4yhyaOLrsugjL/LIYXBy5CCxcSPg8m9j4PJvY+Dyb2Pg8m9j4
PJvY+Dyb2Pg8m9j4PJvY+Dyb2Pg8m9j4PJvf+WEgD/lh4D/5QoCv+WMhH/ojoW/6xCHvy1SSv2v
U878cRVtuzIXGTlymJ53cNqjNK6c5vHr3uovKaDs7Gdir2nlPLFnpGcypmQqM2WkbnNjYvAzYmH
xMalhsi/ilblv4iGyL+lhsi/ilblv4iGyL+lhsi/ilblv4iGyL+lhsi/ilblv/+WEgD/lx4D/5UoCv+YMHd/pDkV/65
BHfq4SCn0wk4578tTTenRWWLi0V942Mtnis3AcJrBtntqx/s6qkhrygnY3FlpeXy4+Vo86LlrPOhZ
K/zoCMwsiBi8fBgYvHwYGLx8GBi8fBgYvHwYGLx8GBi8fBgYvHwYGLx8GBi8fBgYvHwf+XEgD/lx
4D/5YoCv+aMRD/pzkU/7FBHPe8Ryjwx0046tNRS+XcVmHd2lx30tJkicfHbZm7vXSmsLN7sqSrgby
YpljEjZ6Ry4Wcnc6Bna3PfZy/zniUwsp6kcbCepHGwnqRxsJ6kcbCepHGwnqRxsJ6kcbCepHGwnq
RxsJ6kcbCepHGwv+XEQD/mB4D/5YoCf+cMQ//qTkT/LRAGvPARybrzUw15dtQSN/IU2DW4Vp1z
NphiMDPaZi1xXCqlqbx2sJ2zfLqRrIPDhaeMyn2lmM15pqnOdqi/zXGdwslymcbDcpnGw3KZxsNymc
bDcpnGw3KZxsNymcbDcpnGw3KZxsNymcbDcpnGw/+YEQD/mB4D/5coCf+fMA3/rDgR+Lg/GO7
GRiPl1Uusy3uVORdnqUV7P6Fh0xOFehrnZZZeuz2ukosZxr5a+d7mJt33BfrKGx3axlMpysqXLcLS8
ymupw8hso8bCbKPGwmyjxsJso8bCbKPGwmyjxsJso8bCbKPGwmyjxsJso8bCbKPGwv+YEQD/
mR0D/5goCf+iMAz/sDcP9L0/FujNRCDd30ku1uhPRNDuUVzH7VvyvOhbhbHiYJWm22aimtJrrY7
KcbaBxHe+d8GBxHDAkMZswaLHasS5xma5xcRmsMi/ZrDlv2awyL9msMi/ZrDlv2awyL9msMi/ZrD
lv2awyL9msMi/ZrDlv/+ZEAD/mh0C/5koCP+mLwr7tDYN7cU9Et/YQhvV5Uotzu5ORMf0UVu/81Nw
s+9XgqjpW5Kc4l+gkNxxqoTWarN40nG6b9B7vmnPisBmz5rBY8+uwGHOyL9hwMu6YcDLumHAy
7phwMu6YcDLumHAy7phwMu6YcDLumHAy7phwMu6YcDLuv+aEAD/mx0C/5soCP+qLg1ujQK5
c46DtbIQRrO7EouxfRNRL36T1m1+FBtqvRTf5zrVo+P5Fmcgt5cp3bZYK9s1We1ZNNxuGDTfblD0
4y5XNOeuVrUs7dc1s20XNbNtFzWzbRc1s20XNbNtFzWzbRc1s20XNbNtFzWzbRc1s20XNbNtP+
bDgD/nRwC/58nBv+wLAXtwzEG2d0zCs7pQhvF80kvvPxMQ7P/TFeq/k1pn/hQeZLwVleG6VeTeu
NbnW/fX6Rm3GSpYNTsrFzadq5Z2YKuV9qSrlbao61U2rOtVNqzrVTas61U2rOtVNqzrVTas61U2rO
tVNqzrVTas61U2rOtVNqzrf+dQD/nxsB/6UIBPe4KALg0icCz+Y2DMXyQh28+0cwsv9lQqj/SVSf/0
pkltfOcon1Un5971aJc+pakWrnX5dj5GSbXuNrnIricqBX4XyhVeGloVPhl6BS4qSgUuKkoFLipKBS4
qSgUuKkoFLipKBS4qSgUuKkoFLipKBS4qSgUuKkoP+fCwD/oRoB/60hAevEHQDS4iMCxvA5D7z
7QR+y/0Qvp/9FP53/RU+U/0hdiv9Lan/6UXR19lV8bfJag2bvXohh7WSMXOxpjInrcJBW6neRVOqB
kVLqjZFR6pmRUeqZkVHqmZFR6pmRUeqZkVHqmZFR6pmRUeqZkVHqmZFR6pmRUeqZk6hC
QD/pBgA97cVANTVCQDH7ioEvPo4EbH/PR+n/z8tnP9AO5L/Qkml/0VVf/9lYHb/T2hu/VRvZ/pZdG
L4Xnhe92N7WvVofVf1bX5V9HN/U/R7gFHzhYBQ84+AUPOPgFDzj4BQ84+AUPOPgFDzj4BQ84
+AUPOPgFDzj4BQ84+AUPOPgPOIBQD/rBAA1MUGAMnsEgC8+S0Hsf81Eqb/OB6b/zkqkP88No
b/PkJ8/0JMc/9GVGz/S1tm/1FhYf9XZF3/XGda/2FqV/9ma1X/am1T/m9uUf51blD9fm9O/YZvTv2Gb
079hm9O/YZvTv2Gb079hm9O/YZvTv2Gb079hm9O/YZvTv2Gb+CpAADVUAMAYM8GAL34HQGx
/ywJpv8vEpr/MRyO/zQmg/83MXn/Ojpw/z5CaP9DSWL/R05d/01SWv9TVVf/WFdV/11ZU/9iWIH/Zlt
Q/2pcTv9vXU3/dl1L/3xeS/98Xkv/fF5L/3xeS/98Xkv/fF5L/3xeS/98Xkv/fF5L/3xeS/98XtiuAADJwwQ

AvdsEALH/HwOI/yYJmf8pEY3/KxmB/y4id/8xK23/NTJl/zo4Xv8/PVn/Q0FV/0hEUf9NRk//UkhN/1ZJ
TP9bSkv/X0tK/2NLSf9oTEj/bU1H/3JNR/9yTUf/ck1H/3JNR/9yTUf/ck1H/3JNR/9yTUf/ck1H/3JNR/9
yTcu5AQC90AIAsf8OAKX/HASY/x8JjP8hD3//JBd0/yceav8rJWH/MCpa/zUvVf86M1H/PzVN/0M3S
/9HOUn/Szph/047Rv9SPEX/VT1E/1k9Q/9dPkH/Yj5A/2c/QP9nP0D/Zz9A/2c/QP9nP0D/Zz9A/2c/
QP9nP0D/Zz9A/2c/QP9nP7/FAGCy2wAApv8OAZj/FASL/xclfv8aDnL/HRRn/yEaXv8mH1b/KyNR/z
AnTf80KUn/OcTG/zwtRP9ALkL/Qy9B/0YwQP9JMD//TDE9/08xPP9TMjv/VzM6/1szOv9bMzr/WzM
6/1szOv9bMzr/WzM6/1szOv9bMzr/WzM6/1szOv9bM/+HEwL/hh0G/4EnDv96MRj/fDsf/4JEKP+FTj
P/hFtA/4BoTv97dVv/doJm/3GPb/9snHb/aad6/mewff1mun/8ZsSB/GbPgvpm3YPzZOG7WHjhOV
f5YXcW+iG1VrqhdVa6oXVWuqF1VrqhdVa6oXVWuqF1VrqhdVa6oXVWuqF1Vrqhf+HEwL/hh0G/
4EnDv96MRj/fDsf/4JEKP+FTjP/hFtA/4BoTv97dVv/doJm/3GPb/9snHb/aad6/mewff1mun/8ZsSB/G
bPgvpm3YPzZOG7WHjhOVf5YXcW+iG1VrqhdVa6oXVWuqF1VrqhdVa6oXVWuqF1VrqhdVa6o
XVWuqF1Vrqhf+HEwL/hh0G/4EnDv96MRj/fDsf/4JEKP+FTjP/hFtA/4BoTv97dVv/doJm/3GPb/9sn
Hb/aad6/mewff1mun/8ZsSB/GbPgvpm3YPzZOG7WHjhOVf5YXcW+iG1VrqhdVa6oXVWuqF1Vr
qhdVa6oXVWuqF1VrqhdVa6oXVWuqF1Vrqhf+HEwL/hh0G/4EnDv96MRj/fDsf/4JEKP+FTjP/hFtA/
4BoTv97dVv/doJm/3GPb/9snHb/aad6/mewff1mun/8ZsSB/GbPgvpm3YPzZOG7WHjhOVf5YXc
W+iG1VrqhdVa6oXVWuqF1VrqhdVa6oXVWuqF1VrqhdVa6oXVWuqF1Vrqhf+HEwL/hh0G/4EnDv
96MRj/fDsf/4JEKP+FTjP/hFtA/4BoTv97dVv/doJm/3GPb/9snHb/aad6/mewff1mun/8ZsSB/GbPgv
m3YPzZOG7WHjhOVf5YXcW+iG1VrqhdVa6oXVWuqF1VrqhdVa6oXVWuqF1VrqhdVa6oXVWu
qF1Vrqhf+IEgL/hh0G/4InDf97MRf/fzof/4VEJ/+ITTL/iFIA/4RmTv9+clz/eoBo/3SNcv9vmXn+a6V+/G
mtgvtnuLT6Z8KG+WfOh/dn3ljwZN+J6WHhiuBf5lvWW+aL0FzqiNBc6ojQXOql0FzqiNBc6ojQXOql
0FzqiNBc6ojQXOql0FzqiP+JEGl/hx0G/4MmDf98MRf/gzkd/4pDJv+OTDH/jlc//4pjTf+Fb1z/gHtq/3q
ldv10IH/6b5+G+Gypi/Zqs471ab6Q9GnLkvFo2JPoZduU4GLEldZf4ZbNXuOTx2Dmjsg5o7HYOaO
x2Dmjsg5o7HYOaOx2Dmjsg5o7HYOaOx2Dmjv+KEgH/iB0G/4QmDf99MBf/hzkc/45CJP+SSy//I
FY9/5FhTf+MbF3/hnds/oCDef6j4T2dJqN82+kk/Fsrpfwa7ma72vHm+xq1J3hZtie1mLcn8xf3J7FYu
CYwGTjk8Bk45PAZOOTwGTjk8Bk45PAZOOTwGTjk8Bk45PAZOOTwGTjk/+KEgH/iR0F/4UmDP
+AMBX/izga/5JBI/+WSi7/mVQ8/5ZfTP+Sal39jHRT+oZ/fPh/i4j0eJaT8HOHmu1vq5/sbbai62zFpOVq
0abZZtanZWHWqMVj2aO9Zt2duWjgmLlo4Ji5aOCYuWjgmLlo4Ji5aOCYuWjgmLlo4Ji5aOCYuWjg
mP+LEgH/iHwF/4YmDP+DMBT/jjgZ/5VBIf+aSiz/nVM7/5xeS/2YaFz6knFu9ox7ffOFhozvfZKY63ad
oehwp6fkbGr4Wu/rd5s0K3QadOtxWXTb1n1qe3atuhsmvenLJr3pyya96csmvenLJr3pyya96csmv
enLJr3pyya96csmvenP+LEQH/iHwF/4cmDP+GLxP/kTgY/5hBIP+eSSv/oIE5/aFcSvqeZl2mW9u8
pJ4f+2Kgo/ogoyc43iWp91yoK3ZcKyw1nG6sdNyzLHlB9GwvWrRsLZr1KywbtilrG/cn6xv3J+sb9yfr
G/cn6xv3J+sb9yfrG/cn6xv3J+sb9yfrG/cn/+MEQH/ixwF/4gmDP+ILxP/kzcX/5tAHv+hScR+plA4+qZ
aSfakY1vyn21u7Zl1gOiSfPHhh4eg2X+RqdJ5nK/NdqeyynWztMd2w7TBdc+0tnDPs65v0bCqctaop
nPao6Zz2qOmc9qjpnPao6Zz2qOmc9qjpnPao6Zz2qOmc9qjpnPao/+MEQH/jBwF/4gmC/+LLxL/ljc
W/55AHf+kRyj8qk8296tYR/OqYVrupmlu6KFygeKYe5LaklWf0laNqcl/I7DDe6K0v3qtrrx6u7e5e823
rnbNt6d0z7OjddOsoHfYpqB32Kagd9imoHfYpqB32Kagd9imoHfYpqB32Kagd9imoHfYpv+NEQH/j
BwE/4kmC/+NLxH/mDcV/6E/HP+oRyf6rU419bBWRvCwX1nqrWdt5KdvgdygeJHSI4KeyY2KqcCG
krC5gJy2tX6nubF+tbqv8a6p3vLuqB4zbecedGvmnrVqZp61amaetWpmnrVqZp61amaetWpmnrVq
Zp61amaetWpmnrVqf+NEQH/jRwE/4omC/+PLxD/mjcU/6M/G/6RiX3sU0z8rZURE22XFjntWRt365
tf9WmdpDMnX6dwpSGqLmMjrGxhpe3q4Oiu6iCrrylg7+9oIHkvJd8y7uVfdCyk33TrJN906yTfdOsk
33TrJN906yTfdOsk33TrJN906yTfdOsk33TrP+OEQH/jRwE/4omC/+RLg//nTYS/6Y+GfyuRiT1tUwx
77xSQ+q+Wlfju2Js2rRqftCsc47Go3ucu5qDp7KSirGqjJO4o4idvJ6Gqb+bh7i/mlflv4+Ayb6Ngc61jI

DSr4yA0q+MgNKvjIDSr4yA0q+MgNKvjIDSr4yA0q+MgNKvjIDSr/+OEAH/jhwE/4smC/+TLg7/nzYR
/6k+GPqxRSLzuUsw7cFRQefGWFXfwl9q1bpofcqxcl3AqXibtqB/p6yYh7Ciko64mo2YvpWLPmGRi
7LBj43FwYaFx8CGhcy4hYTQsYWE0LGFhNCxhYTQsYWE0LGFhNCxhYTQsYWE0LGFhNCxhY
TQsf+PEAH/jhwE/4wmCv+VLg3/oTUQ/6s9Fvi0RCDxvksu6sdQP+POVVPayV1pz8BlfMW3bYy6rn
WasKZ8pqafg7Ccmlq4kpOTvouQn8KHkK3DhZK/w36MxsJ+isu6fonPs36Jz7N+ic+zfonPs36Jz7N
+ic+zfonPs36Jz7N+ic+zfonPs/+PEAD/jxwE/4wmCv+XLgz/pDUP/q48FfW4Qx7uw0or589OPN/VUI
LV0FpoysZjer+9aou1tXGZqq14paCmfq+Un4W3ipqOvoKXmsJ9l6jEe5i5w3aVxsJ1kMq7d47Otxe
OzrV3js61d47OtxeOzrV3js61d47OtxeOzrV3js61d47Otf+QEAD/kBsD/40lCv+aLQv/pzQO/LI8E/O
9Qxzqykgo4tlMOdrdT1DQ11hmxM1gebnEZ4mvvG6XpLV0o5mueq2Np4G2gqKJvXmglcF0oKTDc
qG1wm+gx8Fumcq8b5XOtW+VzrVvlc61b5XOtW+VzrVvlc61b5XOtW+VzrVvlc61b5XOtf+QEAD/k
BsD/44lCf+dLQr/qjQM+bY7Ee7EQRnj00Yk2+JIN9PkTk/J3lZkvtVcd7PNY4eoxWmWnb5vopK3da
yGsXu0eqyEu3KqkL9tq6DAa6yxwGmtYl9no8u6aJ/OtGifzrRon860aJ/OtGifzrRon860aJ/OtGifzrRo
n860aJ/OtP+RDwD/kRsD/5AlCf+gLAj/rjlk87w5DubMPxXb30Qf0+dLNcvqTUzB5VNitt9ZdazYX4W
h0GSTlslqn4rCb6l+vXaxc7l/t2y4jLpouJ27Zbquu2S8yLphsM23YqrQsmKq0LJiqTcyYqrQsmKq0LJi
qtCyYqrQsmKq0LJiqTcyYqrQsv+SDgD/khsD/5QlB/+kKwb7szAH7MQ2Ct3ZOg/T5UUfy+1MNMLv
Tkq57FBgrudVcqPiWoKY3V+RjdZknlHQaqZ2zHGtbMp7smbJibVjyup1YcyttV/OxrRcwtCxXbjTrV2
4061duNOtXbjTrV24061duNOtXbjTrV24061duNOtXbjTrf+TDgD/lBoC/5gkBv+pKQT1ui0E4s8wBt
PiOg/K7EUhvwRLNLn0TUmw8k5dpe9Rb5rsVH+O6FiNgruRdmXfhY6Jt3WqoZdt1rGDag65d2pSu
W9qlrVnbuKxX2NSrWMvWp1jL1qdYy9anWMvWp1jL1qdYy9anWMvWp1jL1qdYy9anWMvWp/+U
DQD/lRoC/54iA/6vJQLrxCYC1t4oBMvrPBHC9EUuftJNa/6Skem+UtZnPdNaph1UXmF8FWGeOp
ZkG3mXplk4mSfXeBtoljfeKRW34akVN+VpFPgpaRR4LmjU+LTOfPi06BT4tOgU+LTOfPi06BT4tO
gU+LTOfPi06BT4tOgU+LTOfP+WDAD/lxkC/6QfAfa4HQDb1hAAzOgtBsL0PRO4/UMjr/9GNKX/Rk
Wb/0hVkv5KZlF8T3B89lR7cfFYhGjuXoth62SQW+lRk1fodZVV53+WU+eMlIHompVQ6KqVTui5lE7o
uZRO6LmUTui5lE7ouZRO6LmUTui5lE7ouZRO6LmUTui5lP+YCGD/mhgB/6wXANjHCADO5RUA
wvMxCLj9PBWu/z8kpP9BMpr/Q0GQ/0VPh/9HW33/TGZ0/VJva/lYdmT2XXte9GN/WvNqglbxcoRU
8XqFUvGEhVDxkIVP8Z+FTvGqhE7xqoRO8aqETvGqhE7xqoRO8aqETvGqhE7xqoRO8aqETvGq
hP+bCAD/oRMA2bgFAM7PBwDD8SABuPwyCq7/OBak/zojmf89L47/PzyE/0Fle/9FUnP/Svtr/1BiZf
9WZ1//XGtb/WJvV/xocVX7bnJS+3VzUPp+dE/6h3RO+pR0TfqfdE36n3RN+p90TfqfdE36n3RN+p9
0TfqfdE36n3RN+p90TfqfdPmfBQDdrQIAzsEFAMPZBwC4/CUDrv8vDKP/MxaY/zUgjf83K4L/OjZ4/
z5AcP9BSGj/Rk9i/0xUXf9TWFv/WvT/19eVP9lX1H/amFQ/3BiTv93Y0z/f2NL/4ljSv+TY0r/k2NK/5
NjSv+TY0r/k2NK/5NjSv+TY0r/k2NK/5NjSv+TY+ijAADPtwlAw8wEALj8EwCt/yQFov8qDJb/LBSL/y
8egP8yJ3b/Ni9t/zk3Zf8+PV7/QkJZ/0dGVf9NSVL/U0xQ/1INTv9fT0z/ZFBL/2lRSv9vUkj/dVJH/35S
Rv+GU0b/hlNG/4ZTRv+GU0b/hlNG/4ZTRv+GU0b/hlNG/4ZTRv+GU9KuAADEwQMAuNgCAK3/F
wGh/x8Flf8jDln/JhN+/ykac/8slmn/MClh/zQvWv85M1X/PjdR/0M6Tf9HPEv/TD5J/1E/R/9WQEb/WO
FE/2BCQ/9lQkL/akNB/3FDQP93Q0D/d0NA/3dDQP93Q0D/d0NA/3dDQP93Q0D/d0NA/3dDQP93
Q8W5AQC5zgAARPEEAKH/EwKU/xgFh/8bCnv/HxFw/yIXZv8mHV3/KiJW/y8nUf80Kk3/OC1J/z0v
Rv9BMET/RTJC/0kzQf9NMz//UTQ+/1U1Pf9ZNTz/XTY6/2M2Of9oNjn/aDY5/2g2Of9oNjn/aDY5/2g
2Of9oNjn/aDY5/2g2Of9oNrvFAAct2QAAoP8EAJP/DAKG/xAFef8TCW3/Fw5i/xsUWf8gGFL/JR1N
/ykgSP8ulkX/MiRC/zYmP/85Jz7/PCg8/z8oOv9DKTn/Rio3/0kqNv9MKjX/UCs0/1QrMv9YKzL/WCs
y/1grMv9YKzL/WCs/1grMv9YKzL/WCs/1grMv9YK/98EQP/eRsl/3MID/9qMRj/bzke/3RCJv92TS//
d1k7/3JoRv9tdlH/alVa/2WTYP9joWX/Yqxo/2G2av9hwWz/Yc1t/2HZbf9h5W79Y0tv917tb/Bd72/q
W/Fw4ljzcN5X9G/eV/Rv3lf0b95X9G/eV/Rv3lf0b95X9G/eV/Rv3lf0b/98EQP/eRsl/3MID/9qMRj/bzke

/3RCJv92TS//d1k7/3JoRv9tdIH/aIva/2WTYP9joWX/Yqxo/2G2av9hwWz/Yc1t/2HZbf9h5W79Y0tv
917tb/Bd72/qW/Fw4ljzcN5X9G/eV/Rv3lf0b95X9G/eV/Rv3lf0b95X9G/eV/Rv3lf0b/98EQP/eRsl/3MI
D/9qMRj/bzke/3RCJv92TS//d1k7/3JoRv9tdIH/aIva/2WTYP9joWX/Yqxo/2G2av9hwWz/Yc1t/2HZbf
9h5W79Y0tv917tb/Bd72/qW/Fw4ljzcN5X9G/eV/Rv3lf0b95X9G/eV/Rv3lf0b95X9G/eV/Rv3lf0b/98E
QP/eRsl/3MID/9qMRj/bzke/3RCJv92TS//d1k7/3JoRv9tdIH/aIva/2WTYP9joWX/Yqxo/2G2av9hwW
z/Yc1t/2HZbf9h5W79Y0tv917tb/Bd72/qW/Fw4ljzcN5X9G/eV/Rv3lf0b95X9G/eV/Rv3lf0b95X9G/eV
/Rv3lf0b/99EQP/ehsl/3QID/9rMRj/czgd/3hCJf96TC//e1c6/3ZmRv9xc1L/bYFc/2iQY/9mnmn/ZKlt/2
Ozb/9ivnH/Ysly/2LWc/5i43T5Yeh08l/rdetd7XXkWu923FfxdthZ9HPYWFfRz2Fn0c9hZ9HPYWFfRz2F
n0c9hZ9HPYWFfRz2Fn0c/9+EQP/exsH/3UkDv9uLxf/eDcc/35Al/+ASi3/gIU5/31iRv94b1P/c3xf/26K
aP9qmW//Z6R0/2Wud/5kuXn9ZMR7/GTRfPlk333yYuV+61/nf+Jd6n/aWeyA01rwm9c83nPXPn5z
1zzec9c83nPXPn5z1zzec9c83nPXPn5z1zzef9/EQL/fBsH/3YkDv9yLhb/fDYa/4l/lv+GSSz/h1M4/4
RgRv9+bFT/eXhh/3OFbP5uknT8ap96+mepf/lmtlL3Zb+E9mXMhfRI3lbsYuGI4l/kiNlb54nRW+qHyl
3ugcZe8H3GXvB9xl7wfcZe8H3GXvB9xl7wfcZe8H3GXvB9xl7wff+AEQL/fRsH/3gkDv92LhX/gDYZ
/4c/IP+KSCr/jVE3/4ldRf+EAvt+fnVi/HiBbvlyjXn3bZmA9WmkhVnNr4rxZrqM8GXHju1115DIYt6R2l/i
ktBc5JHJX+eMw2Hrhr9i7YK/Yu2Cv2Ltgr9i7YK/Yu2Cv2Ltgr9i7YK/Yu2Cv2Ltgv+AEAL/fhsH/3kkDv
96LRT/hDUY/4s+H/+PRyn/kIA1/49cRP6LZ1P7hXFj9359cfR3iH3xcZSG7mufjuxoqZLpZrSW52XB
mOVI0ZncY9yb0F/em8hg4ZfBYuSRu2Toirhl64e4ZeuHuGXrh7hl64e4ZeuHuGXrh7hl64e4ZeuHu
GXrh/+BEAL/fxoG/3okDf99LRL/iDUW/489Hf+TRif/l08z/pVaQvqRZFP2jG9j8oR5c+59g4HqdY6M5
m2YleJpo5vfZq2f3Ge7n9ppzJ/UadyfyGXcn79k3py6ZuGWtGjIj7Fp6luxaeiLsWnoi7Fp6luxaeiLsWn
oi7Fp6luxaeiLsWnoi/+CEAL/gBoG/3skDf+ALBH/izQV/5I9G/+XRSX/m04y+5tYQfaYYILyk2xj7Yx1d
OiDf4TieYiS3HKSm9hunqDTbKmi0Gy1o81txKPLb9ejwGrao7do26Gyat6brWzjlKpt5Y+qbeWPqm
3lj6pt5Y+qbeWPqm3lj6pt5Y+qbeWPqm3lj/+DEAL/gBoG/3wkDf+DLBD/jjQT/5Y8Gv+bRSP8oE0w
96FWP/OfYFDummpj6JNydeGJeobagYSS03qOnM10mKHlCqOlXHGupsJyvKe/c86nuHDXp69t2K
arb9yfpnDgmKRx45SkceOUpHHjIKRx45SkceOUpHHjIKRx45SkceOUpHHjIP+DEAL/gRoG/30kDP
+GLA//kTQS/5k8GP+fRCH5pEwu9KdUPe+IXU/pomdi4ppvdduSelTSiYGSyoGKnMN7k6O+d52nu
napqbd2taq0d8WqsHbVqqhx1aqjc9mjn3TenJ114JeddeCXnXXgl5114JeddeCXnXXgl5114JeddeC
XnXXgl/+EEAL/ghoF/34kDP+JKw7/IDMR/507F/6jQx/3qEss8a1SO+usW03lqGRh3aFtc9SZdoPLk
H6RwoiGnLuBj6S1fJipsHqjrK16r62qe76tqHzSraB3062cd9anmXjcoJd43puXeN6bl3jem5d43puXe
N6bl3jem5d43puXeN6bl3jem/+EDwH/gxoF/34jDP+LKw3/lzMQ/6A7FfynQh70rEoq7rFQOeizWEvh
rmFf2Kdqcs6ec4LFlnuQvl6Cm7OHlqStgpOqp3+erqR+qrChf7iwn4DKsJh80bCUe9SrXvao5B73J
6Qe9yekHvcnpB73J6Qe9yekHvcnpB73J6Qe9yekHvcnv+FDwH/gxoF/38jDP+NKwz/mTMO/6l6E/
mqQhzysEkn67dPN+S6V0nctF9e0qxocMikclC/nHiPtpR/mq2Nh6Slil+rn4SZsJqDpbKXg7KzIYTES
5CB0LKMftKuin7Ypol+26GJftuhiX7boYl+26GJftuhiX7boYl+26GJftuhiX7bof+FDwH/hBoF/4AjC/+Q
Kwv/nDIN/6U5EvetQRrvtUgl6L1ONOG/VEfYul1czbFmb8Opbn+5onWNsJt8maeUg6Oejourl4mUs
ZGHoLSOh621i4i+tYiHzrSDg9Gxg4LWqlKC2aOCgtmjgoLZo4KC2aOCgtmjgoLZo4KC2aOCgtmjg
oLZo/+GDgH/hBoF/4ljC/+SKgr/njEM/qg5EPSxQBjsukYi5cRMMd3FUKbSv1pbyLdjbr6ua360p3KM
q6F5mKGagKOYIlerj4+PsYiMm7WEjKi3gY24t3+OzbZ6iM+ze4fUq3uG2KZ7htime4bYpnuG2KZ7h
time4bYpnuG2KZ7htime4bYpv+GDgH/hRkF/4UjCv+UKgn/oTEL+6w4DvK2PxXpwEUf4sxKLtjMT0
TOxVhZw7xhbLm0aHyvrW+Lpad1l5uhfKKRm4Kqh5aLsX+TlrZ6kqS3d5Ozt3aUybdy860c43TrHS
L1qd0i9andlvWp3SL1qd0i9andlvWp3SL1qd0i9andlvWp/+HDgH/hhkE/4cjCf+XKQj/pDAJ+LA2DO
+7PRLmyEMb3tZELNPTTELly1ZXvsNearO7ZXqptGuJn65xlZWoeKCKon6pf56GsHebkrVxmQC3
b5uvt22dxLZpmM60a5TTrWyS1qhshtaobJLWqGyS1qhshtaobJLWqGyS1qhshtaobJLWqP+IDgH/

hxE/4oiCP+aKAf/qC4H9bQ1CuvCOw/h0kAW2N5BKs3aSkDD0INVuMtbaK3DYXijvWeHmbZtk46
xc56Cq3mneKeCrm+ljbNqpZy0Z6astGanwLNjpNCyZJ7UrGwa1qdlmtanZZrWp2Wa1qdlmtanZZr
Wp2Wa1qdlmtanZZrWp/+JDQH/iBkE/44iBv+eJwX+rCwF8boyB+XMNgrY3jsS0ORDKMfgST6821
BTsdRXZqfNXXacx2OEkcBokYa7bpx7tnSkcLN9q2mxia5ksZmwYrKqsGCzva9esdKuXqnWqV+l2
KVfpdiiX6XYpV+l2KVfpdiiX6XYpV+l2KVfpdiiX6XYpf+KDQH/iRkE/5lhBf+iJQP5sSkD68MsBNraLQ
XQ5T4Tx+tHJb/nSTu0401Qqt5TY5/ZWHOU0l6Bic1j1n7laZhzhHCgasF5pmPAhqlfwZaqXcKoqVvD
vKlaw9aoWbfZpFqx26BasdugWrHboFqx26BasdugWrHboFqx26BasdugWrHboP+LDQD/ixgD/5cf
A/+olghH0uSMB4NEfAdDjMAfH7T8Uv/BHJbbuSzis60tMoehPX5bkU2+L4Fh9gNtdiXXYY5Nr1Wua
Y9N1n17Tg6Fb1JSiWdWnolfXuqFW2dmgVMndnVXC35pVwt+aVcLfmlXC35pVwt+aVcLfmlXC35
pVwt+aVcLfmv+NDAD/jRgD/5wdAf6uHADpxRMA0uAZAMftMwm+9kAWtfZFJ6z0SDii80lKmPFK
W43uT2qB61J4delXg2vnXI1i5mSTXOZvl1fmfZhV5o2ZU+aemFLmr5hQ58WXT+HdlFDW4pJQ1u
KSUNbiklDW4pJQ1uKSUNbiklDW4pJQ1uKSUNbikv+PCwD/kBcC/6QXAOq4DADSzwgAyOshAb
72Ngu1/D4YrPxJ6L7RDaX+kVGjvIHVYP4TGJ591Fub/ZWd2b1XH5f8mSEWfBth1Xud4lS7oSKU
e6Tik/uoopO77KJTO/JiEv3odL796HS+/eh0vv3odL796HS+/eh0vv3odL796HS+/eh/+RCgD/mBM
A7awJANLBBgDJ1wgAvvYoA7X/NQ2r/zsZof89Jpf/QDSN/0JBg/9ETnn/SFlw/05iaP9VaWL9W29c
+2NzV/lrdIT4c3hR9355T/eLeU73mHIN+KZ5TPi3eEv4xnhL+MZ4S/jGeEv4xnhL+MZ4S/jGeEv4xnh
L+MZ4S/jGeP+UCAD5oQgA07UDAMjJBQC+9hIAtP8qBav/Mg6g/zYZlv84Jlv/OzCB/z47d/9BRW//
RU5n/0tVYf9SW1z/WV9Y/2BjVP9nZVH/b2dP/3hoTf+CaUz/jmlL/5ppSv+qaEn/tWhJ/7VoSf+1aEn/t
WhJ/7VoSf+1aEn/tWhJ/7VoSf+1aP+YBQDVrAAAYL4DAL7TBQC0/xsBqv8oB5//LQ+U/zAYiv8zIX/
/Nit1/zo0bP8+PGT/QkNe/0dlWP9NTVX/VFBS/1xST/9jVE3/aVZL/3FXSv95V0j/ghH/41YRv+bWEb
/pVdG/6VXRv+IV0b/pVdG/6VXRv+IV0b/pVdG/6VXRv+IV9qjAADKtQEAvskDALPjBQCp/xwCnv8jB
5P/Jw6H/yoWff8tHnL/MSZp/zUtYf85M1v/PjhV/0M8UP9IP03/TkFL/1RDSf9bRUf/YUZf/2hHRP9vR
0P/dkhC/39IQP+KSED/kkhA/5JIQP+SSED/kkhA/5JIQP+SSED/kkhA/5JIQP+SSMyuAAC/wAIA9
QAAKj/CwCc/xCKf8cB4X/IA16/yMTb/8nGmb/KyFd/y8mV/80KIH/OC5N/z0xSf9CM0b/RzVE/0w2Q
v9RN0D/Vjg//1w5Pf9iOTz/aDo7/286Of94Ojn/fjo5/346Of9+Ojn/fjo5/346Of9+Ojn/fjo5/346Of9+OsG
5AAC0zAAAp9wAAJv/CQCP/xADg/8VBnf/GQxs/x0RYv8gFln/JBtS/ykfTf8tI0j/MiVF/zYnQv86KT//P
io9/0MrO/9HLDr/Syw4/1AtNv9ULjX/WS40/18uMv9mLjH/ay8x/2svMf9rLzH/ay8x/2svMf9rLzH/ay8x/
2svMf9rL7bFAACp1gAAmusAAI7/AgCB/wcCdP8MBmn/EQpe/xUPVP8ZE07/Hhdl/yMaRP8nHED/
Kx49/y8fO/8zIDj/NiE2/zkiNP88lzP/QCMx/0MkMP9HJC7/SyQt/08IK/9UJSr/WCUq/1glKv9YJSr/WC
Uq/1glKv9YJSr/WCUq/1glKv9YJf9wEQX/axwK/2MnEf9aMhj/Yjgd/2ZBJP9mTCz/ZVkl/2JpP/9ed0f
/W4VN/1mTUf9YoFT/WKtW/1e0WP9Xv1j/WMpZ/1jVWv9Y4Vr/Wota/1n0W/tZ+Vv2WPtb8Fb8W+I
U/VzoVf9a6FX/WuhV/1roVf9a6FX/WuhV/1roVf9a6FX/Wv9wEQX/axwK/2MnEf9aMhj/Yjgd/2ZBJP
9mTCz/ZVkl/2JpP/9ed0f/W4VN/1mTUf9YoFT/WKtW/1e0WP9Xv1j/WMpZ/1jVWv9Y4Vr/Wota/1n
0W/tZ+Vv2WPtb8Fb8W+IU/VzoVf9a6FX/WuhV/1roVf9a6FX/WuhV/1roVf9a6FX/Wv9wEQX/axwK/
2MnEf9aMhj/Yjgd/2ZBJP9mTCz/ZVkl/2JpP/9ed0f/W4VN/1mTUf9YoFT/WKtW/1e0WP9Xv1j/WMp
Z/1jVWv9Y4Vr/Wota/1n0W/tZ+Vv2WPtb8Fb8W+IU/VzoVf9a6FX/WuhV/1roVf9a6FX/WuhV/1roVf
9a6FX/Wv9wEQT/bBwJ/2QmEP9dMRj/ZTYc/2pAl/9rSyv/alY1/2dmP/9idUj/XoNP/1yRVP9bnlj/Wql
a/1myXP9ZvF3/Wche/1nSXv9a31/+Wupf/Fr0YPda92DxWPlg6lb6YeVW/F/jVv5e41b+XuNW/I7jVv
5e41b+XuNW/I7jVv5e41b+Xv9xEQT/bRsJ/2UmEP9iLxf/azUb/3A/lf9xSSr/cVQ1/25jP/9pcUr/ZH9S/
2CMWf9emV7/XaVh/1yuY/9cuGX/XMNm/lzOZvxc3Gf5XOlo9lzyaO9a9GnoWPVp4lb3adxY+2Xa
Wfxj2ln8Y9pZ/GPaWfxj2ln8Y9pZ/GPaWfxj2ln8Y/9yEAT/bhsJ/2clEP9mLRX/cDQZ/3U+IP93Ryn/d
1l0/3VfP/9vbUv/antV/2WIXf9ilWP/YKFn/V6qavxetGz7Xb9t+V3Kbvdd2W/0XeZw8F3uceda8XHfV/

Jy2ln2btNb+mnRW/to0Vv7aNFb+2jRW/to0Vv7aNFb+2jRW/to0Vv7aP9zEAT/bxoJ/2glEP9rLBT/d
DQY/3o9H/98Rif/fVAy/3tdP/91akv/b3dX/mqDYfxlkGj6Ypxt+GCmcfZfsHP1Xrp19F7Gd/Je1HjvXuR
56F3red9Z7nrXWfF40Vv1c8td+W7JXvpsyV76bMle+mzJXvpsyV76bMle+mzJXvpsyV76bP90EAT/
cBol/2kkD/9vKxP/eTMX/388Hf+CRSb/g04x/4JaPv98Zkv8dnNY+XB/ZPdqi230ZZd08mKhefBgq3z
uX7V+7F/BgOpfz4HnX+CD31zohNRa7IPOXPB+yF7zecJg9nPAYPdxwGD3ccBg93HAYPdxwGD
3ccBg93HAYPdxwGD3cf91DwP/cRol/2skD/9zKhL/fjIV/4Q7G/+HRCT/iU0v/4hXPPuDY0v4fW9Z9
HZ6Z/BvhXLtaZB76mSbgedhpYXIYK+l4l+7i+BfyYzdYNyM1V/ni8td6YnEYO2Dv2Hwfrpj83i4Y/V2u
GP1drhj9Xa4Y/V2uGP1drhj9Xa4Y/V2uGP1dv92DgP/chkl/2wkD/93KhD/gjIT/4k6Gf+NQyL/j0st+49
VOveLYUrzhWxa7n12aep1gHblblqB4GaUit1jn47ZY6qQ1mS2kNRlxJHSZtaQzGXlkMFh5Y+7ZO
mJt2Xtg7Jn8H2wZ/J7sGfye7Bn8nuwZ/J7sGfye7Bn8nuwZ/J7sGfye/93DgP/cxkl/20jDv97KQ//hjES/
405F/+RQh/9IEor+JZTOPOSXkjujWIZ6lVyauJ7e3ncc4WF1m2PjNFqmpHNaKWUymivlchpvJXGa
s2Vwmrhlbhm4pWzaOaPr2npiapr7YKpa++AqWvvgKlr74Cpa++AqWvvgKlr74Cpa++AqWvvgP94
DgP/dBkH/28jDv9/KQ7/ijEQ/5E5Ff+WQR36mkko9JxRNU6aXEbplGZY4oxvatqDd3nTe4GFzHSKj
sdwlJPDbp+Xv22qmb1utpm6bsWZuG/ZmbBs4JmrbOOUp23mjaNu64aib+yEom/shKJv7ISib+yEo
m/shKJv7ISib+yEom/shP94DgP/dRgH/3liDf+CKA3/jTAP/5U4E/6aQBv3nkgm8KJQM+qhWUTkm2
RW3JNsaNOLdXjLgn2ExHuGjr52j5W5c5mZtXKknLJysJ2wc76drnTQnaly3Z2jcOCZn3Lkpxy6lub
cuqlm3LqiJty6oibcuqlm3LqiJty6oibcuqlm3LqiP95DgP/dhgH/3UhDP+FKAz/kC8N/5g3EvufPxnzo0
cj7adOMeanV0HfoWfV1ppqZ82RcnbEiXqDvIKCjrZ9ipaweZSbrHefn6l2qqCmd7ehpHjJoKF326Cb
dd2dmHXhlpV15o+UdeeMIHXnjJR154yUdeeMIHXnjJR154yUdeeMIHXnjP96DQL/dxgH/3chC/+H
KAv/ky8M/5w2EPmjPhfxqEYg6axNLuKtVj/ap19T0KBoZceXcHW+kHeCtol+ja+DhpapfpCdpHyaoa
B7paOde7Gkm3zCpJl92KOSeNuhkHjfm0545JKNeOWPjXjlj4145Y+NeOWPjXjlj4145Y+NeOWPjX
jlj/96DQL/dxgG/3ohCv+KJwn/li4L/582DvamPRTurEQe5rJLK96yUz3VrFxRy6VmY8KdbXO4IXWB
sl97jaiJg5aihIudnlGVopd/oKWUfqymkX+8po+A0aaKfNiliHzdnYd74pWGe+OThnvjk4Z745OGe+O
Thnvjk4Z745OGe+OThnvjk/97DQL/eBgG/3wgCf+MJwj/mC4J/al1DPOqPBLrsUMb47hKJ9q4UDv
QsVpPxqljYr2ia3Kzm3J/q5V5jKKPf5abioeellaQpl6Dm6eKggqiph4O3qYWEy6iBgdaogIDbolB/4Jh/f
+KVf3/ilX9/4pV/f+KVf3/ilX9/4pV/f+KVf3/ilf98DQL/eRgG/38gCP+Pjgf/my0l+6UzC/CuOhDot0EY4L
9GJdW9TjrLtVhOwa5hYLioaXCuoW9+ppt2ip2VfJWUkIOejluMplWllqiBh6Oqfoiyq3yJxap4h9Wpel
Xao3iE35t4g+CYelPgmHiD4Jh4g+CYelPgmHiD4Jh4g+CYelPgmP98DQL/ehgG/4lgCP+RJgb/niw
H+KkyCe2zOQ3lvT8U28dCI9HCTDjHu1ZMvbReXrOtZm6pp2x8oKFziZeceZSOloCdhJKHpH2Pkg
l4jp+rdl6uq3KPwKtwj9WqclvZpHGJ3pxxiN+ZcYjfmXGI35lxiN+ZcYjfmXGI35lxiN+ZcYjfmf99DQL/e
xcG/4UfBv+UJQX/oioF9a0wBuq5NgrhxjwP1s0+lczlSjbCwVNkt7pcXK60Y2ykrWI7mqhvh5GjdZK
HnnucfZmDo3WWjqhvlZurbJaqq2qXvapomNWpaJPZpWmQ3Z1qjt+aao7fmmqO35pqjt+aao7fmm
qO35pqjt+aao7fmv9+DQL/fBcF/4geBf+YJAT+pSgD8bltBObAMQbc0jMM0dM7H8fORzS8x1BlssF
YWqi7X2qetWV4lBrhYqrcZCApneadqN/oW2giqZon5eoZaCnqWOhuahhotKnYJ3Zo2KY3pxjlt+a
Y5bfmmOW35pjlt+aY5bfmmOW35pjlt+aY5bfmv9/DAH/fRcF/4wdBP+clgL6qiUC7LknAuDLKAPV3
SwKy9s5HcHWRDGD20E1FrMpVV6LEXGeXvmF2jblngoO0bl14sHOWb617nWerhqJiq5SkX6ulpF2
st6Ncrc+iW6jboFyi35pcoOGXXKDhl1yg4ZdcoOGXXKDhl1yg4ZdcoOGXXKDhl/+ADAH/fxcF/5AcA
/+gHwH1sB8A5cMcANbcGQDM4zllw+l8GrreQi6w2UpCpdRRVJvPV2SQyV1yhcRifnu/allxvG6Sa
Lp3mGG5gpxduZKeWrmjnli6tZ1Xu82cVrbfmlau4pVXrOOTV6zjk1es45NXrOOTV6zjk1es45NXrO
OTV6zjk/+CCwH/ghYE/5UaAf+mGQDuuBIA1c8JAM3IHwHE6jYKu+pCF7LnRCuo40c+nd9NUJPb
UmCl1ldufdFcenLOYoRpy2mMYclzkVvJf5RYyo+WVsqliVTLs5VTzMyUUcjklK95o5SuueMUrrnjF
K654xSuueMUrrnjFK654xSuueMUrrnjP+ECwH/iBUC/5sVAPCtDQDUwAcAzdUJAMPsJgK78DgM

svBAGanvRCmf7EY6lepJS4nnTVt+5FJpc+FWdGnfXH1h3WSEWt1uiFbdfItT3oyMUd+ei1DgslpP
4caJTd/liE3S6oVOzeyDTs3sg07N7INOzeyDTs3sg07N7INOzeyDTs3sg/+GCgD/jxIA96MLANW1B
QDLxwYAw+kNALr0KwSy9zcOqfc9G5/2QCmU9UM4ivNFRoDySVR18E5ga+9TamLuWXJb7WF3
Vu1re1Lud35P7oZ+Tu+XfkzwwqH5L8Lt9SvHTfEnq6XpK5ex5SuXseUrl7HIK5ex5SuXseUrl7HIK5ex
5SuXsef+JCQD/lwoA16sCAMy7BADCzwYAufcZALH7LQao/TUQnv45G5T9PCeK/D80f/xCQHb7R
Utt+0tVZfpRXV76WGNy+mBnVPpqaH7dGxO+4FuTPyQbkv8oG5K/bFtSf3EbEf94GtH/eprR/3qa
0f96mtH/eprR/3qa0f96mtH/eprR/3qa/+MBwDboQAAzblCAMLEBAC52AYAsP8fAaf/Kwid/zAQk/80
Goj/NyV+/zsvdP8+OWv/Qkji/0dJXf9NT1j/VVNU/11XUP9mWk7/b1tL/3lcSf+GXUj/IV1H/6NdR/+zX
EX/yFtF/9RbRf/UW0X/1FtF/9RbRf/UW0X/1FtF/9RbRf/UW+2XAADQqgAAw7sCALnOAwCu9wwA
pf8fA5v/JgiR/ysQhv8uGXz/MiFy/zYqaf86MmH/Pjhb/0M9VP9IQID/T0VN/1dHsv9fSUj/Z0tG/3BMRf
96TUP/hk1C/5NNQf+hTUH/skxB/7pMQf+6TEH/ukxB/7pMQf+6TEH/ukxB/7pMQf+6TNSjAADFtAA
AucYBAK7YAAQCK/xAAmf8bA4//lQIE/yUPef8pFm//LR1m/zEkXv81Klj/Oi9S/z4yTf9DNUR/SThH/086
RP9WO0L/XTxA/2U9Pv9tPj3/dz47/4E+Ov+MPjn/mz45/6E+Of+hPjn/oT45/6E+Of+hPjn/oT45/6E+
Of+hPseuAAC7vwAAr9EAAKLhAACX/w0BjP8VA4H/Ggh2/x4ObP8iFGL/Jhla/yoeU/8vI07/MyZJ/z
gpRf88K0L/QSw//0YuPf9MLzv/UjA5/1gwN/9fMTX/ZjEz/24yMv93MjH/gzlw/4cyMP+HMjD/hzlw/4cy
MP+HMjD/hzlw/4cyMP+HMr24AACwygAAo9kAAJX0AACK/wcBfv8NA3P/Egdo/xcmXv8bEVX/Hx
VO/yMZSf8oHET/LB9A/zAgPf80ljr/OCM4/zwkNf9AJTP/RSYx/0omL/9Pjy7/VScs/1snK/9iJyn/aigo/
24oKP9uKCj/bigo/24oKP9uKCj/bigo/24oKP9uKLLAAACI1AAAlt8AAIj5AAB7/wABb/8EA2T/CQZa/
w4KUP8TDkr/FxvFD/xwUP/8hFjv/JRg4/ykANf8sGzP/MBww/zMcLv82HSz/OR4r/z0eKf9CHif/Rh8l/0
sfJP9QHyl/Vh8i/1kflv9ZHyL/WR8i/1kflv9ZHyL/WR8i/1kflv9ZH/9iFAB/XB4M/1MpEv9PMhf/VDgb/1
dBlf9WTCj/VFkw/1JpN/9QeDz/ToZA/06UQ/9NoEX/TqpG/06zRv9OvUf/TsdH/0/SR/9P3Uf/T+hH/1
DxR/9Q+Uj9Uf9l+1H/SPZQ/0jxUP9G8VD/RvFQ/0bxUP9G8VD/RvFQ/0bxUP9G8VD/Rv9iFAB/XB
4M/1MpEv9PMhf/VDgb/1dBlf9WTCj/VFkw/1JpN/9QeDz/ToZA/06UQ/9NoEX/TqpG/06zRv9OvUf/T
sdH/0/SR/9P3Uf/T+hH/1DxR/9Q+Uj9Uf9l+1H/SPZQ/0jxUP9G8VD/RvFQ/0bxUP9G8VD/RvFQ/0b
xUP9G8VD/Rv9iFAB/XB4L/1MpEv9RMRf/Vzcb/1pAlf9aSyj/V1cw/1VnOP9Tdj7/UYRC/1CRRv9Qn
Uj/T6hJ/1CxSv9Qukr/UMVL/1DPS/9R3Ev/UedL/1LwS/1S+Ev7Uv9M91L/TPJR/0zuUv9J7IL/Se5S/
0nuUv9J7IL/Se5S/0nuUv9J7IL/Sf9kEwb/Xh0L/1UoEv9VLxb/XDQa/2A+IP9gSSf/XIQw/1xkOP9Yc0
D/VoBG/1SNSv9Tmk3/U6RO/1KuT/9St1D/U8FR/1PMUf9T2VH9VOVS+ITvUvhU+FL2Vf9T8FP/U
+tT/1HnVP9O51T/TudU/07nVP9O51T/TudU/07nVP9O51T/Tv9IEwb/Xx0L/1YoEf9aLRX/YjMZ/2Y8
H/9nRyb/ZVlv/2JhOf9eb0H/WnxJ/1iJTv9WIIH/VaFU/1WqVf9Vs1b+Vb5X/VXIWPtV1Vj4VuNZ9lbu
WfNW91rvVv1a6VT/WuRW/1bfV/9S31f/Ut9X/1LfV/9S31f/Ut9X/1LfV/9S31f/Uv9mEwb/YB0L/1cnEf
9eKxT/ZzIX/2w7Hf9tRSX/bFAv/2ldOf9kbEP/X3IL/1yFUv9aklb9WJ1a/FenXPpXsF35V7pe+FfEX/Z
X0WDzV+Bh8FjtYe1Y92LmVfpj4Vf9X9xY/1vWWf9X1lr/V9Za/1fWWv9X1lr/V9Za/1fWWv9X1lr/V/9
nEgX/YRwK/1knEf9jKhP/bDEW/3l6HP9zQyT/c00t/3FaOP9raEP/ZnVO/GGBVvpdjVz4W5hg9lqjY/
RZrGXyWbZn8VjAaPBZzWntWd1q6Vnra+VY9WzdWPhp11n8ZdJb/2HMXp9czFz/XMxc/1zMXp9
czFz/XMxc/1zMXp9czFz/XP9oEgX/YhwK/1smEP9oKBH/cjAU/3c5Gv96QiL/ekss/3lWN/1zZEP6bX
BP9md8WvNiiGLxXpNo7lyebOxap2/qWrBx6Fm7cudZyHPkWdh14FnodtpY8nXSW/dvzFz7a8de/
mbCX/9hwl//YcJf/2HCX/9hwl//YcJf/2HCX/9hwl//Yf9pEQX/ZBsK/18kD/9tJxD/dy8S/304GP+AQB//g
Ukp/YFTNfh8YEP0dWxQ8G53XexngmfpYY1v5V6XdeJboXjgWqt73Vq1fNtcw3zZXdR81F3mfM9c
8XzlXvV2wmD4cb1h+2y5Yv5nuWL+Zrli/ma5Yv5muWL+Zrli/ma5Yv5muWL+Zv9qEQX/ZRsJ/2Mj
Dv9yJw7/fC8R/4M3Ff+GPx3+h0gn+IhRM/OEXUHufWhQ6XVzX+RtfGvfZYZZ222GRfNdgnH/UX6a
A0WCxgc9gvYHMYcyByWLfgcVh7oG9YvF8uGP0d7Rk93GwZftssGX7bLBI+2ywZftssGX7bLBI+2

ywZftssGX7bP9rEAT/ZhoJ/2chDf92Jg3/gS4P/4g2E/+MPhr6jkYk9I9PMO6MWj/phWVP4n1uX9x0e
G3VbYJ30GmMfsxmloLIZaCFxWWqhsNltobAZsSGvmfWhrpm6la0Zu2Cr2fxKxo9HeoafhxqGn4c
Khp+HCoafhwqGn4cKhp+HCoafhwqGn4cP9sEAT/ZxoJ/2ofDP96Jgz/hS0N/4w1Ef6RPRf3IEUh8J
VOLeqTWDzjjWJN24VrXtN9dGzMdX13xnCGf8FskIW+apqlummlirhqr4u1ar2Ls2vOi7Br4ourauqlp
2vugqRt8XygbvV1oG71daBu9XWgbvV1oG71daBu9XWgbvV1oG71df9tEAT/aBoJ/24eC/9+JQR/iS
wL/5A0D/uWPBXzmUQe7JxMKuWaVzndIGBL1lpxXMyEcWvEfXl3vneCgLhyi4e0cJWLsW6fjq5uq
o+rb7aPqW/Gj6dw24+ibuaOn3Drh5xw7oCZcPJ5mXDzeZlw83mZcPN5mXDzeZlw83mZcPN5mX
Dzef9tDwT/aRkl/3EeCv+BJQn/jCsK/5QzDfiaOxLwn0lb6KJKJ+GhVDbYml5Jz5JnW8aLb2q+g3Z2t
31+gLF4hoisdZCNqHOakaVzpZKic7CToHO/k55005OZc+OTInPoi5Rz7IWS/B9knPwfZJz8H2Sc/
B9knPwfZJz8H2Sc/B9knPwff9uDgT/ahkl/3QeCf+EJAj/jysl/5gyC/WfORDspEEY5KhII9ymUjTSoFx
HyZhlWcCRbWi4inR1sIN7gKp+goikeouPoHiV5x3oJWZd6yWl3e6l5R4zZaRd+GWjnbklx26YmL
du6Bi3bugYt27oGLdu6Bi3bugYt27oGLdu6Bi3bug9vDgT/axkl/3cdCP+Glwf/kioH/JsxCfKjOA3pqT
8U4a5GH9irTzLOpFpFxJ1jV7uWa2azkHF0q4l4f6SEf4idgleQmH2QlZN7m5iQeqeZjnu1mot7x5mJ
fN+Zhnrk4V554yEeeyEhHnshIR57ISEeeyEhHnshIR57ISEeeyEhHnshP9vDgP/axgl/3odB/+Jlwb/
SkG+p8vB++nNgvmrj0R3bNCHdOvTTHJqVhEwKJgVbecaGWuW9ypo91fp6KfliXhYOQkIKMlot/
5qHf6OchH+wnlJ/wpyAgNqcf7hl3195Y99fOqHfXzqh3186od9fOqHfXzqh3186od9fOqHfXzqh/9w
DgP/bBgl/3wcBv+MlgX/mCgE96ltBuyrNAjttDoN2bg/HM+0Sy/FrVVCu6deU7KhZmOpm21xoZVzfZ
iQeYeQi3+QiYell4OFkpt+hJ+ee4SsnmEvZ53hdSddYPfmXaC5JJ2gOmJdoDpiXaA6Yl2gOmJdo
DpiXaA6Yl2gOmJdoDpif9xDQP/bRgH/38cBf+PIQT/myYD9KYrBOiwMABfuzYJ1L08Gsq4SS3AsIN
At6xcUa2nY2GkoWpvm5xwe5OWdoaKkXyPgo2El3uLjpx2ipqfcoqon3CKuZ9ui9CebYnem26H45N
uheilBoXoi26F6ltuheilBoXoi26F6ltuheilBoXoi/9yDQP/bhcH/4lbBP+SIAP+nyMC8KonAuS2KwPa
xC0l0MI6GMW9Riu7uFE9srJZT6itYF+fqGdtlqJteY2dcoSEmXmOe5WAlnOSipttkZeeapGln2eStp5
mk8yeZJHem2aO4pRni+eMZ4vnjGeL54xni+eMZ4vnjGeL54xni+eMZ4vnjP9zDAP/cRYG/4YaA/+V
HgH6oyAB67AhAN++lgHVyicGysg3FsDERCm2vk47rLIWTKO0XVYzr2NqkKlpd4alb4J9oXWLC51
8k2ybhlmpmpOcYpqnWCbs5xfnMmbXZvfmV+W45NgkueMYJLojGCS6lXgkuiMYJLojGCS6lXgkui
MYJLojP90DAL/dhQF/4oZAv+ZGwD1qBoA5rcWANnMCwDP0SMFxm80FLrLQCaxxko4p8FSSZ2
8WVmTt2BnibJldlCua352qnGlb4d4kGWmg5VgpZCYXKWgmFqmsJdZp8aWWKbilVmh5ZBanOm
JWpvpivq6Ylam+mJWpvpivq6Ylam+mJWpvpif91CwL/exME/44WAF+eFQDurg4A1r8IANDRCA
DH2CADvtgwEbTUPSOqz0c1oMpORpbFVVaMwVtkgrxhcHi4ZnputWyDZrN0il+xf49asY2RWLkd
kVayr5FVs8SQU7PgjlSt6lpUp+yFVKfshVSn7IVUp+yFVKfshVSn7IVUp+yFVKfshf93CgL/gRIC/5M
SAPCkCwDVtAYAZsQGAMfWCQC/4CYCtuE0Dq3eOh+j2kMxmdZKQo/RUFGEzVZfeslca3DFYX
Vnw2h9X8Fxxg1nAfldWwluU8GciVLCrYhRwsOHT8Lghk+97YNQtPB+ULTwflC08H5QtPB+ULTwf
IC08H5QtPB+ULTwfv95CgL/hw8B+5oLANaQBADNuQUAxcGAL7kEAC26CwEruk5DaboPxab5U
EskeJGPYbeS0x821FZcdlWZWjVXG5f1GR1WdNtelTSeX1R04h+T9Safk7VrH1N1cJ8S9Xfe0vR8
HIMxvV1TMb1dUzG9XVMxvV1TMb1dUzG9XVMxvV1TMb1df98CQH/jgkA2qEAAM6wAwDFwAQ
AvdEHALXsGgCt8C4GpfE3D5vwPByS7z8qh+1COH3rR0Zy6UxSaOdRXV/mV2VY5l9rU+Vob0/m
dHFM5oNyS+eUckropnFJ6LhwR+nRb0fn7W5l3fZrSN32a0jd9mtl3fZrSN32a0jd9mtl3fZrSN32a/+B
BwDplwAA0agAAMa3AgC9yAQAs9kHAKz1IAGk+C0ImvkzEZH4NxyH+DsofPc/M3L2Qz5p9UdIYf
VOUFv0VVZV9F1bUfRmX031cWFL9X5iSfaOYkj2n2JH969hRvfDYEX3219E9fZeRPX3XKT1915E
9fdeRPX3XKT1915E9fdeRPX3Xv+LAADV0AAAYLAAAL3AAgCz0AQAqvkPAKL9lQOZ/ykjj/8vEY
X/Mxt7/zclcf87Lmj/PzZg/0Q9Wf9JQ1T/UUhQ/1lTf9lTkr/bFBI/3dRRv+EUKT/k1JE/6NRQ/+0UUL/x
VBB/+NPQf/jT0H/409B/+NPQf/jT0H/409B/+NPQf/jT9yXAADKqQAav7kBALTJAQCp2AIAoP8TAJf

/HgSN/yUJg/8pEXn/Lhlv/zlgZv82KF7/Oy5Y/0AzUv9FN03/SztK/1M9R/9bP0T/Y0FC/21CQf94Qj//h
UM+/5NCPf+hQj3/sEE8/8hBPP/JQTz/yUE8/8IBPP/JQTz/yUE8/8IBPP/JQc6jAADAswAAtcMAAKn
SAACd5gEAIP8RAYr/GQSA/x8Jdf8kD2z/KBZi/ywcW/8wlVT/NSZP/zoqSv8/LEb/RS9D/0sxQP9S
Mj7/WTM8/2E0Ov9qNTj/dTU2/4A1Nf+MNTT/mjU0/6o0NP+rNDT/qzQ0/6s0NP+rNDT/qzQ0/6s0N
P+rNMKtAAC3vQAAq80AAJ3aAACR9gAAh/8LAXz/EwRy/xglaP8dDI7/IRNW/yYXUP8qG0r/Lh9G/
zMhQv83lz7/PCU7/0EmOf9GJzb/TCg0/1MpMf9bKi//Yyot/2wqLP92Kir/gCop/48qKf+PKin/jyop/48q
Kf+PKin/jyop/48qKf+PKrm4AACsyAAAn9UAAJDfAAACE+wAAeP8EAW7/CwRj/xAIWv8VDFH/GhB
L/x4TRf8jFkD/Jxg8/ysaOf8uHDb/Mh0y/zYeMP86Hy3/Px8r/0QgKP9KICb/USEk/1ghlv9glSH/aCEf/
3MhH/9zIR//cyEf/3MhH/9zIR//cyEf/3MhH/9zla/EAACh0QAAkdwAAILkAAB1/gAAav8AAV//AgRV/w
cHTf8MCKX/EQ0//xUPOv8aETb/HhMz/yIVMP8mFi3/KRYr/ywXKP8wGCB/Mxgk/zcZlv88GSD/QR
ke/0YaHP9MGhv/UhoZ/1oaGf9aGhn/WhoZ/1oaGf9aGhn/WhoZ/1oaGf9aGv9UFwj/TCEN/0MrE/9
DMRb/Rzca/0hAH/9HTCX/Rloq/0RpL/9DeDL/QoY1/0KUNv9Dnzf/Q6k3/0SyN/9Euzf/RcQ3/0XPN/
9F2jf/RuU3/0buN/9H9jf/SP02/Uj/Nvxl/zb6SP82+Un/NfIJ/zX5Sf81+Un/NfIJ/zX5Sf81+Un/Nf9UFwj/
TSEN/0MrE/9FMBb/SjYZ/0s+H/9KSiT/SFgq/0ZnMP9FdjT/RIU2/0SSOP9EnTn/Rac5/0WwOf9GuT
n/RsM5/0fNOF9H2Dn/SOQ5/0jOf9J9jn9Sf05+0r/OfpK/zn4Sv8490v/N/dL/zf3S/8390v/N/dL/zf3S/83
90v/N/9VFwj/TiAN/0UrEv9JLxX/TjQZ/1A9Hv9QSCT/TIYr/0tkMf9Jczb/SIE5/0iPO/9Imj3/SKQ9/0itP
v9Jtj7/ScA+/0rKPv9K1T7/SuE+/kvsPvtL9T75TP0+90z/PvVM/z70Tf888U7/O/FO/zvxTv878U7/O/F
O/zvxTv878U7/O/9WFgj/TyAN/0YqEv9NLRT/UjMY/1Y8Hf9VRyT/U1Mr/1BhMv9OcDj/TH48/0uLP/
9LI0H/S6FC/0uqQv9Ls0P/TLxD/0zGQ/5M0kP8Td9E+k3qRPd09ET1TvxE807/RPFP/0PuUP9A61
H/PutR/z7rUf8+61H/PutR/z7rUf8+61H/Pv9XFgf/UCAM/0gpEv9RLBP/WDEX/1w7HP9bRSP/WVAr
/1ZeMv9TbTn/UXo//0+HQ/9Ok0X/Tp5H/06nSP5OsEj9TrIJ+07DSfpPzkr4T9xK9U/pSvJQ80rvUPx
L7VD/SutS/0foU/9F5FP/Q+RT/0PkU/9D5FP/Q+RT/0PkU/9D5FP/Q/9YFQf/UR8M/0wnEf9WKRL/X
jAV/2I5G/9jQyL/YE4q/11bM/9aaTv/VnZC/1SDR/1Sj0r7UZpN+IGjTvhRrE/3UbVQ9VG/UfRRy1HyU
dIS71LnUuxS8IPpUftU5IP/UORV/03fv9K21f/R9tX/0fbV/9H21f/R9tX/0fbV/9H21f/R/9aFQf/Ux8M/1
AmEP9bJxH/ZS4U/2k3Gf9qQSD/aUsp/2ZXmV9hZTz9XHJF+II+S/dWiLD1VZVU81SfVvFTqFjwU7F
Z7IO7WuxTxlvR9U9Vc51PkXONT8V3gVPpb3Vf/VtIZ/1PUWv9P0Fr/TdBa/03QWv9N0Fr/TdBa/03Q
Wv9N0Fr/Tf9bFAf/VB4M/1UkD/9hJg//ay0S/3A2F/9yPx7/cUkm/25TMftqYTz3ZG1H8195T/BbhFbt
WI9b61aaX+hVo2HmVaxj5FS2ZOJUwmbgVNBn3VXiZ9IV72fVWPpi0lv/Xc1c/1nIXf9VxV7/UsVe/1
LFXv9SxV7/UsVe/1LFXv9SxV7/Uv9cFAB/Vh4L/1oiDv9nJQ7/cSwQ/3c0FP95PRv/eUYk+ndQL/Vz
XTvwbGIH7GV0U+hgf1zkW4lj4ViTaN5XnWvbV6ds2VixbdZZvW3UWcxt0Vrdbc1Z7G3JXPIpxl7/Z
MFg/1+9YP9bumH/WLph/1i6Yf9YumH/WLph/1i6Yf9YumH/WP9dEwb/Vx0L/14fDP9sJAz/dysO/30
zEv+AOxj8gEQh9X9OLO98WTnqdWVH5G1vVd5leWHZYRrP1V6ObdFdmHDPXKJxzFyscspdtN
LHXsRyxV7UcsFe5nK+X/RwumL8arZj/2WzZP9gsGT/XbBk/12wZP9dsGT/XbBk/12wZP9dsGT/Xf
9eEwb/WB0L/2IeC/9xIwv/fCoM/4IyD/+GOhX3iEMe8IdLKeqEVzbjFWFG3HvRvDvudWHPaH5qym
SlcMZiknTDYZx2wGGmd75hsHi8Yrx4umPMeLdj33izY+93sGb4cK1n+2upaP5mp2n/Yqdp/2Knaf9
ip2n/Yqdp/2Knaf9ip2n/Yv9fEgb/WRwL/2YcCv92Iwn/gCkK/4cxDfuMOBLzjkEa649JJJeSMVTPdhI9
D1H5oU812cWDGcHpqwWuDcrxojHe5Z5Z6tmage7NmqnYxZ7Z9r2fEfa1o2H2pZ+p8p2r1dqRs+H
ChbPtrn2z+Z59s/mefbP5nn2z+Z59s/mefbP5nn2z+Z/9gEgb/WhwK/2obCf95Igj/hCgl/4wvC/iRNw/v
ID8X55ZIld+TUy/WjVxBzoVIUcZ+bl+/d3ZquHJ+c7Nuh3mwbJF9rGuaf6prpYCna7CBpWu+gaNsz4
GgbOSBnm7xe5tv9XWZb/hvl277a5du+2uXbvtrl277a5du+2uXbvtrl277a/9hEgX/WxwK/24bCP99I
Qf/iCcH/pAuCPSWNQzrmj0T451GHdqZUC3Rk1o/yItjT7+Ea164fnNpsXh6c6x0gnqncYt/pHCVgqF
volSeb6uFnHC4hZpwyYWXcN+FIXHugJNx8nqRcfZ0j3H4b49x+G+Pcfhv3H4b49x+G+Pcfhv3H4

b/9iEQX/XhoJ/3EaB/+AlAX/iyYF+5QsB/GbNArnoDsP36NDGdWfTSvMmFk9w5FhTbqKaVyyhHB
oq393cqV6fnqgd4eAnHWQhZhzm4eVc6aJk3OziZFzw4mOdNqJjXTshYt08H6JdPR3iHT2c4h09n
OldPZziHT2c4h09nOldPZziHT2c/9jEQX/YRkJ/3QaBv+DHwT/jyUE+JgrBe2fMQfktgM26c/F9GjS
ynHnVY7vpZgS7WQZ1qtim5npov0cp+Ae3qZfIOBIHmMhpB3loqNdqKMinaujIh3voyGd9SMhHjpiY
N37oKcd/J7gXb0doF29HaBdvR2gXb0doF29HaBdvR2gXb0dv9jEQX/Yxcl/3cZBf+GHgP/kiMD9Zs
oA+qjLgXgqjUI1qw8FsynSifDoVQ5uptdSbGVZVipkGxmoypycZmGeHqTgX+CjX6Hilh8koyEe52O
gXqj397uo99e8+Pe3znjHt77IV7evB+enrzeXp683l6evN5enrzeXp683l6evN5enrzel9kEAX/ZhYH/
3oZBP+JHQL/ISEC8p8IAuaoKgPdsC8F0rA6FMirRyW+plI3taBbR62bY1aklWpknJBvb5SLdXmNh
3yChoOEilCBjY18f5mQeH+nkXZ/tpF0gMqRcoDjj3OA6odzfu6Ac33xe3N98XtzffF7c33xe3N98Xtzff
F7c33xe/9IEAX/aRQG/30YA/+MGwH9mB8B7qMhAeKtJAHYtigEzbQ4EsOwRSS6q1A1saVZRaig
YFSfm2dil5Ztbo+Rc3iHjXmBf4mAiHiGiY5zhZWRclWjkm2FspJrhSaRaYbgkWuF6Ylsg+2CblLwfW
yC8H1sgvB9bLwfWYc8H1sgvB9bLwff9mEAT/bBIF/4AXAv+PGQH5nBoA6qcaAN6zGQDTuiUDy
bg2Eb+1QyG1sE0yrKtWQ6OmXIKaomRgkp1qblmYcHaAlHaAeJB9h3GOho1rjJKRalygkmWMr5J
jjcKRYy3ckGOM6Ypkie2DZYjwfmWl8H5liPB+ZYjwfmWl8H5liPB+ZYjwfv9nDwT/cBEE/4QVAf+TF
gD1oBQA5a0OANe6CQDNvylCw74zD7q6QB+wtkswp7FTQJ6tW0+VqGFdjKRnaYOgbHR6nHJ9
cZh5hWqWg4tklY+PYJWdkF6VrZBclsCOW5bajVyV6ohdke2CXo/wfl6P8H5ej/B+Xo/wfl6P8H5ej/B
+Xo/wfv9oDgT/dRED/4gTAP+WEQDjpQkA1bEHAM+/BwDHxR4BvcUvDbTBPRyrvUctorIQPZi0V
0yPsF1ahqxjZn2oaXBzpW96a6J2gWOgf4den4yKW5+ai1igqotXoL6KVqHYiFaf7IVXm+9/WJjxe1i
Y8XtYmPF7WJjxe1iY8XtYmPF7WJjxe/9qDQT/eg4C/4wOAOycCADVqQUAzrUGAMfDBQDAzBg
At8wrCq7KORmlxkMpm8JMOZK9U0iluVIWf7VfYnWyZWxsrt2t1ZK1yfF2rfIFZq4mEVquYhFSsqYR
SrLyDUazWglGr7n9SpvJ6UqP0d1Kj9HdSo/R3UqP0d1Kj9HdSo/R3UqP0d/9sDAP/gAwA+pEIANi
hAgDOrgQAx7sEAL/JBgC41BIAsNUmCKfUNBae0D8lIMxHNYvITkSBxFRrd8FaXG6+YGZlvGdv
XrpvdVi5eXlUuYZ7UbmWfFC5p3tOurt6TbrVeU257ndOtPZzTrD4cE6w+HBOsPhwTrD4cE6w+HB
OsPhwTrD4cP9vCgL/hggA3JgAANCmAgDHswMAv8EDALfPBwCv3hQAqeArBaDfMxKW3Dshjdl
CMIPVST550k9Lb89VVmbNW19ey2JnWMprbFPJdm9PyoRxTcqVcUzLpnBLy7pvSszVbknK7m1J
xvtqSsH9aErB/WhKwf1oSsH9aErB/WhKwf1oSsH9aP94BgHzjQAA1J4AAMmtAQC/uglAt8gEAK7
VBwCn5x4AoOktBpjpNRCO6DschOY/KnrkRDdv4UpEZuBPTI7eVIZX3V5dUt1nYU7dc2RL3oFISd
6SZUjfo2RH4LZjRuDPYkXe6mFF3PpfRtf/XUbX/11G1/9dRtf/XUbX/11G1/9dRtf/Xf+BAADalgAAzK
YAAMG0AQC4wglArs8EAKXrDACE8SEClvlsCl3zMhGD8jccfe7J2/wQDFm70U7Xu5LQ1juUkpS
7lpPTu5jUkrubIVi7ntWRu+LVkXwnFVE8K1VQ/DAVELw2INB7/JSQez/UUHS/1FB7P9RQez/UUHS
/1FB7P9RQez/UeKMAADPnwAAw64AALm8AACvygEApNYDAJz3EwCU+iADi/soCYH8LhF4/Dla
bvs3l2X7PCtd+0EyVvpGOFH6TT1N+1ZASftfQ0b7aUVE+3RGQvyBRkH8kUZB/aJGQP2zRT/9xk
Q+/d5DPf3vQj3970l9/e9CPf3vQj3970l9/e9CPf3vQtWXAADFqAAAurYAALDFAACk0gAAmd0BAJ
H/EwGI/x0Efv8jCnX/KRBr/y0YY/8yHlv/NyRV/zwpT/9BLUr/SDFG/08zQ/9XNUH/YDc+/2o4Pf91OD
v/gzg6/5l4Of+iODn/sTc4/8Q2N//VNjf/1TY3/9U2N//VNjf/1TY3/9U2N//VNsiiAAC8sgAAAscAAAKXOA
ACY2QAAjflCAIT/EAF7/xgEcf8eCWj/lw9f/ycUV/8sGVH/MR5M/zUHR/87JEP/QCY//0YoPP9NKjr/V
Cs3/10sNf9nLDP/ci0x/34tMP+NLC//miwv/6krL/+1Ky//tSsv/7UrL/+1Ky//tSsv/7UrL/+1K76tAACzvAA
Ap8oAAJnVAACL3gAAgPoAAHb/CgFt/xEEY/8XCFr/HA1T/yERTP8lFuf/KhhC/y4aPv8zHDv/Nx43
/zwfNP9BIDL/SCEv/04iLP9Wlir/XyMo/2ojJv91lyX/gCMj/44il/+YliP/mClj/5gil/+YliP/mClj/5gil/+YlrW
3AACpxgAAm9IAAlzbAAB94wAAcv8AAGj/AgFe/wkEVf8OB07/FatH/xkOQf8dET3/lhM4/yYVNf8p
FjH/LRcu/zEYK/81GSj/Ohom/z8al/9GGyH/TRse/1UbHP9eGxv/ZxsZ/3EbGP95Gxj/eRsY/3kbGP9
5Gxj/eRsY/3kbGP95G6vCAACdzAAjkdAAH7gAABv5wAAZP8AAFr/AAFQ/wAESf8FBkH/Cwk7/x

ALNv8UDTL/GA8u/xwQKv8gESj/lxll/yYSlv8pEx//LRQd/zEUGv82FBj/OxUW/0lVFP9JFRL/UBUQ/
1cVDv9eFQ7/XhUO/14VDv9eFQ7/XhUO/14VDv9eFf9GGgr/PiQO/zQvEv86MRT/PTYy/z0/HP87S
yD/Olkk/zhoKP83dyr/N4Ur/ziSK/85nSz/Oacs/zqvK/86tyv/O8Ar/zzKK/881Sr/PeEq/z3rKv8+9Cr/Pvs
p/z//Kf0//yn7P/8p+kD/KPpA/yf6QP8n+kD/J/pA/yf6QP8n+kD/J/9HGgr/PyMO/zctEv89LxT/QDQX/0
E9HP9ASSD/Plcl/z1IKf87dCz/O4lt/zuPLv88mi7/PaQv/z2tLv8+tS7/Pr4u/z/ILv8/0i7/QN4u/0DpLf9B
8i39Qfot+0L/LfIC/y33Qv8s9kP/KvZD/yr2Q/8q9kP/KvZD/yr2Q/8q9kP/Kv9IGQn/QCMO/zorEv9BLR
P/RDIX/0Y7G/9FRyD/Q1Qm/0FjKv8/cS3/P38w/z+MMf8/mDH/QKly/0GqMv9BsJL/Qrsy/0LFMv9Cz
zH/Q9wx/kPnMftE8TH5RPox90X/MfVF/zHzRv8v8kb/LvJH/y3yR/8t8kf/LfJH/y3yR/8t8kf/Lf9JGQn/Q
SIO/z0qEf9FKxP/STAW/0s6Gv9LRSD/SFIm/0ZgK/9Ebi//Q3wy/0OJNP9DITX/Q582/0SoNv9Erzb/
Rbg2/0XCnv5FzDb8Rtk2+kbmNvdH8Db0R/k28kf/NvBI/zXvSf8z7Ur/Me1K/zHtSv8x7Ur/Me1K/zHt
Sv8x7Ur/Mf9KGQn/QiIO/0EnEP9JKRL/Ty8V/1I4Gf9RQx//T04m/0xcLP9JazH/SHg1/0eFOP9HkTr/
R5s6/kekO/xHrDv7SLU8+ki+PPIlyTz3SdU89EnjPPFK7zzuSv97Er/POpM/zroTf84507/NeZO/zX
mTv815k7/NeZO/zXmTv815k7/Nf9LGAAn/QyEN/0YID/9PJxH/Vi4T/1k3GP9ZQR7/Vksl/1NYLf9QZj
P/TXQ5/kyAPPtLjD/5S5ZB+EugQvZLqEL1S7FD80u6Q/JLxETxTNFE7UzgROpM7UXnTPdF5E7/
QuJQ/z/gUf893IL/Ot5T/zreU/863IP/Ot5T/zreU/863IP/Ov9MGAn/RSEN/0sjDv9UJA//XSwR/2E1Fv9
hPhz/X0kk/1tULP5YYjT6VG8891F7QfRPh0XyTpFH8E6bSe5OpErsTq1L6061TOIOwE3oTs1O5U
7dTUFO60/dT/ZN21L/SdhU/0bWVf9D1Ff/QNNX/z/TV/8/01f/P9NX/z/TV/8/01f/P/9OGA/RiAN/1Ah
Df9blw3/ZCoP/2kzE/9qPBr/aEYi/WRQK/hgXjXzXGo+71d2RexUgUvpUoxP51GWUuRQn1TiUKhV
4FCxV95QvFfcUclX2ILaV9VS6VfRU/VVz1b/UMxY/0zLWv9JyFv/Rcdb/0XHW/9Fx1v/Rcdb/0XHW/
9Fx1v/Rf9PFwj/SCAN/1UeDP9ilgz/aykN/3AxEf9yOhf+cUMf925NKfFqWjTsZGY/515xSeNZe1HfV
oVX3FSQW9lUmlzWVKRd1FWsXdFVt17PVsNezVbTXslW5F7GV/Jcw1r9V8Fc/1PAXv9PvF//S7tf/
0q7X/9Ku1//Srtf/0q7X/9Ku1//Sv9QFgj/Sx4M/1kcCv9nIQr/cSgL/3cwDv96OBT4ekEb8XdKJetzVjHl
bWI+3mZsS9lgdIXTIBb0FqKX8xZIGHKWZ5jx1mnY8VasGTDWrxkwVvLZL5b3WS6W+1kuF76X
rdh/1m1Yv9VsmP/Ubfj/1CxY/9QsWP/ULFj/1CxY/9QsWP/UP9RFgj/Th0L/14aCf9tIAj/dycJ/30uDP
yANhDzgT4X7IBHleR9Uy7ddl481W9oSs9oclXJZHtdxWGFYsFfjma+XphovF6habpeq2m3X7ZqtV/
DarNg1WqvX+hqrmL2Za1l/1+qZ/9bp2n/VqZp/1Wmaf9Vpmn/VaZp/1Wmaf9Vpmn/Vf9SFgf/UhsK/
2IZCP9xHwf/fCUH/4MsCfeHNA3viTwT54IFHd+FUSrWfls6zndISMdwbIXAa3Zeu2d/ZLdliWm0ZJJ
ssmObba9jpW6tY69vq2S8b6lkzW+mZOFvpGbx6Nq/WWha/9gnmr/W55q/1qeav9anmr/Wp5q/1q
eav9anmr/Wv9TFQf/VRkJ/2cYB/92HgX/gCQF/ocqB/OMMgrrqjzoP4pBCGNmMTifQhVk4x35iR8B4
alO5cnJes257Zq9rg2uraYxvqGiWcaZooHOJaKpzoWi2dJ9pxnSdadt0mmrtcZpt+mqYbf9llm3/YJVt/
16Vbf9elW3/XpVt/16Vbf9elW3/Xv9UFQf/WRcl/2oXBv95HQT/hCIE+owoBe+RLwfmITcM3ZZAFdO
STCXKi1Y1wYVgRbl+aFKyeW9drXR3Zqdx22jbodxoG2RdJ1sm3aabKV3mGyxeJZswHiUbNR4k
WzpdPjw+G+QcPxpjm//ZI5v/2KOb/9ijm//Yo5v/2KOb/9ijm//Yv9VFAf/WxYH/24WBf99HAP/hyEC95
AmA+yWLATimzQI2Zs8E86WSiPFkVUzvlpeQ7SEZVCtf2xcp3pzZqF2e22cc4NzmHGMd5Vwlnm
Sb6F7j2+tf11vu3yLb898iG/le4ly9nOlcvpthnL9Z4Zx/maGcf5mhnH+ZoZx/maGcf5mhnH+Zv9WFAf/
XhQH/3EWBP+AGgL/ix8B85MjAeiaKALeoC8E06A6EcqbSCHBIVMxuJBcQbCKY0+ohWpboYBxZ
Zt8d22VeH90kXWHeY10kXyJc5x+h3Kpf4Rzt3+Cc8p/f3Phf4F29HeAdfhxf3T7a390/GI/dPxp3T8a
X90/GI/dPxp3T8af9XFAf/YRMG/3QVA/+DGQH+jhwB8JcfAOSflwHapSgDz6Q4EMWfRh+8mlEvt
JVAP6uPYU2kimhZnIZuZJWBdG2PfXt0inqDeoV4jX6Bd5iBfnalgnx2s4J5d8WCd3fdgXI58np4efZ0
eHj6bnh3+2x4d/tseHf7bHh3+2x4d/tseHf7bP9YEwb/ZBEF/3cUAv+GFwH7kRkA7JoaAOCjGwDVq
SUCy6c2DsGkQx24n08tsJpYPaeVYEufkGZXI4tsY5CGcmyJgnh0g3+Aen19iX95e5SCdnuhhHN7r
4RxfMGEb3zZg3B+731xffR2cXv4cHF7+W5xe/lucXv5bnF7+W5xe/lucXv5bv9YEwb/Zw4E/3oTAF+I

FAD4IBQA6J4SANuoDgDQrCICxqwzDL2oQRu0pEwrrq59WOqOaXUmbImRVk5FqYYuMb2uDiHV
zflV8enaChYBxgZCdbYGdhWuBrIVpgb2FZ4LVhGiD7X9pgvN4aoD3cmp/+HBqf/hwan/4cGp/+HB
qf/hwan/4cP9ZEgb/aw4D/34RAf+MEQDylw0A2aMIANSsCADLsB8BwbAxC7itPxmWqUopp6VTOJ
6gW0aWnGFTjpdnX4aTbWI9j3Jydox5eW+Jgn9pil2DZYiahWOlqYVhiLqEX4nRg2CJ639iiPN5Yob
3cmKF+HFihfhxYoX4cWKF+HFihfhxYoX4cf9bEgb/bw0C/4EOAPmPCwDanAUA06YGAM2wBgD
FtRwAvLUuCbOyPBerr0cmoqtQNZmnWEORol5QiJ5kXICaamZ3l3Bwb5R2d2iSf31iklqBX5CXg1y
Qp4NakbiCWZHPgVmR6X1bkPR4Xl33clyM+HFcjPhxXlZ4cVyM+HFcjPhxXlZ4cf9dEAX/cwsB/4U
KAOOTBADUUnwQAZaoFAMe0BAC/uhcAt7srB664ORSmtUQjnLJNMpSuVECLqItNgqZhWHqjZ2N
xoGxsaZ1zc2KbfHlcmo9WZqVfLeapX5Vm7Z9VJvNfFOb6HIVmfV1VpX4cFaU+W5WIPluVpT5bla
U+W5WIPluVpT5bv9iDQT/eAkA9okFANIYAQDOpAMAxq4DAL+4AgC4wBEAsMImBajANRGgvU
Agl7pJLo62UDyFslJfK9dVHOsY15qqmlnYqhwblymeXNXpYV2VKWTeFKmo3dRprV2T6bMdU6
m53NQpfhvUaD7a1Gf/GpRn/xqUZ/8alGf/GpRn/xqUZ/8av9oCAL/fgUA3Y8AANGdAADlqQIAwLM
BALi+AgCwyAoAqcshA6LKMA6ZxzscKMLEKofATDd9vVJEdLpYT2y3XlhjtWVgXLRsZ1eydmtSso
JuULKRb06zom5Ns7NtS7PLbEqy52tLsfhoTK3/ZEys/2NMrP9jTKz/Y0ys/2NMrP9jTKz/Y/9vAwH2h
AAA1pUAAMqjAADBrgEAubkBALDEBACnzwcAodUaAZrVKgqR0jYXiNA/JX/NRzF1yk09bMdTSG
TFWVFDw2BYVsJpXIHCC2JOwYBkTMKPZErCoWRJw7JjSMPKYUfC52BHwPheSL7/W0i8/1tlvP9
bSLz/W0i8/1tlvP9bSLz/W/94AADdjAAAzpwAAMOpAAC6tQAAscACAKjLBACf1ggAmeAdAJLhKw
eJ4DMSgN46H3bbQStt2Uc2ZNdOQFzVVEhW1FxOUNRIU0zUcFZK1H1XSNWNWEfVn1dF1rFW
RNbKVUPV5VRD0vVTQ9D/UUPQ/1BD0P9QQ9D/UEPQ/1BD0P9QQ9D/UPKBAADUIAAAxqQAA
LywAACzvAAAqcgCAJ/SBACW6A0AkOogAojSkwh/6zIRdus3G2zpPCVj6EluXOdJNIXnUD1Q5ld
CS+ZhRkjna0hG53hKROiHskPomUIC6aplQem+R0Dp2kY/5/FFP+T/RD/j/0Q/4/9EP+P/RD/j/0Q/4/
9EP+P/RNuMAADKnQAAvqwAALW4AACqxAAAn88BAJTYAwCN8xIAhfUfA332Jwl09i0Ra/YzGW
L2OCBa9T0nU/VDLU71SjFJ9Vl1RvVbOEP2ZTpB9nE7QPZ+Oz73jjs+96A6PfixOjz4xjk7+N44Ovb
1Nzr1+Tc69fk3OvX5Nzr1+Tc69fk3OvX5N8+XAADBpwAAtrQAAKzBAACgzAAAINUAAIniAQCB/h
IBef8cBHD/lwlo/ygPX/8uFlj/MhtR/zggTP89JEf/QydD/0sqQP9TLD3/XC07/2YuOf9yLzf/gC82/5AuN
v+hLjX/si00/8MsNP/fKzP/5Csz/+QrM//kKzP/5Csz/+QrM//kK8OiAAC4sAAArr4AAKHJAACU0gAAh
9sAAH3zAgB1/w4BbP8XBGP/HQlb/yIOVP8nEk7/LBZl/zEaRP82HD//PB88/0lgOf9lIjb/UCMz/1gk
Mf9jJC//biQt/3wkK/+MJCv/myMq/6kjKv+8lir/wSlq/8EiKv/Blir/wSlq/8EiKv/BlirtAACvugAAo8cAAJX
QAACH2QAAeeAAAG/7AABn/wgBXv8QBFB/FghP/xsLSf8gD0P/JRl/ykUOv8uFjf/Mxcz/zcZMP89
Gi3/Qxsq/0obJ/9SHCX/XBwj/2ccf9zHB//gRwe/44bHv+dGx7/oRse/6EbHv+hGx7/oRse/6EbHv+hG
7K3AAClxAAAI88AAIjWAAB53wAAa+UAAGH/AABZ/wECUf8IBEr/DQdD/xMJPF8YDDn/HA40/yEP
Mf8IES3/KRIq/ywTJ/8wEyT/NRQh/zoVHv9BFRv/SRUZ/1IVFv9cFRT/ZhUT/3EVEf99FRD/gRUQ/4
EVEP+BFRD/gRUQ/4EVEP+BFafBAACZzQAAitUAAHrdAABq5AAAXOwAAFP/AABL/wABRP8A
Az3/BAU3/wkHMv8OCS3/Ewsq/xcmJv8bDSP/Hg0g/yEOHf8kDxr/KA8X/ywQFP8xEBH/NhAO/z0Q
C/9FEAn/ThAG/1YQBP9gEAP/YhAD/2IQA/9iEAP/YhAD/2IQA/9iEP85HQv/MCcO/y0uEf8yMBL/N
DUV/zl+Gf8wShz/L1gf/y1nlf8tdSL/LIMj/y2Ql/8vmiP/L6Qi/zCslv8xsyL/Mrwh/zLFlf8zzyD/M9sg/zTm
IP818B//Nfkf/zB/H/02/x77Nv8e+jf/Hfk3/x35N/8d+Tf/Hfk3/x35N/8d+Tf/Hf86HQv/MScO/zAsEf81LhL
/NzMV/zY8GP81SBz/M1Yg/zJklv8xcyT/MIEI/zGNJf8ymCX/M6Ek/zSqJP80sST/Nbkk/zBDI/82zSP/
N9kj/zflv847yL9OPgi+zn/lfk5/yH3Ov8g9zr/IPY7/x/2O/8f9jv/H/Y7/x/2O/8f9jv/H/87HQv/MiYO/zMqE
P84LBL/OzEU/zw6GP86Rhz/OFMg/zZil/81cCb/NX4n/zWKJ/82lSf/N58n/zinJ/84ryf/Obcm/znAJv86
yib/OtYm/jvjJfw77SX5PPcl9zz+JfU9/yXzPf8j7/lvE+/yLxPv8i8T7/lvE+/yLxPv8i8T7/lv88HAv/MyY
O/zcoEP89KRH/Py4U/0E4GP9BRBz/PIAh/zxeJf87bCj/Onop/zqHKv87kiv/O5wr/zykK/88rCv/PbQq/

z29Kv4+xyr8Ptlq+j/gKvc/7Cr0QPUq8UD+Ku9B/ynuQv8n7UL/JutD/yXrQ/8l60P/JetD/yXrQ/8l60P/J
f89HAr/NCUO/zsID/9CJxD/RiwT/0k2F/9lQBz/RU0h/0NaJv9BaCr/P3Ys/z+CLv8/ji/+QJgv/EGhMPt
BqTD6QbEw+EK5MPdCwzD2Q88w80PdMPBD6jDtRPQw6kT9MOhF/y7nRv8s5Uf/KuNI/ynjSP8p
40j/KeNI/ynjSP8p40j/Kf8+Gwr/NiQO/0AjDv9lJA/TisR/1E0Ff9QPhr/TUkh/0pWJ/9HZCz+RnEw+0V
+MvIFiTT3RZM19UWdNvNFpTbyRa028Ua1N+9GvzfuRss37EfaN+hH6DjK/M44kn9Nt9K/zPdS/8
x20z/L9pN/y7aTf8u2k3/LtpN/y7aTf8u2k3/Lv9AGwr/OilN/0UgDf9OIQ3/VikP/1kyE/9ZPBj/VkYf/1JR
JvtPXy33TGwz9Et4N/FJgzrvSY487UmYPetJoD7pSak+50mxP+ZJu0DkScZA4knVQd5J5ULaS/JA
1038PdRP/zrSUP830FH/Nc9S/zPPUv8zz1L/M89S/zPPUv8zz1L/M/9BGgr/PyAM/0sdC/9VIAv/Xig
N/2lWEP9iORb/YEMd+ItOJfRYWy7wVGc161FzO+hOfkDITYhD4kySReBMm0feTKRI3EytSNpNtkj
YTsNI1U7RSNJ04knOT/Bly1H7RMIT/0DHVf89xVb/O8RX/zjEV/84xFf/OMRX/zjEV/84xFf/OP9CG
Qn/Qx0L/1AbCv9cHwr/ZSYL/2ouDv9rNhP6aUAa82ZKI+1hVyznXGM24lhuP95UeEbaUoJK1lIGNT
NRRik7RUZ9Oz1GoT81SsU/LUrtPyVPJT8ZT20/CU+tPv1X4S71Y/0e8Wf9Dulv/QLlc/z65XP8+uV
z/Prlc/z65XP8+uVz/Pv9DGQn/RxsK/1UYCP9iHgJ/bCQI/3EsC/1zNA/0cz0W7XBGH+ZrUyrfZl812W
BpQNNbc0jOWH1Oy1eGUchWkFPFVplUw1aiVcFXq1W+V7VWvFfCVrpY0la3WOVWtFr0U7Nc/k
6xXv9JsGD/Rq5j/0OuY/9DrmP/Q65j/0OuY/9DrmP/Q/9FGAn/SxkJ/1oWB/9oHAb/ciMG/3gqCPh7
MQzvezoS53lDGt91UCbXb1s0z2hlQMlj0nEX3hQwF2BVb1cili6W5NauFucW7VbpVuzXK9csVy7
XK9dy1ysXd5cql3vWqlg+1SmZP9PpmX/TKVm/0ilZv9lpWb/SKVm/0ilZv9lpWb/SP9GGAn/Txcl/18
VBv9tGwX/dyEE/n0nBfOBLwjggzcN4YJAFdh9TCPPd1cxyHBhP8Frakm7ZnNRt2N8V7NhhVuWYI
5erWCXYKtgoGGpYKphp2C1YaVhxGGjYddin2LqYJ9l+FqeZ/9VnWj/UZ1p/02daf9NnWn/TZ1p/02
daf9NnWn/Tf9HGAn/UxUH/2MUBf9yGgP/ex8D+YIIA+6HKwXlijMJ3lk9EdKESSDjflUvwXhePbpyZ
0i0bW9Sr2p3Watnf16nZohhpGWRZKJlm2WfZaVmnWWwZptlvmaZZdBnlmTIZ5Zn9WCWaf9aW
v/VpRr/1KUa/9SIGv/UpRr/1KUa/9SIGv/Uv9lFwj/VhMG/2cTBP9lGAL/fx0B9YchAuqMJwPgcC8F1
o86D8yKRx7DhFltu35c07R5ZEetdGxRqHBzWaNte1+fa4NknGqMZ5lplmmXaKBqlGira5JouWuQ
aMprjWjha41q82WNbP5fjW7/WoxT/1aMbf9WjG3/VoxT/1aMbf9WjG3/Vv9lFwj/WREF/2sSA/95FgH/
gx0B8oseAOaQlgHclSkD0ZM4DsePRRy+ilArtoRaOa9/YUaoemlQonZwWZxyd2CYch9mlG2laZF
skWyObJxujGunb4lrtG+Ha8VvhWvcb4Vt8GmFb/1jhXD/XoRw/1qEcP9ahHD/WoRw/1qEcP9ahHD
/Wv9KFwj/XA4E/24RAv98FAH8hhYA7o4YAOGUHADWmSQCzJg1DMOUQxq6j04psolYN6qEX0
SjgGZPnHttWZZ3dGCRdHtmjXKDa4lwjW+Gb5dxg2+jcoFvsHJ/b8FyfW/Ycnxw7W1+c/tfnP/YX1y
/119cv9dfXL/XX1y/119cv9dfXL/Xf9MFQf/XwwE/3EQAf9/EgD5iRIA6pESAN2YEgDSnSIByJwzC7+
YQRi2k00nro9WNaaKXUKEhWRNI4FqV5F8cWCLexhnhnZ/blJ0iXF+c5Nze3OfdXlZrHV2c711dH
PSdXR06nF2dvpqdnb/ZHZ1/2B2df9gdnX/YHZ1/2B2df9gdnX/YP9PFAf/YgsD/3QOaf+BDgDzjAw
A3ZUJANacCQDNoB8BxKAXcbudPxaymEslqRUM6KPXECai2JMk4ZoVoyCbl+GfnVngHt8bXt5
hXJ2el91c3ecd3B3qXdud7l3bHjOdmx46HNuevhsbnr+Z255/2Juef9ibnn/Ym55/2Juef9ibnn/Yv9SE
gb/ZQoC/3cMAPuECwDfjwUA1ZgGANGfBwDlpBwAv6QvCLahPRSunUgjpplSMJ6VWj6WkGBKjo
xmVlelbF6AhHJmeYF5bXN/gXJvfYx2a32YeGh9pnhmfbZ4ZH3Ld2N+5XVmf/ZuZ3/9aGd9/2Rnff9
kZ33/ZGd9/2Rnff9kZ33/ZP9VEAX/aQkB/3oKAO2HBQDYkgMA0ZsFAMujBQDDqBkAu6gsBrKmO
hKqokYgop5QLpmaWDuRII5HipJkUoKOaVx6i29lc4h2bG2FfnJnhIh2Y4OveGGDo3hfhLN4XYTId
1yE4nVfhfVuYIT8aWCD/2Rgg/9kYIP/ZGCD/2Rgg/9kYIP/ZP9YDQT/bQgB/34GANyLAADTIgMay5
8EAMWnAwC9rBUAtq0pBa2rNxClqEMenaVNK5WhVTiMnVxEhJlHt3yVZ1l0km1ibZBzamaNe3B
hjlV0XYuSdlqLoXZYjLF1V4zGdFWM4HNYjPRtWYz9aFqJ/2Raif9kWon/ZFqJ/2Raif9kWon/ZP9dC
gP/cgUA8IIAANEPAADOMglAxqMCAL6rAQC3sRAAsLIIA6ixNA6grkAbmKtJKI+oUjWHpFhBf6FeT
HaeZFZum2peZ5hxZmCXeWxbiYNwV5WQcVWVn3JTIbBxUpbEcFCV4G5SIfRqVJX/ZVSS/2JUk

v9iVJL/YISS/2JUkv9iVJL/Yv9iBgL/dgIA3ocAANGUAADInwAAwKgBALiWAAcwtgoAqrghAqK4MA
uatjwXkrJGJImvTjGBrFQ9eKlbR3CnYFFopGdZYaNtYFuhdmZWolFpUqCOa1CgnmtPoa5qTaHD
aUyh32dNoPNkT5//YE+c/11PnP9dT5z/XU+c/11PnP9dT5z/Xf9nAAH2fAAA2lwAAMuZAADCpAA
Auq0AALG1AACovQMAosAbAJvAKwiUvjCti7tBIIK5SSx6tIA3cbNWQmmxXEthr2NTW65qVWVtc
15RrH5hTqyMYkysnGJLra1hSq3CYEmt3l5Jq/NcSqr/WUqp/1dKqf9XSqn/V0qp/1dKqf9XSqn/V/9v
AADfgwAA0JMAAMWfAAC8qgAAs7MAAKq7AQChwwQAmcoTAJPLJQWMyTIPg8c8G3vERCZy
wksxacBRO2K+V0RbvF5LVbtmUFC6cFRNunxWSrqKV0m7m1dHu6xWRrvBVUW73IRFufNSRbf/
UEW2/05Ftv9ORbb/TkW2/05Ftv9ORbb/Tvp3AADXiwAAyZoAAL6mAAC2sAAArLkAAKPCAQCYY
gUAj9QJAIRWHQKD1isKetQ1FXLSPSBp0EUqYc9MM1rNUjtUzFpBT8xiRkvLbUllY3IKRsyIS0XMm
UtEzatKQs3ASUHM3khByvJGQcj+RUHH/0RBx/9EQcf/REHH/0RBx/9EQcf/ROCBAAOOkwAAwal
AALitAACutwAApMEAAJnJAQCP0QQAheEKAlHjHwF65CoHcuMyEGniOBlg4T8iWeBGKIPfTTB
O31U2Sd9eOUbfaTxE33U9QuCEPUHglT1A4ac8P+G7Oz7h1jo93u85PNz7OTzb/zg82/84PNv/O
Dzb/zg82/84PNv/ONSMAADFnAAAuqoAALG1AACmvwAAm8gAAI/PAACE2AMafe0RAHbvHgNv
8CclZvAuD17wMxZX7zkcUO8/lkvvRidH704qQ+9YLUHvYi8+8G4wPfB7MDzxizA78Z0vOvKvLjnyx
C448t8tN/DzLDbt/yw27f8sNu3/LDbt/yw27f8sNu3/LMmXAAC8pgAAs7IAAKi9AACcxwAAkM4AAIP
WAAB43wEAcvkRAWv7GwRj+yIIW/woDIT8LhNO/DMXSfw5G0T8Px5A/UYhPf1Olzr9WCQ4/mlIN
f5uJjT+fSYz/40IMv+fJDL/sSMx/8UjMP/cli//7iEv/+4hL//ulS//7iEv/+4hL//ulb+iAAC1rwAAqrsAAJ7FA
ACQzQAAg9QAAHbcAABs8QEAZf8NAV7/FgRW/x0IUP8iDEr/Jw9F/y0SQP8yFTz/Nxc4/z0ZNf9E
GjL/Sxsv/1QcLP9fHSr/ax0o/3kdJ/+KHCf/mxsm/6obJv+8GiX/zBkl/8wZJf/MGSX/zBkl/8wZJf/MGba
sAACsuQAA8QAAJLMAACD0wAAAddsAAGjiAABf+wAAWP8HAVH/DwRK/xUHRP8bCT//IAw7/y
UON/8pEDP/LhEv/zITLP84FCj/PhQm/0UVI/9OFSD/WBYe/2QWHP9xFRr/gRUZ/48VGf+dFBn/qh
QZ/6oUGf+qFBn/qhQZ/6oUGf+qFK62AACiwigAAk8sAAITSAAB12gAAZ+EAAFrnAABS/wAAS/8A
AUT/BwM//wwFOf8SBzT/FwkW/xsLLP8gDCn/JA0l/ycOlvr8Dh//MA8c/zYQGf89EBb/RRAT/04QEP
9ZEA3/ZRAL/3EQCv9+Dwn/iQ8J/4kPCf+JDwn/iQ8J/4kPCf+JD6TAAACVygAAhtIAAHbaAABm4A
AAWOcAAE3wAABF/wAAPv8AATj/AAMy/wMELf8IBin/DQcl/xEliv8VCR//GQoc/xwKGP8fCxX/lws
S/ycMDv8sDAv/MgwH/zoMBP9CDAH/TAwA/1YMAP9fDAD/aAwA/2gMAP9oDAD/aAwA/2gMAP9
oDP8sIQv/lvsO/yYtD/8pLxH/KTQT/yY8Ff8kSRj/l1ca/yJmG/8hdBv/IYIb/yOOG/8kmBr/JaEa/yapGf8
msBn/J7gY/yjAGP8oyhf/KdUX/yniFv8q7Bb/KvYV/yv+Ff0s/xX7LP8U+i3/FPkt/xT5Lf8T+S3/E/kt/xP
5Lf8T+S3/E/8tIQv/JCsO/ykrD/8sLRD/LTIS/yo6Ff8pRxj/J1Ua/yZjHP8lchz/JX8d/yaLHP8olhz/KZ8b/
ymnG/8qrhr/K7Ya/yu+Gv8sxxn/LNMZ/y3gGP8u6xj+LvUY/C/9F/kw/xf3MP8W9zD/FvYx/xX1Mf8V9
TH/FfUx/xX1Mf8V9TH/Ff8tIAv/JSoO/ywpD/8wKhD/MS8S/zE5Ff8wRRj/LVlb/yxgHf8rbh7/K3wf/yul
Hv8skx7/LZwe/y6kHf8vqx3/L7Md/zC7HP8xxBz/Mc8c/jLdG/wy6Rv5M/Ma9zT8GvQ0/xrzNf8Z8jX/G
fE1/xjwNf8Y8DX/GPA1/xjwNf8Y8DX/GP8vIAv/KCgO/zAmDv81Jw//NiWR/zk2Ff83Qhj/NU8c/zNdH/
8xayD/MXgh/zGEIf8yjyH/M5kh/zShlf80qCD/NbAg/jW4IP02wSD8N8wf+TfaH/Y45x/0OPEf8Tn7Hu
45/x7tOv8d6zr/HOk8/xvoPP8b6Dz/G+g8/xvoPP8b6Dz/G/8wHwv/LCUN/zUjDf86JA7/PioQ/0AzFP
9APxj/PUsc/zpZIP84ZiL/N3Mk/zeAjf44iyX8OZQl+zmdJfk6pSX4Oqwl9zu0JfU7vSX0PMgk8jzVJO
895CTsPfAk6T76JOc//yPIP/8i4kH/leBC/x/gQv8f4EL/H+BC/x/gQv8f4EL/H/8yHgv/MSIM/zogDP9AI
Q3/RygP/0kxEv9IOxf/RUcc/0JUlf9AYSX8Pm4n+T57KfY+hir0PpAq8z+ZK/E/oSvvQKkr7kCxK+1B
uSvrQcQr6kHSK+ZC4SviQu4s30P5Kt1E/yjaRv8n2Eb/JdVH/yTVR/8k1Uf/JNVH/yTVR/8k1Uf/JP8z
Hgv/Nh8M/0AdC/9IHgv/TyYN/1IvEP9ROBX/TkMb/kpPIfHXCB0RWkq8UR1Le5EgC/rRlsw6USUM
edEnTLIRKUy5EWtMuJFtjPgRcAz3kXONNtG3zTXRu0000j4MdBK/y/OS/8tzeZ/K8pM/ynKTf8pyk3
/KcpN/ynKTf8pyk3/Kf81HQv/OxwL/0UZCF9QHqN/VyUL/1stDf9bNhL9WEAY91NKH/FQVybS2WQQt

50twMeRJezXhSIU33kiPOdxImDnaSaA62EmpOtVJsTrTSrw70UrIO89L2TvLS+k7x032OMVO/zX
DUP8zwVH/ML9S/y6/Uv8uv1L/Lr9S/y6/Uv8uv1L/Lv82HQR/QBoJ/0sWCP9XHAf/XyMI/2MqC/9kMg
/3YjwV715GHehaUyXiVl8u3VJqNdIPdTrUTn890U6JP89OkkDMTppByk6jQchOq0HGT7VCxE/BQ
sJQ0EK/UONCvFHxQLIT/Ty3Vf85tlb/NrVX/zS0V/8ztFf/M7RX/zO0V/8ztFf/M/83HAr/RBcl/1AUBv9
eGgb/ZiEG/2soCPItLwvwazgR6GhCGOFkTyLZX1st01pmNs1WcD3JVHIBxIOCRMNTi0bBU5RHvI
OdR7xTpki6VK9luFS6SLZVyUi0VdtJsFXsSK9Y+UOtWf8/q1z/PKle/zmpXv84qV7/OKle/zipXv84q
V7/OP86Gwr/SBQH/1YTBf9kGAT/bB4E/nlIBfN0LAfqdDUM4XI+E9luSx/QaFcsyWJiNsReaz6/W3
REu1I9SLhYhku2WI5Ms1iXTbFYoe6vWKlOrVm0T6tZwk+pWtNPpInnT6Rc9UqiX/9FomD/QqFh/z
6hYf89oWH/PaFh/z2hYf89oWH/Pf89GQn/TBIG/1wSBP9pFwP/chwC+XgiAu57KATkfDAH23s7D9
F2SB3JcFQqwmpeNbtLZz+2YnBGsl94S69egE6sXYIRqV2SUqddm1OIXaVUo12vVKFevFSfXsxVn
F7hVZpg8lCaYv5LmWP/R5lk/0OZZP9CmWT/Qplk/0KZZP9CmWT/Qv9AFwj/UBAF/2AQA/9tFAH/
dhkB9HweAeiAlwHegysD1IE4Dct9RRrCd1Eou3JbNLRtYz6vaWxGqmZzTaZke1GjYoRUoGKNVp
5hllibYaBZmWGqWZdhtlqVYcdak2HcWpFi71aRZPxQkWb/TJFn/0iRZ/9HkWf/R5Fn/0eRZ/9HkWf/
R/9DFQf/Uw0E/2QOAv9xEgH+ehUA8IAYAOOFHQDZiCUCz4c1C8WDQxi9fU4mtXhYMq5zYD2o
b2hGo2xvTZ9pd1ObZ39XmGalWpVlkVyTZZtdkWSmXo9ksl6MZMJeimTWXoll7FuJZ/pViWj/Ullq/
0yJav9LiWr/S4lq/0uJav9LiWr/S/9GEwf/VgoD/2gNAf90EAD7fREA64QSAN6JFADUjCIByosZCsCl
QRa4g0wjsH5WMKI5XjyjdWVFnXFSTZhuc1SUbHtZkGqDXl1pjF+LaJdhiGiiYoZnrmKEZ71igmjSY
oBo6GCBavhZgWz/VlJt/0+Cbf9Ogm3/ToJt/06Cbf9Ogm3/Tv9JEQb/WQgD/2sMAf93DADzgAsA4
YcJANmMCGDPk8AxZAxCLyNPxS0iEohrlINTLqV/XDqee2NEmHdpTZJzcFSOcHdaiW5/XoZsiG
KDa5JkgGueZX5rqmZ7a7lmeWvNZndr5WR5bfZdem//Whtw/1N7cP9Re3D/UXtw/1F7cP9Re3D/U
f9LEAX/XQcC/24KAPt6CQDigwUA14oGANOQCADKIB0AwZQvB7iRPRKwjUgfqlSLKCEWjiZgG
BDk3xnTi14bVOldXRag3J7X39whGR7b45meG+aaHZvp2lzb7VpcW/JaG9v4mdxcFrgcnL/WnR0/
1V0c/9UdHP/VHRz/1R0c/9UdHP/VP9ODgX/YAYB/3EHAO98BADZhGMA0o4FAM2TBgDFIxoAvZ
gsBbSVOxGskkYdpl5QKpyKWDAvhV5BjoFkS4h9aOCenFafXd4YHh1gGV0dltoCHOWam5zpGtr
c7JraXTFamdZ3mppdfNia3b+XWx3/1hsd/9WbHf/Vmx3/1Zsd/9WbHf/Vv9RCwT/ZAUB/3MFAN5/A
ADVigIAzpEEAMiXBADAmxYAuJwqBLCaOA+olkQboJNOKJiPVjSRi1w/iodiSYODaFJ8gG5Zdn11
YHF7fWVseYdpaXmTa2Z5oGxjea9sYXnCa2B522pievFkZHv9XmV8/1lle/9YZXv/WGV7/1hle/9YZ
Xv/WP9UCQP/ZwQA9ncBANqDAADRjQEAYZUCAMObAgC7nxMAtkEnA6yfNg2knElZnJhMJZS
UVDGNkFo8hY1gR36JZIB3hmxYclRyX2qBemVlgIRpYX+Qa19/nmxcf61rW4C/a1mA2WpbG09kX
YH9X16B/1pegf9YXoH/WF6B/1hegf9YXoH/WP9YBgL/awEA5noAANWHAADMkQAAXJkBAL2fAA
C2ow4Ar6UkAqekMwufoT8WI55JlpCbUS6l1g5gJNeRHmQZE1xjmlVaotwXWSJd2JfilFnW4eNaV
iHm2pWh6tpVYe9aFOl1mdUiO5jVoj8Xlil/1IYh/9YWlf/WFiH/1hYh/9YWlf/WP9cAgH/bwAA3X8AAN
GMAADHIQAAv50AALekAACvqAKaqaogAaKqLwlaqDwTk6VGH4uhTiuDnlU2e5tbQHOYYUlsImd
SZZRtWV6SdV5ZkX9jVpCLZVOQmmVRkallUJG8Y0+R1WJPkO1fUJD8W1KQ/1dSkP9WUpD/VI
KQ/1ZSkP9WUpD/Vv9hAADydAAA2IQAAMuRAADCmgAAuqlAALKoAACprQMAo7AbAJywKwaV
rjgQjaxCHIWPsid9pIEydaRYPG2iXkRmn2RMX55rU1mcc1lUm31cUZuJXk+bmF5Nm6heTJu7XU
uc1FtLmu1ZTJr8VUyZ/1JNmF9RTZn/UU2Z/1FNmf9RTZn/Uf9nAADhegAA0YoAAMaWAAC9oAA
AtKgAAkyuAACitAAAm7cUAJW4JgSOtjMnhrQ9F36yRiJ2sE0sbq5TNmasWj5fqmBGWahnTFSnc
FFQp3pUTaeHVUunl1ZJp6dVSKe6VEen1FJGpu1QR6X7Ttik/0xlo/9LSKP/S0ij/Otlo/9LSKP/S/xuA
ADZgQAAY5AAAMCcAAC3pgAArQ0AAKW0AACbugAAkb8LAI3BIAGGwC0Jfr84Ena9QR1uu0gm
ZrIPL1+3VTdZtlw+U7VvkQ0+0bUdLtHhKSbSFS0e0lUtGtKZKRLS5SUO000hDs+1GQ7H7RUOW/0
NDsP9DQ7D/Q0Ow/0NDsP9DQ7D/Q+R3AADRiQAAXJgAALqjAACxrAAAqLQAAJ26AACTwAEAi

MgFAILMFgB8zSYFdcswDW3KOhdlyElgXsdJKFjGUC9SxVc0TcRgOUneATxGxHU+RMSDP0PE
kz9CxaU+QMW4PT/F0jw/w+07PsH7Oj6//zk+v/85Pr//OT6//zk+v/85Pr//OdqBAADlkgAAvaAAALsQ
AACqswAAoLsAAJXBAACJxwEAfs8FAHTYCgBx2x0CatspCGPaMhBc2ToYVthCH1DXSiVL1llqR
9ZbLUTWZTBB1nExP9eAMj7YkTI92KMxPNm2MDvZ0S871uouOtT4LjnS/y440f8uONH/LjjR/y440
f8uONH/Ls6MAADAmwAAatqgAAK2yAACiugAAIsEAAIvIAAB/zgAAc9YEAGzmDgBn6B0CYeknB1r
pLgxT6TUSTeg7F0joQhxE6EsfQehUl7pXiQ86WolOup3JjnqhyU565kIOOusJDfrwSM1694iNOjzIT
Pl/yly5f8iMuX/ljLI/yly5f8iMuX/lsSXAAC4pQAAR7AAAKW6AACywgAAi8gAAH/OAABz1QAAZ90B
AGL0EABc9RoDVvYiB1D2KQtK9y8PRfc1EkD3OxU990IYOfhKGjf4Uxs0+F4cMvlqHDD5eRww+o
ocL/qcGy76rhot+8UZLPrdGSv58hgq+PcYKvj3GCr49xgq+PcYKvj3GLuiAACxrQAAP7gAAJrBAAC
NyAAAf84AAHLVAABI3AAAXO0AAf/DAFR/xUDS/8cBkX/IglA/ycMPP8tDjj/MxA0/zgSMf8/Ey7/Rx
Qr/1AVKP9aFSb/ZxUk/3YVI/+HFSL/mRQi/6sTlf+8EiD/1Blg/90SIP/dEiD/3Rlg/90SIP/dErOrAACpt
wAAnMEAAI7IAAB/zgAAcdUAAGTcAABY4gAAUPgAAEv/BgFF/w4DP/8VBTTr/Ggc2/x8JMv8kCi7/
KQwq/y4NJ/8zDiT/OQ8h/0APHv9JDxv/VBAZ/2AQFv9vDxx/fw8U/5AOFp+fDhP/rg0T/7QNE/+0D
RP/tA0T/7QNE/+0Dau1AACevwAAkMgAAIDOAABx1QAAYtwAAFXjAABK6QAARP8AAD//AAE5/
wUCNP8LBC//EQUr/xYGKP8aByT/Hwgh/yIJHf8mChr/KwoX/zELE/84CxD/QAsN/0oMCv9XCwf/Z
AsF/3ILBP+ACgP/jgoD/5MKA/+TCgP/kwoD/5MKA/+TCqG+AACSxwAAgs4AAHLWAABi3QAAVO
MAAEjpAAA+8gAAOP8AADL/AAEt/wACKP8BAyT/BgQg/wsFHf8QBRr/EwYX/xche/8aBxD/HggM
/yIICP8nCAT/LggB/zUIAP8/CQD/SggA/1YIAP9hCAD/bQgA/3AIAP9wCAD/cAgA/3AIAP9wCP8gJ
Qv/GCsN/x4rDf8fLQ//HTIQ/xk6Ev8YSBP/F1YU/xZIFf8WcxX/FoAU/xIME/8ZlhP/Gp4S/xqmEf8brR
H/HLQQ/xy8EP8dxQ//Hs8P/x7dDv8f6A7/H/IN/yD7Df0g/wz7lf8M+iH/DPki/wz5lv8M+SL/DPki/wz5l
v8M+SL/DP8hJQv/GyoN/yEqDf8kLA7/ljAQ/x84Ev8eRhT/HFQV/xTiFv8bcBb/G34V/xyJf8dkxT/HP
wU/x+kE/8gqxL/ILIS/yG5Ev8iwhH/Is0R/yPaEP8k5hD+JPEP+yX6D/kl/w73Jv8O9ib/DvUm/w71Jv8
O9Sb/DvUm/w71Jv8O9Sb/Dv8iAv/HygN/yUndf8oKQ7/Jy4Q/yY3Ev8lQxT/l1EW/yFfF/8hbRf/lXoX
/yKGF/8jkBb/JJKW/yWhF8lqBX/Jq8U/ye3FP8nvxP/KMoT/ijWEvwp5BL5Ku8S9ir5EfQr/xHyK/8R8
Sz/EfAs/xDuL8Q7i3/EO4t/xDuL8Q7i3/EP8jJAv/JCYM/yokDP8tJQ3/LioP/y81Ev8tQBT/K00X/yIbG
P8oaRn/KHYZ/yiCGf8pjBn/KpUY/yudGP8spRj+LKwX/S2zF/suvBb6LsYW+C/SFvYv4RXzMO0V7
zH3Fe0x/xXrMv8U6jL/FOc0/xTINf8T5TX/E+U1/xPINf8T5TX/E/8klwv/KCMM/y8gC/8zlQz/NigO/zgy
Ef83PRT/NEkX/zFWGv8wZBv/L3Ec/y99HPwwiBz6MZEc+TKaHPczoRv2M6gb9TSwG/M0uBvyNc
la8TbOGu423hrqN+sa5zf2GuQ4/xniOf8Z3zr/GN07/xfbO/8X2zv/F9s7/xfbO/8X2zv/F/8mlwv/LR8L/
zUdCv86Hgv/QCYM/0EvD/9AORP/PUUX/zpSG/44Xx36N2wf9jd4IPQ4gyDyOlwg8DmVIO45nSDs
OqUg6zqslOo7tCD0O74g5jzKIOQ82iDgPOkh3T31INo+/x/VQP8e00H/HdFB/xzQQf8b0EH/G9BB/
xvQQf8b0EH/G/8nlgv/MhwK/zoZCF9CHAn/SSMK/0ssDf9KNRH/R0AW+0NNG/VAWh/xP2Yi7T5yJ
Oo+fSXnPocl5T+QJuM/mSbiQKEm4ECpJ95AsSfcQbsn2kHHJ9hB1yfUQuco0EP0JsxF/iXKRv8jx
0b/lsZH/yHER/8gxEf/IMRH/yDER/8gxEf/IP8rHgr/NxkJ/0AVCP9KGgf/USEI/1QpCv9TMg76UDwT8
0xHGexJVB/nRmEk40VtJ99EeCrcRIlr2USLLNdFIC3URZwt00WkLTFGrC7PRrUuzUfBLspHzy7HR
+EvxEjwLsFK+yu+S/8pvEz/J7tN/ya6Tf8luk3/JbpN/yW6Tf8luk3/Jf8wHAr/PBUI/0YRBv9SGAX/WR
8G/10mCptdLwvyWjgQ6lZCF+NSUB7dT1wl2ExnK9NLci7PSnwxzUqFMspKjjPISpY0xkueNMRLpj
XCS681wEy6Nb5MxzW8Tdk2uE3qNbVP+DKzUP8vsIH/LbBS/yuuVP8qrlT/Kq5U/yquVP8qrlT/Kv8
0GQn/QRIG/00RBf9ZFgT/YBwE/2QjBfVIKwfrZDMM42A+EttdTBvTWfclzVRiLchSbDLEUHY1wV/
B/OL5Qhzm8UJA6ulCYO7hQoTu2UKk8tFGzPLJRwDyUte8rVHkPKtT9DmpVf81plj/M6VZ/zCk
Wf8vpFn/L6RZ/y+kWf8vpFn/L/84Fgj/RQ4F/1MOA/9fFAL/ZxkC+msfAu5tJgTkbS8H22o6DdJmRxn
KYVMkw1xeLb5ZaDS6V3E5tlZ5PLRVgT6xVYpAr1WSQa1Vm0GrVaRCqVauQqdWukIvSpDo1b

eQ6BY70CeWvw8nVv/OJ1c/zWcXP8znFz/M5xc/zOcXP8znFz/M/88FAf/SQwE/1gNAv9kEQH/bBY
A9HEaAOhzlQHedCgD03l2C8puRBfCaVAjvGRaLbZgZDWxXmw7rVx0P6pbfEKOwOREpVqNRq
NalkehWp9Hn1qpSJ1atEibW8NImVvXSZZb60aWXfIBIV7/PZRf/zqUX/84IF//OJRf/ziUX/84IF//OP8/
Egb/TQkD/10LAf9oDgD9cBEA73UUAOJ4GQDXeiMBzXkzCcR1QRW8cE0htWxXLK9nYDWqZGg
8pWJwQaJgeEWfX4BlnF+ISppekUuYXppMll6ITZResE2SXR5OkF7RT01d50yNX/dHjWH/Qoxi/z+
MYv88jGL/Plxi/zyMYv88jGL/PP9CDwX/UAYC/2AJAP9rCwD2cwsA6XgMANx7DQDRfyABx38xCL
57PhO2d0ofr3JUKqluXTSjamU8nmhsQpplc0eXY3tLIGKDTZJhjE+PYZZRjWGgUothrFKJYLpSh2
DNUoRg41KEYvVMhWT/R4VI/0OFZf9AhWX/QIVI/0CFZf9AhWX/QP9FDQX/VAUC/2QHAPpuBw
DldgUA2XwGANWACADMgx0AwoQuBrqBPBGyfUgdq3hRKKR0WjKecGE7mG1pQpRqcEiQaHd
MjWZ/UlpliFOHZJFUhWScVYNkqFaAZLZWfmTIVnxj31Z8ZfJQfWf/S31n/0d9aP9EfWj/RH1o/0R9
aP9EfWj/RP9HCwT/VwQB/2cFAO9xAwDaeglA1IAFAM+EBgDHiBoAvogsBbaGOg+ugkYbpn5PJ
p96WDGZdl86k3JmQo5ubEiKbHNNhmp7UoJog1WAal1XfWeYWXtnpVp4Z7JadmfEWnRn3Fp0af
BUdWr9TnZr/0p2a/9Hdmv/R3Zr/0d2a/9Hdmv/R/9KCQP/WwMB/2kCAN90AADWfQEaz4QDAMql
BQDCixcAuowqBLKKOA2qh0QZooNNJJt/VS+Ue105jndjQYlzaUiEcHBOF253U3ttgFd4blladWuVX
HNroVxwa69dbmvAXWxr2Fxsbo5Xbm78UW9v/0xvb/9Jb2//SW9v/0lvb/9Jb2//Sf9NBwP/XgEA92w
AANx4AADSgQAAY4cCAMWMAwC9jxQAtpAnA66PNgyim0IXnohlpeEUy2QgFo3inxhQIR5Z0d+
dm1OeXN0VHVxfFhxclZcbnCRXmtvnl9ob6xfZnC9XmRw1F5lcOxZZnL6U2hz/09oc/9LaHP/S2hz/
0toc/9LaHP/S/9PBAL/YQAA6m8AANh7AADOhQAAxosBAMCQAQC5kxEAsZQIAqqTNAqikEAVm
o1JIJOJUSuMhVg1hYJfPn9+ZEZ5e2tNc3lxU253eVlqdoJcZnWOX2N1m2BhdalgX3W6X1110V9e
depaX3f5VGF3/1BieP9NYnj/TWJ4/01ieP9NYnj/Tf9TAQH/ZAAA4HMAANR/AADKiAAAw08AALuU
AAC0lwwArZkiAaaYMQielT0Sl5JHhO+PTyili1YzgYhcPHqFYkR0gmhMbX9uUmh+dlhjFh9cX3uLXI
x7mGBae6dfWHu3X1d7zl5XfOhaWXz4VVp9/1Bbff9NW33/TVt9/01bff9NW33/Tf9WAAH7aAAA3H
cAAM+DAADGjQAAPMAALaYAAcCumwgAqJ0eAKGdLgaamzoQkphFG4uVTSaDkVQwfl9aOXW
MYEJuiWZJaldsUGKFc1ZdhH1aWYOIXFaDII1Ug6VdU4O2XFGDzFtRg+dYU4P4U1SD/09Vg/9N
VYP/TVWD/01Vg/9NVYP/Tf9aAADrbAAA1nwAAMuIAADBkQAAuZgAALGdAACooAlAaoqMaAJyJk
gSVoTcNjZ5CglacSiJ+mVEsd5ZYNXCTXj5pkWRFYo9qTFyOcVJXjHpWVlyGWFGMIFIPjKNZTo
y0WE2My1ZMjOZUTYv3UE6L/01Pi/9KT4v/Sk+L/0pPi/9KT4v/Sv9fAADhcgAA0YIAAMaNAAC9lgA
AtJ0AAKyIAAchpgAam6gUAJWpJgOPqDMKh6Y+FICjRh54oU4ocZ9UMWqcWzljm2FAXZloR1e
Xb0xTlnlQT5aFUK2Wk1JLIqNSSpa0UUmWy09lluZOSZX3S0mU/0hKIP9GSpT/RkqU/0ZKIP9GSp
T/RvtmAADbeQAAY4gAAMCTAAC4nAAAR6MAAKaoAACbrAAAK64MAI6wIQGIry4HgK45EHmsQ
hpxqkojaqhQK2OmVzNdpV46V6RIQFKibUROonZHS6KDSUmikUploqJJRqKzSEWiykdEoeVFRK
D3Q0Wf/0FFnv9ARZ7/QEWe/0BFnv9ARZ7/QOZtAADTgAAAXY4AALuZAACyogAAqqkAAKCuaA
CVsgAAirYEAIW4GQB/uCkEebc0DHG2PRRqtEUdY7NMJVyxUixWsFkyUa9hN02vajtJrnQ+R66A
P0Wujz9ErqA/Qq6yPUGvyTxArU7QKz3OkCr/zk/qv84P6r/OD+q/zg/qv84P6r/ON13AADLiAAAv5
YAALagAACtqAAAo68AAJm0AACNuAAAgb0BAHnCDgB1wyABb8MtB2jCNg9hwT8WW79GHIW
+TSRQvIUpS71dLUe9ZjFEvXEzQr1+NEG9jTQ/vZ4zPr2wMj29yDE8vOUwPLr3Lzu5/y86t/8vOrf/L
zq3/y86t/8vOrf/L9OBAADDkQAAuZ0AALCnAACnrwAAAnLYAAJC6AACEvwAAeMUBAG3MBgBn0
BUAY9EjA13RLglY0DYPUs8/FU3PRhtlzk8fRc5XI0HOYSU/zm0nPc56JzzPiic7z5wnOs+vJjnPxYU
4zuUkN8v2JDbJ/yQ1yP8kNcj/JDXI/yQ1yP8kNcj/JMmMAAC8mgAAs6YAAKmvAACetgAAkrwAAI
bBAAB6xgAAbSwAAGPTBABa3wwAWOAcAVPhJgRO4S8JSeE2DUXhPxFB4UcVPuFRFzzhWxk
64mYaOOJ0Gzfgxs25JUaNeSoGTTkvRky5NsYMODxGC/e/Bku3f8aLt3/Gi7d/xou3f8aLt3/Gr+WA
AC1pAAARk4AAKG2AACVvQAAh8IAAHrHAABuzQAAYtMAAFjbAQBS7g0ATu8ZAUrwlgRF8CkH

QPAvCjzxNg048T4PNfFGETPyTxlw8loTLvNmFCzzdRQs9IYTK/SYEyr1rBlp9clRKPXeeCby8hA
m8P4PJvD+Dybw/g8m8P4PJvD+D7ehAACurAAAO7YAAJa9AACJwwAAe8kAAG3OAAABh1AAAVt
sAAEzIAABl+woARPwUAj/9GwQ7/iIGN/4oCDP/Lgkv/zMLLP86DCn/Qg0m/0sOJP9WDiH/Yw4f/3I
OHv+EDh7/lw0d/6oMHP++DBv/1gsa/+gLgv/oCxr/6Asa/+gLgv/oC7CrAACmtQAAMl4AAIrEAAB8
ygAAbc8AAF/WAABT3AAASelAAELzAAA9/wQAO8MATX/FAMx/xkELf8fBin/JAcm/ygllv8uCB//N
Akc/zsKGf9EChb/TwoT/10KEf9sCg//fQkO/5AJDv+hCA3/sAgN/8AHDf/ABw3/wAcN/8AHDf/AB6i0
AACbvQAAjMQAAH3LAABt0QAAX9gAAFHeAABG5AAAPoKAADb8AAAY/wAALv8EASn/CgIm/x
ADlv8VBB//GQUB/x0FGP8hBhT/JgYR/ywHDf8zBwr/PaCh/0cHA/9UBwD/YgcA/3IGAP+CBgD/kA
YA/5wGAP+cBgD/nAYA/5wGAP+cBp28AACPxAAAf8sAAG7SAABf2QAAUOAAAETmAAA56wAA
MPEAACv/AAAm/wAAlv8AAR7/AABe/wQCGP8JAXX/DQMR/xEEDf8UBAn/GAQG/x0FAv8iBQD/
KQUA/zEFAP88BQD/SAUA/1UFAP9iBQD/bgQA/3gEAP94BAD/eAQA/3gEAP94BP8TKAv/EioL/x
YpDP8WKwz/EjAN/ww5D/8LRw//CIUQ/wlkd/8Jcg//Cn8O/wyKDF8Nkwz/DpsM/w+jC/8Qqgv/ELAK/
xG4Cv8RwAn/EsoJ/xLWCP8T4wj/FO4H/xT4B/0U/wb6Ff8G+RX/BvkW/wb4Fv8H+Bb/B/gW/wf4Fv
8H+Bb/B/8UJwv/FSgL/xonDP8aKQz/Fy4N/xM3D/8SRBD/EVIQ/xBhEP8QbxD/EHwP/xKHDv8TkQ
7/FJkN/xWgDP8Vpwz/Fq4L/xa1C/8XvQr/GMcK/xjTCf8Z4Qn+GewJ+xr3CPga/wj2G/8l9Rv/CPQc/
wjzHP8l8xz/CPMc/wjzHP8l8xz/CP8VJwv/GiYL/x4lC/8fJgz/HSsN/xw1D/8aQRD/GU8R/xddEf8Xax
H/F3gR/xiDEP8ZjQ//GpYP/xudDv8cpA7/HKsN/x2yDf8eugz/HsQM/h/PC/sg3Qv4lOoK9SH1CvIh/g
rwlv8K7yL/Cu4i/wrrJf8K6iX/Cuol/wrqJf8K6iX/Cv8WJgv/HyML/yMiC/8llwv/JSkN/yYyD/8kPhD/IUsS
/yBZE/8fZxP/H3MS/yB/Ev8hiRH/lpIR/yOaEP0joRD8JKgP+yWvD/oltw74JsAO9ybLDvQn2g3xKOg
N7ijzDesp/Q3pKf8N5yr/DeMs/w3hLf8N4S3/DeEt/w3hLf8N4S3/Df8bJAv/lyAK/ygeCv8qHgr/LyYM/
zAwDv8uOhD/K0cS/yIUFP8oYhX/J28V/Ch6FPophRT4Ko4U9iqWE/UrnRPzLKUS8iyrEvEtsxLvLr
wR7i7IEewv1hHoL+UR5DDyEelx/BHfMf8R2jT/Edk0/xDXNP8Q1jT/ENY0/xDWNP8Q1jT/EP8glQR/
KBwJ/y4ZCf8zGwn/OCMK/zosDP85Nw//NUIS/zJPFfsXRB3MGkX8zB1F/ExgBfuMokX7DOSF+s
zmRbpNKEW6DSofuY1sBbINbkW4zbEFuE20hbdN+MW2TjwFtU4+xbROv8Wzjv/Fc07/xTLO/8U
yJv/FMo7/xTKO/8UyJv/FP8kHgn/Lhgl/zUVB/88GQf/QiEI/0MpCv9CMg3+Pz4R9zxKFf16VxjtOGQZ6
ThwGuY5ehvjOYQb4TqNG987lRvdO50b3DukHNo8rBzYPLUc1TzAHNM9zh3QPd8dzD7uHchA+
hzFQP8bw0H/GsFC/xm/Qv8Yv0L/GL9C/xi/Qv8Yv0L/GP8pGwn/MxQH/zsRBv9FFwX/Sx4G/00mC
P5MLgv2SDgP7kVFE+hCURjjQF4b3kBqHdpAdR/XQH8g1ECHldJAKCLQQZcizkGflsxCpyPKQq8
jyEO5l8ZDxiPERNYkwEToJLxF9iK6Rv8ht0f/H7ZH/x61SP8dtEj/HbRI/x20SP8dtEj/Hf8tFwj/OBEG/
0IOBP9NFAP/UxsE/1YiBfdVKgftUjQL5U4+Ed5MTBbYSVkc0kdklc5GbiTKRngmx0aBJ8VHiSjDR5
EpwUeZKb9loSm9SKkqu0izKrlJvyq3Sc8qtEniK7FK8imvTP0nrUz/JapP/yOpUP8iqVD/lalQ/yGpUP
8hqVD/lf8yFAf/PQwF/0kMA/9TEgL/WhcC+10eAvBdJQTmWy4H3V6kDNRVRxXNUIQdx09fJMJO
aSi/TXlrvEx7LbpMgy63Tlsvtk2TL7RNmzCyTaQwsE2tMK5OuTGsTsgxqk7bMadP7TCkUfotolP/Kq
FT/yigVP8moFT/JqBU/yagVP8moFT/Jv82EQb/QQkD/08KAv9ZDgH/YBMA9GMYAOhkHwHeYyc
D1GI1CsteQxTEWk8dvldaJbIVZCq1U20uslJ1Ma9SfT0tUYU0q1KNNalSljanUp82pVKoN6NTszez
U8E3n1PTN5xU5zeaVfcmVb/MJhX/y2YV/8rl1f/K5dX/ylXV/8rl1f/K/86DgX/RQUC/1QIAf9eCwD9
ZA0A7mgRAOFpFgDWailBzGoyCMRnQBK8Y0wctI9WJbFcYCysWWgxqVhwNKZXeDejV4A5oVa
lOp9XkdDudV5o8m1ejPJlXrj2XV7s9lVfNPZJW4z2RWPQ5kFn/NZBa/zKPWv8wj1r/L49a/y+PWv8vj
1r/L/89CwT/SQIC/1gGAP1iBwDtaAcA32sHANpsCgDPcR4AxnEvBr1uPBC2akgbr2ZTJKpiXCylYG
UyoV5sN51ddDqaXHs9mFuDPpZajECUWpVBklqfQZBaqqKOWrdCjFrIQ4lZ3kOJW/E+iFz+Oohd/
zeHXv80h17/M4de/zOHXv8zh17/M/9ACQP/TQIB/1wDAPFIAGDcbAEA1nAFANJyBwDJdhsAwHc
sBbh0Og6wcEYZqmxQl6RpWSueZmEymmNoOJZhcDyTX3c/kF5/Qo5eh0SMXZFFil2bRoddpkeF

XbNHg13ER4Fc2keAXu9DgF/8PoBg/zuAYP83gGH/N4Bh/zeAYf83gGH/N/9DBgP/UQAA/18AAO
FoAADYcAAA0XUDAMx3BQDEexgAu3wqBLN6OA2sdkMXpXJNIZ5vViqZa14ylGhIOJBIBD2MY3
NBiWJ7RIZhg0eEYlxJgWCXSn9go0t9YK9Le2DAS3hg1Ut4YOxHeGL7Qnhj/z55Y/87eWT/Onlk/zp
5ZP86eWT/Ov9FBAL/VQAA9mIAAN1sAADTdAAAzhHoCAMZ8AwC/fxUAAt4AnA69/NQuoe0EVoXh
LH5p0UymUcFsxj21iOlppaT2GaG9CgmZ3Rn9lf0l8ZlhLemSTTXdjn051Y6xOc2O8TnBj0U5wZOIL
cWX5RXFm/0FyZ/8+cmf/PXJn/z1yZ/89cmf/Pf9IAgL/WAAA6mUAANlwAADPeAAAYH4AAMGBAQ
C6gxlAs4UIAquDMwmkgD8TnX1JHZZ5USeQdVkvInJfN4VvZj2AbGxCfGpzR3hpe0t1aIVOcmePU
G9nnFFtZ6IRa2e5UWlnZlFoZ+dOaWn3SGpq/0Rrav9Aa2r/P2tq/z9rav8/a2r/P/9KAAH/WwAA4Wg
AANV0AADLfAAAXlIAAL2FAAC2hw4Ar4kiAailMQighT0RmYJHG5J+TyWLeIYuhXddNoB0Yz16c
WIDdm9wSHFueExubIFPa2yMUmhsmFNmbKZTY2y2U2FsylNgbORQYm32SmNu/0Zkbv9CZG//
QWRv/0Fkb/9BZG//Qf9NAAH8XgAA3mwAANF4AADlgAAAwIYAALmJAACxiwoAqo0fAKSMLwaci
jsPIYdFGY6DTSOHgFQsgX1bNHt6YTt1d2dCcHVtSGtZdUxncn5QY3GJU2FxlVRecaNUXHgzVFt
xx1NZceJRW3L1S11z/0dec/9DXnP/Ql5z/0Jec/9CXnP/Qv9QAADxYQAA2nAAAM18AADEhAAAvl
oAALSNAACsjwYAppEcAJ+RLAWYjzgNkYxDF4qJSyGDhllqfINYMnaAXjpwfmRBantrR2V6ckxge
HtQXXeGUlp3k1RYd6FUVnexU1R3xVJTd+BRVXj0S1Z4/0dXeP9EV3j/Q1d4/0NXeP9DV3j/Q/9U
AADIZQAA1XUAAMmAAADAiAAAul4AALCSAACnIAIAoJYYAJqWKQOUIDYLjZJAFIWPSR5+jFA
nd4pWL3GHXDdrhWl+ZYNpRF+BcEpbghHov3+DUFR/kFFSf59RUX+vUU9/w1BOf95OT3/zSIB//
0ZRf/9DUX7/QIF+/0JRfv9CUX7/Qv9ZAADgawAA0XoAAMWFAAC8jQAAtJMAAKuXAACHmQAA
mpsTAJWcJQKPMjlliJg9EYCWRhp5k00jcpFULGyPWjNijWA6X4tmQFqJbUZViHZKUoiBTE+Hjk1
Nh51NTliuTEqlwktJiN1KSofyRkuG/kNLhv9ATlb/QEyG/0BMhv9ATlb/QPdeAADbcQAAy38AAMC
KAAC3kgAAr5gAAKecAACanwAAk6ENAl6ilAGloS4GgqA5DnqeQhdzm0ofbZIRJ2aXVy9gll41Wp
RkO1WTbEBRknVETZGARkuRjUdJkp1GSJKtRUeSwURFkd1DRZDyQEaQ/j5Gj/88Ro//PEaP/zx
Gj/88Ro//POdlAADUeAAAxoYAALyQAACzmAAAq54AAKGiAACWpQAAl6cGAlaoGgCBqSkDe6c
1CnSmPhJtpEYaZqNNImChVClaoFovVZ9hNFCeaTIMnXM8SZ1+PkedjD5GnZw+RJ2sPUOdwTt
Cnd06QpvyOUGa/zdBmf82QZn/NkGZ/zZBmf82QZn/Nt9tAADNfwAAwIwAALeWAACungAApqQA
AJuoAACQqwAAgq4AAHywEgB4sSMBcrAvBmyvOQ5lrkEVX61IHFmsTyJUq1YoT6peLEupZjBHq
XAYraI8NEOpjRCqZo0QKmrMz+pwDI+qdwWpafyMD2m/i88pP8uPKT/Ljyk/y48pP8uPKT/LtZ3AA
DGhWAAupQAALKdAACppQAAn6sAAJSuAACIsQAAlUAaHC4BwBsuhoAaLooA2K6MghduTsP
V7hCFVK4ShpNt1EfSbdZl0W2YiZCtm0oQLZ5KT62iCk9t5gpPLEqKDu3vyc6ttwmObTyJTiy/yY3sf
8mN7D/Jjew/yY3sP8mN7D/JsyBAAC+kAAAtZwAAKylAACjrAAAmLEAAly0AAB/uAAAc7wAAGfB
AgBexg0AW8gdAFfIKQRSyDIITsc7DUnHQxJFxsWQsZUGT/GXR8s8xmgdO8Z1HjnHhB44x5UdN
8eoHDbHvRs1x9sbNMTyGzLC/hsxwP8cMcD/HDHA/xwxwP8cMcD/HMOMAAC4mQAAR6QAAKa
sAACbsgAAjrcAAIK6AAB1vgAAaMMAAF3JAQBT0AUAS9YNAEnYHABG2ScDQ9kwBj/ZOQk82U
IMOtIMDzfZVhA12mlSNNpvEjPbfhly25ASMdyjES/cuBAu3NQQNLjtECvV+hEp1P8SKdP/EinT/xlp
0/8SKdP/EruWAACxogAAqawAAJ6zAACRuQAAL0AAHbBAABpxgAAXcsAAFLRAABJ2AIAQuY
KAEDoFwA96SECOukpBD bqMAYz6jgIMOtBCS7rSgos7FULKuxiDCjtcAwn7YELJu6UCyXuqAok7
r8JI+7dCSHr8wgf6P8JH+f/CR/n/wkf5/8JH+f/CbShAACrqwAAoLMAAJ06AACFvwAAAd8QAAGnIA
ABczQAAUdMAAEfaAAA+4AAAOvUHADf3EgA09xoBMPghAy35JwQp+S0FJvo0BiT6PAch+0YIH
/tRCBz8Xgga/G4IGf2ABxj9kwcX/qgGFv2+BhX91gUU/e8FFPzzBRT88wUU/PMFFPzzBa2qAACis
wAAlbsAAIfAAAB4xQAAacoAAFvQAABP1gAARNwAADvhAAAz6gAAMP4BAC3/CgAp/xIBJv8YAi
P/HQMf/yIDHP8oBBn/LgUW/zYFE/8/BRD/SgUN/1gFCv9oBQn/egUI/44EB/+hBAAb/sGmf/8cDBf/M
AwX/zAMF/8wDBf/MA6WYAAcYuwAAicEAAHnHAABqzAAAW9IAAE3ZAABC3wAAOOQAAC/oA

AAp9QAAJf8AACL/AAAf/wcBHP8NARn/EgIV/xCEv8bAw7/IAML/yYDB/8tAwP/NgQA/0EEAP9PB
AD/XwMA/3ADAP+CAwD/kgIA/6ICAP+IAgD/pQIA/6UCAP+IApq6AACMwQAAfMgAAGvOAABb1
AAATNwAAEDiAAA15wAALOsAACPVAAAe/gAAGv8AABf/AAAU/wAAEf8BAQ7/BgEL/woBB/8NA
gP/EglA/xCAP8dAgD/JAIA/ywCAP83AgD/RAIA/1MCAP9iAgD/cAIA/34CAP+BAgD/gQIA/4ECAP
+BAv8FKgr/CygK/w0nCv8LKQr/Bi4L/wA3C/8ARQz/AFQL/wBiC/8AcAr/AHwJ/wCHCP8BkQf/AZk
H/wKgBv8Dpwb/A60F/wS0Bf8EuwT/BcQE/wXQBP8G3QP/BuoD/wb1A/wH/QL6B/8C+Aj/Avcl/wP
3Cf8D9wn/A/cJ/wP3Cf8D9wn/A/8IKQr/DyYK/xlICv8QJwr/CysL/wg1DP8HQWz/BVEM/wRfC/8EbQ
v/BXkkK/waECf8Ilgj/CJYI/wmdB/8KpAf/CqoG/wuxBv8LuAX/DMEF/wzMBP8N2gT9DecE+g3zA/cO/
AP1Dv8D8xD/A/IQ/wTxEf8E7xP/BO8T/wTvE/8E7xP/BP8NJgr/FCMJ/xciCf8WJAR/EigL/xlZDP8QP
wz/Dk0N/w1cDP8MaQz/DXUL/w6ACv8Qign/EZIJ/xGaCP8SoQj/EqcH/xOuB/8TtQb+FL4G/RXJBv
oV1gX3FuQF9BbwBPAX+gTuF/8F7Rj/BesZ/wXnG/8G5hz/BuYc/wbmHP8G5hz/Bv8TJAn/GSAJ/x
weCf8clAn/HSYK/xwwC/8aPA3/GEKxN/xdXDf8WZQ3/FnEM/xd8DP8Yhgv+GY8K/BqWCvsanQn6
G6QJ+ByrCPccsgj2Hbol9R3FB/Me0gfvHuEH7B/uBugg+QfmIP8H4yH/B98k/wjeJP8I3ST/CN0k/wj
dJP8I3ST/CP8YIQn/Hhwl/yIbCP8iGwj/JyQJ/yguC/8IOAz/lkUO/yBSDv8fYA79H2wO+iB3DfchgQ3
1looM8yOSDPIjmQvwJKAL7yWnCu0lrgsJrcK6yfBCuknzgnmKN4J4ijtCd4p+ArbKv8K1iz/C9Qs/w
vSLP8L0iz/C9Is/wvSLP8L0iz/C/8dHQj/JBkH/ygWB/8sGAf/MSEI/zlqCf8xNAz/LUAO/CpND/cpWhD
zKWcQ8ClyD+0qfA/rK4YP6SyODuctlQ7ILZ0O5C6kDeluqw3gL7MN3zC+Dd0wyw3aMNwN1THrD
tEy9w7NNP8PyjT/Dsg0/w7GNf8OxTX/DsU1/w7FNf8OxTX/Dv8hGgj/KRQG/y4QBf82FgX/Ox4G/z
wmB/86MAR6ODsN8zRHD+0yVBHpMmER5TJtEulzdxLfNIER3TWJEds1kRLZNZkS1jagEtQ2pxL
TN68T0Te5E884xRPMONUTyDnmFMQ79BTBO/8TvJv/E7w8/xK6PP8Sujz/Ero8/xK6PP8Sujz/Ev8
mFgf/LxAF/zYMBP8/EwP/RBoE/0UiBfpEKwfxQTUK6T1BDuM7ThDeO1sS2TtnFNU7cRbSO3sXz
zyDF808ixjLPZMYyT2aGmc+ohnFPqkZwz+yGcE/vhm/QM0avEDfGrhB7xq1QfwZs0L/GLFC/xevQ
/8WrkT/Fq5E/xauRP8WrkT/Fv8rEgb/NASe/z4KA/9HEAL/TBYC/U4dAvJNJQToSi8H4EY7C9hESR
DRQ1UVzEJhGMhCaxvFQnQcwK9HcBDhR6+Q4wfvESUH7pEnB+5RKQft0WtILVFtyCzRcUgs
UbYla1G6iCrR/gfqEj/HaZK/xylSv8bpEr/GqRK/xqkSv8apEr/Gv8vDgX/OQYD/0UIAf9ODAH/UxIA9I
UYAOpUHwHfUigD1IA1CM5OQxDHTFAXwUpbHL1JZR+6SW4ht0I2I7VJfiSzSYYksUmOJa9JliWt
Sp4mrEqoJqpKsiaoS78npkvQJ6NL5SegTfQlnk7/I51O/yGcTv8gm07/H5tO/x+bTv8fm07/H/80CwT/
PQIC/0oFAP9TCAD7WAsA7IkQAOJZFgDXWSIBzVkb8VXPw++VUsXuFJWHbRQYCKwT2klrU
5xJ6tOeSmpToAqp06IK6VOKCujT5ksoU+jLJ9PrS2dULktm1DKLZhP3y6WUPeRlVH+KJRS/yaTUv
8kk1L/I5NS/yOTUv8jk1L/I/83CAP/QgAB/08CAPHXAwDnXAQA3F0GANldCgDPYR4AxWluBb1gO
w62XEgWsFISHqtXXConVWQopFRsK6FudC2fU3sunVODMJtTizCZU5Qxl1OeMpVTqDKTU7Qy
kVPEM49T2jONU+4xjFT8LYtV/yuLVf8pilb/J4pW/yeKVv8nilb/J/87BQL/RwAA/1MAAOhbAADbYQ
AA1GQDANBkBwDlaBoAv2kqBLdnOAyWY0QVqmBPHaReWCSgW2ApnFpoLZIZbzCWV3cylFd+
NJJWhzWQVpA2jlaZN4xWpDeKVrA4iFbAOIZW1DiEVuo2g1f6MoNY/y+CwF8tgln/K4JZ/yuCWf8r
gln/K/8+AgL/SwAA91YAAN9fAADVZwAAzmoCAMlrBADCbRYAuW8oA7JtNQqrakETpGdMHJ5k
VSOZYV0plV9ILpJdbDKPW3M1jFp6N4pagjmlWYs6hlmVO4RZoDyBwa08f1m8PX1Z0D17Wec7
e1r4Nntb/zN7XP8we1z/L3tc/y97XP8ve1z/L/9AAAH/TgAA6loAANpkAADRawAAyW8AAMNwAgC8
chMAAtXQIAq1zMwmMcD8Sn2xJGplpUiKUZlopj2NhL4xhaDOIX283hV52OoJdfjyAXYc+flyRP3tcn
UB5XKIAd1y4QXVczEFzXOQ/c132OnNe/zZ0X/8zdF//MnRf/zJ0X/8ydF//Mv9DAAH/UQAA4I0AAN
ZoAADNcAAAXQAAL51AAC3dhAAsHkjAal4MQeidTwQm3JHGZVuTyGP1coimheLoZmZTOCY
2w4fmJzO3thez54YIRAdmCOQnRgmKnxYKZEB1+1RG1fyERrX+FDbGH0PWxh/zltYv82bWL/NG
1i/zRtYv80bWL/NP9FAAD7VAAA32EAANJsAADJdAAAawXgAALp6AACzewwArH0gAKV8Lgaeego

OI3dEF5FzTSCLcFQnhW1cLoBqYjN8aGg4eGZvPHRId0BxZIBDb2SKRWxjlkZqY6NGZ2OyRmVj
xUZjY95GZGTyQGVl/zxmZv84Zmb/NmZm/zZmZv82Zmb/Nv9HAADyVwAA22UAAM9wAADFeAA
AvXwAALZ+AACufwgAqIEdAKGBLAWafzgMk3xCFY14Sx6GdVImgXJZLXtvXzN2bWY4cmtsPW5
qdEFqaX1EZ2iHRmVok0hiaKBIYGivSF5owkhcZ9tHXmjwQl9p/j1faf86YGn/OGBp/zhgaf84YGn/O
P9KAADnWgAA2GkAAMt0AADCFAAAuoAAALKDAACqgwUAo4UaAJ2GKgOWhDYLj4FAE4I+SR
yCe1AkfHhXK3Z1XTJxc2M4bHFqPWhvcUFkbnpFYG2ER15tkElcbZ5JWm2tSVhtv0hWbdhIV23v
Q1hu/T5Zbv87Wm7/OVpu/zlabv85Wm7/Of9OAAAdjXwAA024AAMh4AAC+gAAAtOUAAC6HAACli
AEAnokWAJmKJwKSiTMJi4Y+EYSDRhI+gU4id35UKXJ8WzBseWE2Z3dnPGJ2bkBedHdEWnOB
R1hzjUhVc5tJVHOrSFJzvUdQc9ZGUXPuQlJz/D5TdP87VHT/OVR0/zlUdP85VHT/Of5SAADfZAA
Az3IAAMR9AAC6hAAAsokAAKqMAACgjAAAmY4SAJSPJAGNjjEHh4w7DoCKRBd5h0sfc4VSJm
2CWC1ngV40YX9IOVx9bD5YfHRCVHt+RVJ6i0ZQepIGTnupRk17u0VLe9RETHrtQEx6+z1Nev86
Tnr/OU56/zlOev85Tnr/OfBXAADaagAAy3cAAL+CAAC3iQAARo4AAKaRAACakgAAk5MNAI6UIAC
IIC0FgZI4DHuQQRR0jkkcboxQl2eKViphiFwwXldjNleFajpThHI+T4N8QU2DiUJLg5hCSOoQUiDu
kBHg9M/R4PsPEeC+zplgv84SIH/NkiB/zZlGf82SIH/NuddAADUcAAAxn0AALuHAACzjwAAqpQAA
KGXAACUmAAAJkHAlaaGgCBmykDe5k0CXWYPhBulkYYaJRNH2KTUyVckVorV5BgMFKPaDV
OjnA4S417O0mNiDxHjZc7Ro2nO0SNuTIDjdI4QozsN0KL+zVDiv8zQ4r/MkOK/zJDiv8yQ4r/MuBIA
ADODgAAwYQAALeNAACvIAAAppoAAJydAACRnwAAhKAAAH6hFAB5oiQBdKEwBm6gOQxon0
ITYp1JGlycUCBXm1clUppeKk6ZZi5KmG8xR5h6M0WYhzNDmJYzQpinMkGYuTE/mNlwp5fsLz6
V+y4+IP8tPpT/LT6U/y0+IP8tPpT/LdluAADHfgAAu4oAALOUAACqmwAAoaAAAJejAACLPQAAfKc
AAHOpCwBvqh0Aa6oqA2apNAhgqDwOWqdEFFWmSxIqPVMETKValkikYiZFPgwoQqR3KkCkh
So/pJQqPqSIKT2kuCg8pNlnO6LsJjqh+yY5n/8mOZ7/Jjme/yY5nv8mOZ7/JtB3AADAhgAAtpIAAK6
bAACmogAAm6cAAJCpAACDrAAAdq4AAGixAQBJshMAYLMiAVyzLQRXszYJUrl+Dk6yRhJJSu0
WRrFVGkKwXhxAsGgePbB0IDywgIA7sZlOrGkHziXt43sNEdNq/sHDWt+x00q/8dM6v/HjOr/x4zq/
8eM6v/HseBAAC6jwAAsZoAAKmiAACfqAAAIKwAAlevAAB6sgAAbrQAAGG4AABWwQYAUr8WA
FC/lwFMwC0ESL82B0W/PgtBv0cOPr9QETu/WRM5v2QUN79wFTa/fhU1wl8VNMChFDLAtBMxv
88SML3rEi67+xMtuf8ULLj/FSy4/xUuP8VLLj/Fb6LAAC0mAAArKIAAKOpAACXrwAAi7IAAH21AA
BwuAAAY7sAAFjAAABNxEAQ8wGAEDOFaa+zyAAPM8qAjrQNAQ30D0GNNBGCDLQUAkW0
VwLL9FpCy3SeAss0okLK9OcCirTsAoo0soJJ9DoCSbN+Aoky/8MJMn/DCTJ/wwwkyf8MJMn/DLeW
AACuoQAAPqoAAJqwAACOtQAAGLgAAHK7AABkvwAAWMQAAE3JAABDzgAAO9QCADPaBwA
w4BMAL+EeAC3iKAER4jACKeM6AyjjRAQm5E8EJOVcBSPlawUi5nwFleaPBCDnowQe57oEHOb
ZAxri8gMZ4P0EGN//BRjf/wUY3/8FGN//BbGgAACoqQAAnbEAAJC3AACCuwAAc74AAGXCAAB
YxwAATMwAAELRAAA51wAAMdwAACrnAgAo8A0AJvEXACTyHwAi8iYBH/MtAh30NQla9D8DG
PVLaxb1WAMU9mgDE/d6AxL3jwIR+KQCDve7Ag332AEL9u8BCvP6AQrz+gEK8/oBCvP6AaupA
ACfsQAakrgAAIS8AAB1wQAAZcUAAfKAABL0AAAQNUAADfbAAAU3wAAJuMAACHzAAAF/gU
AHf8NABv/FQAY/xoBFf8gARL/JwEO/y8BC/84Agj/RAIF/1ICAv9iAgH/dQEA/4oBAP+gAQD/tAEA/8
kAAP/cAAD/3AAA/9wAAP/cAKKxAACVuAAAhR4AAHbDAABmyAAAV80AAErTAAA+2gAANN8AA
CvjAAAI5wAAG+oAABj7AAAV/wAAE/8CABH/CQAN/w4ACv8TAAb/GQED/x8BAP8nAQD/MAEA/
zsBAP9JAQD/WgEA/20BAP+CAAD/IAAA/6QAAP+vAAD/rwAA/68AAP+vAJe5AACJvwAAecQAA
GjKAABY0AAASdcAAD3eAAAx4wAAKOcAAB/rAAAX7gAAEfYAAA7/AAAL/wAACP8AAAb/AAAD
/wEAAP8FAAD/CgAA/xAAAP8WAAD/HgEA/yYBAP8xAQD/PwEA/1AAAP9iAAD/cgAA/4EAAP+L
AAD/iwAA/4sAAP+LAP8AKAj/BCUI/wQICP8AJwj/ACwI/wA2CP8ARAJ/AFIH/wBhB/8Abgb/AHoF/w
CEBP8AjtT/AJYD/wCdA/8AowL/AKkC/wCwAv8AtwH/AL8B/wDJAf8A1gH/AOQA/gDwAPsA+gD4

AP8A9gD/APYA/wD1AP8B9AD/AfQA/wH0AP8B9AD/Af8EJgj/CSMI/wkiCP8FJAj/ACgl/wA0Cf8AQ
Qj/AE8I/wBdB/8Aawb/AHcF/wCBBf8AiwT/AJME/wCaA/8AoAP/AKcC/wCtAv8AtAL/ALwB/wDGAf8
A0gH8AOEB+QDuAfUA+ADyAf8A8QL/AfAD/wHuBf8B6wj/Aesl/wHrCP8B6wj/Af8JlWj/DiAl/xAgCP
8MIQj/ByYI/wcxCF8FPgn/A0sJ/wJZCP8BZwf/AnMG/wN9Bf8EhwX/BY8E/waWBP8GnQP/B6MD/g
eqA/0HsAL8CLgC+gjCAvkJzgH1Cd0B8grrAe4K9gHsDP8B6gz/AucP/wLjEv8C4xL/A+MS/wPjEv8
D4xL/A/8OIQj/FB0H/xUbB/8THQf/FCQI/xMuCP8ROgn/DkcJ/w1VCf8MYgj/DG4H/w15Bv0Ogwb7
EIsF+hCSBfgRmQT3EaAE9hKmA/QSrQPzE7UD8hO+A/AUyglTFNoC6RXpAuUW9QLjF/8D3xn/
A9sb/wTaG/8E2Rr/BNka/wTZGv8E2Rr/BP8VHQj/GRkH/xsWBv8bGAb/HyEH/x8rCP8cNgn/GUMJ
/xhQCf0XXQn6F2kl9hh0B/QZfgfyGocG8BqOBU4blQXtHJwF6xyjBeodqgToHbEE5x67BOUexwTj
H9YD3iDnBNoh9AXWlv8F0ST/Bs8k/wbOJP8GzCT/Bssk/wbLJP8GyyT/Bv8ZGQf/HhQG/yERBf8
mFgX/Kh4F/yonB/8oMgj/JT0J+SJKCvQhWArvIWQJ7CJvCekjeQjnJIH5SWKB+MlkQfhJpgG3yaf
Bt4npwbcKK4G2ii4BtgoxAbVKdMG0SrKB80r8gjlL4JxS3/CcMt/wnBLf8Jvy3/Cb8t/wm/Lf8Jvy3/Cf8
eFQb/JBAF/ycLBP8wEwP/NBoE/zQjBf4yLQb2Lzgl7yxFCekrUgrkK14K4CqxqCt0tdAnaLn0J2C6FC
tUvjQrTL5QK0TCbCs8xogvNMaoLyzKyC8kyvQvHM8sMxDtDMA17Q28NfsNuTb/Dbc2/w22Nv8
NtDX/DLQ1/wy0Nf8MtDX/DP8jEQX/KQoE/zEJAv85DwL/PRYC/z0eA/U7JwTsODEG5DU+CN40S
wnYNFgL0zVkDc82bg7MNncPyjd/D8g4hhDGOI4QxDmVEMI5nBDAOqQRvjqsEbw7thG6O8QRu
DvVERQ86BKxPPcSrij3/Eaw9/xGqPv8RqD7/EKg+/xCoPv8QqD7/EP8oDQT/LwUC/zkGAf9BCwH/
RBEA+EUYAexDIAHiQCoD2j83Btl+RQRMPVIOxz1dEcM+ZxPAPnAUvT54Fbs/gBW5P4cWtz+PFr
ZAlha0QJ4XskGnF7BBsReuQb0XrEHOGKIC4hinQvMXo0T/FqJE/xagRP8Vn0T/FJ9E/xSfRP8Un
0T/FP8sCQP/NAAB/0ADAP9HBgD7SgoA70oQAONJFwDZSCMBz0gyBcdHQA vBRkwQvEVXFLh
FYRe1RWoZskVyGrBFehuuRYEbrEWJHKtGkRypRpkdp0aiHaVGrB2jr7ceoUfHHp9H3B6cSO4d
mkj8HJhl/xqXSP8Zlkj/GJZl/xiWSP8Ylkj/GP8wBQL/OQAB/0UAAPVMAADITglA3E8FANIMCgDPU
B4AxlltBL5ROwu4TkcRs01SFq5MXBqrS2UdqEttHqZKdCCkSnwhokqDlaBLiyKfS5QinUudl5tMpy
OZTLikl0vBJJVL1SSSTOojkUz5IY9N/x+OTf8ejk3/HY5N/xyOTf8cj3/HP80AQL/PwAA/EkAAOJQ
AADaVgAA1FcDANBVBgDHWBkAvlopA7dZNwqwVkmRq1ROF6ZSWByjUWAfn1BoIp1QbySaT3
clmE9+JpdPhieVT48ok0+YKJFPoymPT64pjU+8KYtP0CqJT+YpiFD3JoZQ/ySGUf8ihVH/IIVR/yCF
Uf8ghVH/IP83AAH/QwAA700AAN1WAADUXAAAzV4BAMhdBADAXxUAuGEmArFgNAiqXUAQp
FtLF59ZVBybV1whmFvkJJVUayeSU3lpkFN5Ko5SgSuMUoosilKULYhSni6GUqouhFK4LoJSyy+A
Uuluf1P0K35U/yh9VP8mfVT/JH1U/yR9VP8kfVT/JP86AAH/RwAA5FIAANlCAADOYgAAx2UAAMF
kAQc6ZBEAs2cjAatmMQelZD0Pn2FIFpleURyVXFkikVpgJo5YZymLV24riFZ1LYZWfS+EVYYwgl
WQMYBVmzJ+Vacye1W0M3IVxzN3VN4zdlbyL3ZX/yx2V/8pdf/J3ZX/yd2V/8ndlf/J/89AAD6SgAA4
FYAANRgAADKZwAAwmoAALxqAAC1ag0ArmwgAadsLgagaToNmmdFFZRjThyPYVYhi19dJodd
ZCqEW2stgVpyMH5ZeTJ8WYI0eliMNXhYlZ1WKQ2c1ixN3FYwzdvWNs3blnwm25a/i9uWv8sb1r/
Km9b/ypvW/8qb1v/Kv8/AADxTQAA3VsAANBIAADGawAAvm8AALdvAACwbgoAqXEeAKNxLASc
bzigMIWxCE49pSxqKZIMhhWNaJoFhYSp9X2guel5vMXdddjR1XH82clyJOHBclDluXKE6a1yuOml
cwDpnW9c6Z1zuNmdd/DJnXv8vaF7/LGhe/yxoXv8saF7/LP9BAADoUAAA2V8AAMxpAADCCAAA
unMAALN0AACscwcApXUbAJ91KgOYczYKkXFAEotuSRmFa1AggGhXJXxmXip3ZGUudGJsMnF
hczZuYHw4a2CFOmlgkTtmYJ48ZGCsPGJfvTxgX9Q8YGDsOGBh+zRhYf8xYWH/LmFh/y5hYf8u
YWH/Lv9FAADIVQAA1WMAAMItAAC/dAAAAt3cAALB4AACndwMAoXkYAJp6JwKUeDQljXY+Eldz
RheBcE4efG1VJHdRWypyaWlvbmdpM2pmcDdnZXg6ZGSCPGFkjj1fZJs+XWSpPltkuj5ZZNE9W
WTqOlPk+jZbZf8yW2X/MFtl/zBbZf8wW2X/MP9IAADhWQAA0WcAAMVxAAC8eAAAthWAAKx9A
ACjfAAAnH0UAJZ+JQGQfTEHiXs7DoN4RBV9dUwdd3NSI3JxWSltb18uaG1mM2RrbTdganU6X

Wl/PVtpiz5ZaZg/V2mnPIVpuD5Tac49U2noO1Rp+TZVaf8zVWn/MVVp/zFVaf8xVWn/MfVMAADd
XgAAzWsAAMJ1AAC5fAAAsYAAAKiCAACegQAAI4IRAJKDIgGMgi8FhYA5DH9+QhN4e0kac3lQl
W13VydodV0tY3NjMI9yazZbcHM5V298PFVviD5Tb5Y+UW+IPk9vtj1Ob8w8TW/nOk5v+DZPb/8zT
2//MU9v/zFPb/8xT2//MetQAADZYwAAyXAAAL56AAC1gQAArYUAAKWHAACZhgAAkYcMAlyIHg
CHiCwEglY2CnqEPxF0gkcYboBOH2h+VCVjfFsQXnphL1I5aDRVeHA3Und6Ok92hjtNdpQ8THejO
0p3tDpJd8s5SHbmOEI29zRJdf8ySnX/MEp1/zBKdf8wSnX/MOZXAADTaAAAAXUAALt/AACyhgA
AqooAAKGMAACTiwAAi4wHAlaNGgCBjigCe4wzB3WLPA5viUQVaYdLG2OFUiFehFgnWYJfLFS
BZjBQgG40TX94Nkp/hDdlf5l3R3+iN0V/szZEf8k1Q3/IM0R+9zFEff8vRHz/LkR8/y5EfP8uRHz/LuB
dAADObwAAwHsAALeFAACuiwAAppAAAj2SAACQkgAAhJIBAH6TFAB6lCQBdZMvBW+SOQtp
kUERY49IF16OTx1ZjFUiVltcJ0+KZCtMiWwuSII2MEaJgjFEiZEXQ4mhMUKJsi9AickuP4jILT+H9yw
/hv8rP4X/Kj+F/yo/hf8qP4X/KtplAADIdQAAvIIAALOLAACrKQAAOpYAAJiYAACMmQAAfJkAAHWa
DQBxmx4AbZsqAmiaNAdimT0NXZhFEliXTBhTIIMcT5ValUuUYSRHk2onRJN1KUKTgSpBk5ApP
5OgKT6Tsig9k8kmPJLIJTUR9yU6kP8lOo//JDqP/yQ6j/8kOo//JNFuAADCFQAAt4gAAK+RAACnmA
AAnZwAAJKfAACGoAAAd6EAAGuiBABmoxYAY6QkAV+jLwRaozglVaJADVGHrXJNoE4WSaBW
GkWfXh1Cn2gfQJ9yID6ffyE8n44hO5+flDqfsR85n8geOJ7kHTec9x02mv8eNZn/HjWZ/x41mf8eNZ
n/Hsp3AAC8hQAAspAAAKqYAACinwAAI6MAAlyAAB/pgAAcagAAGGqAABaqwsAV6wbAFStJwF
RrTEETaw5CEmsQQtFq0kPQqtREj+rWhQ8q2QWOqtvFzmrFbC3q4wXNqudFjWrrxUzq8YUM6rkF
DGo9xUvpv8WLqX/Fi6l/xYupf8WLqX/FsGBAAC2jgAArpgAAKagAACcpQAAkKkAAIOrAAB2rAAA
aK4AAFuxAABOtQAASbcOAEe3HABFuCcBQrgxAz+4OQU8uEIOrhLCje4VAw1uF8NM7hqDjK4
eA4wuYgOL7maDS65rQwsuMQMK7fjCyq19gwos/8OJ7H/Dyex/w8nsf8PJ7H/D7qLAACxlgAAqaA
AAKcNAAcUqwAAh64AAHmxAABsswAAX7UAAFO4AABlvAAAPsECADfFDAa1xhkANMckADL
HLQEwxzcCLshAAyzlSwQryFYFKcliBSjJcAYnyoEFJsqUBSTKqAQjyr8ElcjfBCDF9QUfw/8GHsH/
Bx7B/wcewf8HHsH/B7SVAACroAAAo6gAAJetAAcKsQAAfLQAAG63AABguQAAVL0AAEnBAAA/
xQAANsoAAC3PAwAl1AcAlDcRACDYHAAg2SYAH9owAB7bOwAe20gAHdxVABzdZAEb3XQAG
d6lABjenQAW3rMAE93SABLa7gAS2PsBEdX/AhHV/wlR1f8CEdX/Aq6fAAClqAAAmq8AAI2zAAB
+twAAcLoAAGG9AABUwQAASMUAAD7KAAA1zgAALNMAACTYAAAc3AAAGOGlABfpEgAW6ho
AFosjABPsLAAR7DcAEO1DAA3uUAAL72EACu9zAAAnwiAAH8J8ABe+3AALt1QAA6+8AAOn/AA
Dp/wAA6f8AAOn/AKioAACcrwAAj7UAAIG5AABxvQAAYsAAAFPFABHygAAPM8AADLTAAAp2
AAAlDwAABrgAAAT5AAAEfMAAA74CAAM+Q8ACvoWAAf6HgAF+yYAAvwwAAD9PAAA/UoAAP
1bAAD9bwAA/IUAAPucAAD6sgAA+coAAPfmAAD35wAA9+cAAPfnAJ+wAACStgAAg7sAAHO/AA
BjwwAAVMgAAEbOAAA61AAAMNkAACbeAAAE4gAAFuUAABDoAAAj7gAAB/sAAAX/AAAC/wM
AAP8IAAD/DwAA/xcAAP8fAAD/KAAA/zMAAP9CAAD/VAAA/2gAAP9+AAD/kwAA/6QAAP+3AAD/
uAAA/7gAAP+4AJ53AACGvAAAdcEAAGXGAABVzAAARtIAADnZAAAt3wAAJOMAABvnAAAT6g
AAC+0AAATvAAAA+AAAAP8AAAD/AAAA/wAAAP8AAAD/AAAA/wYAAP8NAAD/FgAA/yAAAP8r
AAD/OQAA/0oAAP9eAAD/cQAA/4IAAP+SAAD/kgAA/5IAAP+SAP8AJQf/ACMH/wAiB/8AJQb/ACo
G/wA0Bv8AQgX/AFEE/wBfA/8AbAP/AHgC/wCCAf8AiW/AJIB/wCZAP8AoAD/AKYA/wCsAP8Asg
D/ALoA/wDDAP0AzWd6AN0A+QDqAPcA9gD2AP8A9QD/APQA/wDzAP8A8QD/AO8A/wDvAP8A
7wD/AP8Alwf/AiAH/wAfBv8AIQb/ACYG/wAyBv8APwX/AE0F/wBbBP8AaAP/AHQC/wB+Av8Ahw
H/AI8B/wCWAf8AnQD/AKMA/wCpAP4ArwD8ALcA+gDAAPgAywD2ANkA8wDoAPEA8wDwAP0A
7wD/AO4A/wDqAP8A6AD/AOcA/wDnAP8A5wD/AP8EIAf/CB0G/wYcBv8BHQb/ACQG/wAvBv8A
Owb/AEkF/wBxAP8AZAP/AHAD/wB6Av8AgwL/AIsB/wCSAf4AmQH8AJ8A+gCmAPgArAD3ALM
A9QC8APMAxwDwANQA7QDIAOsA8QDpAP0A5wD/AOME/wDgBv8A3wb/Ad8G/wHfBv8B3wb/A

f8KHQf/DhoG/w0YBf8JGQX/CiEG/wgsBv8GNwb/BEUF/wJSBf8BXwT/AmsD/QN2AvsDfwL4BlcB
9wWOAfUFIQH0BZwB8gWiAPEGqADwBq8A7ga4AOwHwwDqB9EA5gjiAOIJ8ADfC/wA2g7/AdY
R/wHVEf8B0xH/AtMR/wLTEf8C0xH/Av8RGQb/ExUF/xMSBf8VFgT/Fx8F/xYoBf8TMwX/EEAF/w5
NBfoNWgT2DWYD8w5xA/APegLuEIMC7BGKAeoRkQHPEZgB5xKeAeYSpQHkE6wB4hO1AOAU
wADeFM4A2hXgANYX7wHRGfsCzRv/Asob/wPIHP8Dxhz/A8Yc/wPGHP8Dxhz/A/8VFQX/GQ4E/x
kLBP8fEgP/lhsD/yIkBP8fLwX7HDoF9RplBe8ZVQTrGWEE5xpsA+QbdQPiG34C4ByGAt4djQLcH
ZQB2h6bAdgeogHWH6kC1B+xAtlgvALQlckCzSLbAsgk6wPEJfkEwSX/Bb4m/wW8Jv8Fuyb/Bbo
m/wW6Jv8Fuib/Bf8aEAX/HgkD/yIIAv8pDgL/LBYC/ysfAvopKQPXJjQE6iRBBOQjTgTfl1sE2yRmA9
glcATUJnkE0ieABNAoiATOKI8FzCmVBcoqnAXIK6QFxiusBcQstgXCLclGvy3SBruw5Qe4LvUHtS/
/CLlv/wixL/8lrzD/CK4w/wiuMP8lrjD/CP8fCwT/JAQC/ysFAf8yCwH/NREA+zQZAfAylgHnLy0C3yw
6A9gsSATSLVQGzS9gB8owaQjHMXIIXDF6CMlygQnAM4gJvjOPCb00lwm7NJ4KuTWnCrC1sAq1
NrwKszbLC7A23wutN/AMqjf+DKc4/wyIOP8MpDj/DKQ4/wukOP8LpDj/C/8kBwP/KQAB/zQBAP85
BQD9OwoA8jsRAOY4GQDcNSUA0zYzA8s3QQbGOE4JwThZC705Ywy7OWwNuDp0DbY6ew60
OolOszuKD7E7kQ+vPJkPrjyHd6w8qxCqPbYQqD3FEKY92RGiPuwRnz76EJ0+/xCcPv8Qmz7/D5
o+/w+aPv8Pmj7/D/8oAgL/LwAA/zoAAPU/AADnQAEA3z8GANw7DADRPx4AyEEuA8FBPAe7QU
gltkBTDrNAXRCwQGYRrUBuEqAdROqQX0UqEGEFKZBixSIQpMV00KcFaFCphWfQ7AWnUO/
FptD0RaYQ+cWiKp3FZRD/xWSQ/8UkkP/E5FD/xORQ/8TkUP/E/8sAAH/NGAA+T8AAONEAADbS
QAA1EkCANFEBwDISBkAv0opArhKNweySEMMrUhoEKIHWBOMRmEVpEZpF6FGcBifRncYnkZ
+GZxHhhqaR44amEeXG5dHoRuVR6wbk0e5HJFHzByOR+IcJef0G4tH/xmJR/8YiEj/F4hl/xelSP8X
iEj/F/8wAAH/OgAA6EMAAN1MAADTUQAAzFIAAMhOAwDAUBUAuFIiArFSMwarUD8Mpk9KEaF
NUxWeTVwYm0xkGphLaxyWS3ldIEt5HpNLgR6RS4kfj0uSII1LnSCLS6ghiUu1IYdLxyGFSt4ig0vx
H4FM/x2ATP8cgEz/G4BM/xqATP8agEz/Gv8zAAD8PgAA40kAANZSAADNWAAXlkAAMBXAAC
5VhEAslkiAatZMAWkVzwLn1VGEZpTTxaWUlgak1FfHJBQZx+OT24gjE51lopOfCOIToUkhk6OJI
ROmSWCTqQlgE6xJn5OwiZ8Ttknek7uJHIP/SJ4T/8geE//HnhP/x54T/8eeE//Hv81AADxQgAA3k8
AANJYAADIXgAAwGAAALpdAACzXAwArF8fAKVfLQSFXTkKmVtDEZRZTBaQV1QajFVcHolUYy
GGU2ojhFJxJYJSeCZ/UYEnfVGKKHxRISl6UaEqd1GuKnVRvipzUdQrcIHsKHFS+yVxUv8jcFP/IX
BT/yFwU/8hcFP/lf84AADpRgAA2lQAAM1dAADEYwAAvGUAALVjAACuYgkAp2QcAKFkKgOaYz
YJIGBAEI9eSRWKXFEahlpZH4JYYCJ/V2YlFZtJ3pVdCI4VX0qdIWGLHNvKs1xVZ4tb1SrLm1Uu
y5rVNEualXpLGIV+ihpVv8maVb/JGIW/yNpVv8jaVb/l/88AADISwAA1lgAAMpiAADAaAAAUgoAAL
FpAACpZwUAomkZAJxqJwKWaDMlkgU+DopjRxSFYE8agF5WH3xdXSJ5W2MmdlpqKHNZcStw
WHotbliDL2xYjjBqWJswZ1ioMWVYuDFjWM4xYlJnL2JZ+CtiWf8oY1n/JmNZ/yVjWf8lY1n/Jf0/AADi
TwAA0lwAAMZmAAC9bAAAtW4AAK1uAAClawEAnm0VAJhuJQGSbTEGjGo7DYZoRBOAZUwZ
e2NTHndhWiJzX2Amb15nKWxdbixqXHcvZ1yAMWVcizJiXJgzYFymM15ctjNcW8szW1zIMVxc9y1
cXf8qXF3/KF1d/yddXf8nXV3/J/RCAADfUwAAz2EAAMNqAAC6cAAAsnMAAKpzAACgcAAAmXE
SAJRylgGOcS8Fh285C4FtQhJ8akkYdmhQHxJmVyJuZF4mamNlKmZibC1jYXQwYGB+MI5giTRc
YJU0WmCkNFhgszRWYMg0VWDjMIZg9i5WYP8rV2D/KVdg/yhXYP8oV2D/KOxGAADBWAAY2
UAAMBUAAC3dAAAr3cAAKZ3AACcdQAAlXYOA193IACJdiwEg3Q3Cn1yPxB3cEcWcm5OHG1sV
SFoalsmZGliKmBnaS1dZnEwWmV6M1hIhjRVZZM1VGWhNVJlsTRQZcY0T2XhM1BI9S9QZf8sU
WX/KIFI/yIRZf8pUWX/KelLAADWXQAAX2kAAL1yAAC0eAAArHwAAKN9AACXegAAkHoLAlp7HA
CFeyoDf3o0CHI4PQ5zdkUUbXRMGmhyUh9jcFkkX29fKVttZixXbG4wVGx4MIJrgzRQa5A0TmufN
ExrrzNLa8QzSWvfMkpr9C5Lav8sS2r/Kktq/yI Lav8pS2r/KeVRAADSYgAAxG4AALI3AACxfQAAqIE
AAJ+CAACSfgAAin8GAISAGAB/gSYBeoAxBnR+OgxufEISaXtJGGN5UB1fd1YiWnZdJIZ1ZCpSd

GwuT3N1MExygTFKco4ySXOdMUdzrjFGc8lwRHLdL0Vy8yxFcf8qRXH/KUZw/yhGcP8oRnD/KO
BXAADNZwAAwHQAALZ8AACTggAApYYAAJyIAACNhQAAhIUBAH2GFAB5hyIAdIYuBG6FNwlp
g0APY4JHFF6ATRpaf1QfVX5bl1F9YidNfGoqSntzLEd7fy1Ge4wuRHucLUN7rCxCe8ErQHvdKkB
68ilAef8nQHj/JkB4/yZAeP8mQHj/JtpeAADlbgAAvHkAALKCAACqiAAoowAAJiNAACLjAAAfYwA
AHaMDQByjR4AbY0qAmiMNAZjizwLXopEEVmJShZUiFEaUIdYHkyGXyJjHwglRoRxJ0OefShCh
IsoQISbJz+FrCY+hMAIPYTcJDyD8iM7gf8jO4D/ljuA/yI7gP8iO4D/ltJIAADDdAAAuIAAAK+IAACnjg
AAnpIAAJOTAACHIAAAAdpMAAG2UBwBoIbCazzUIAWGULwRclDglV5NADFOSRxFPkU4VS5BV
GEeQXRxEj2YeQY9wID+PfCE+j4ohPI+alDuPqx86j8AeOY7cHTiN8h03i/8dNor/HTaJ/x02if8dNon/
HctuAAC9fAAAs4YAAKuPAACjIAAAmZgAAI6aAACBmwAAcpsAAGSbAABdnA8AWp0dAFidKQF
UnTIEUJw7CEycQgtlm0oPRZtSEkKaWhU/mmMXPJpuGDuaeHg5mogYOJqZGDeaqhc1mr8WNJ
nbFTOX8hUxlv8WMJT/FzCU/xcwIP8XMJT/F8R3AAC4hAAAr44AAKeWAACemwAAIJ4AAlegAAB
6oQAAbKIAAFyjAABSpQQATqYUAEynIABJpysBRqc0A0OmPAZApkQJPaZMCzumVQ04pl8PNq
VqEDWmdxAzpoUQMqaWDzGmqA4vpr0OLqXaDS2j8Q4rof8PKqD/ECqf/xAqn/8QKp//EL2AAAC
zjAAAqpYAAKOdAACyogAAjKUAaH+mAABYqAAAZKkAAFarAABKrQAAQa8HAD6xFQA8sSAA
O7EqADmyMwl2sjwDNLJFBTKyTgYwslkHLrJkCC2ycggrsoEIKrKSBymYPqcnsroGJbLXBSSv8AY
jrf8llqz/CSGr/wkhq/8Jlav/CbaKAACulQAAp4AAJykAACRqAAAg6sAAHWsAABnrgAAWrAAAE6y
AABDtQAAOLgAAC+8BQAsvREAKr4cACm/JgAovzAAJ8A5ACbARAEkwE8BI8FbAiLBaQlgwXkC
H8KLAR7CnwEcwbQBGsDRARm+7wEYvP4CF7r/Axe6/wQXuv8EF7r/BLCUAACpngAAoKUAAJ
SqAACHrgAAeLAAAGqyAABctAAAULcAAES6AAA6vQAAMcEAACjFAAAfyQMAGc0JABbOEwA
Wzx4AFdAnABTRMgAT0T4AetJLABHTWgAQ02sAdTr+AA3TkWAL06kACdLDAafQ4wAlzvgAC
Mz/AAjL/wAlY/8ACMv/AKueAACjpgAAI6wAAIqWAAAB7tAAAbLYAAAF25AABQvAAARL8AADrDAAA
wxwAAJ8sAAB/PAAAX0gAAEdYAAAnaBAAG3AwABd0XAATeIQAD3ysAAuA4AAHhRgAA4IcAA
ONqAADjfgAA45UAAOosAADixQAA4uMAAOD0AADg+gAA4PoAAOD6AKWmAACZrQAAjLIAAH
62AABuuQAAxrwAAFDAAABExAAAOmGAAC7NAAAI0QAAHdUAABXZAAAO3AAAB98AAADiA
AAA6AEAAOkJAADpEgAA6hsAAOsIAADtMgAA70AAAPFSAADxZgAA8nwAAPKTAADyqQAA8r4
AAPLVAADy4QAA8uEAAPLhAJyuAACPtAAAgLgAAHC8AABgwAAAUMQAAEPJAAA3zgAALNM
AACLYAAAa3AAAEt8AAARiAAAE5QAAAOcAAADrAAAA8wAAAPQAAAD0AwAA9QsAAPYUAAD
3HgAA+SoAAPs5AAD+TAAA/mEAAP93AAD/jQAA/6AAAP+vAAD/twAA/7cAAP+3AJG1AACDug
AAcr4AAGLCAABRxwAAQ80AADbUAAAp2gAAIN4AABfiAAAO5gAABugAAADrAAAA7AAAAO4
AAAD1AAAA/QAAAP0AAD+AAAA/wAAAP8DAAD/DAAA/xcAAP8jAAD/MQAA/0QAAAP9ZAAD/b
QAA/4AAAP+PAAD/lgAA/5YAAP+WAP8Alwb/ACAF/wAgBf8AlgX/ACYE/wAzA/8AQAP/AE8C/wB
dAf8AaQH/AHUA/wB/AP8AhwD/AI8A/wCWAP8AnAD/AKIA/wCoAP4ArgD9ALUA/AC9APsAxwD6
ANUA+ADkAPcA8AD2APsA9AD/APEA/wDuAP8A7AD/AOWA/wDsAP8A7AD/AP8AIAb/AB0F/wAc
Bf8AHgT/ACME/wAwBP8APQP/AEsC/wBZAf8AZQH/AHEA/wB7AP8AhAD/AIwA/gCSAPwAmQD
6AJ8A+QCIAPgAqWd3ALEA9gC5APQAwwDzANAA8gDfAPAA7QDuAPkA7AD/AOKA/wDmAP8A
5QD/AOQA/wDkAP8A5AD/AP8AHQb/ABOF/wAYBP8AGQT/ACEE/wAsA/8AOQP/AEcC/wBVAf8A
YQH/AG0A/wB3AP0AgAD6AlcA9wCOAPUAIQDzAJSA8gChAPAApwDvAK0A7gC1AOWAvwDrA
MsA6QDaAOcA6gDIAPYA4wD/AN4A/wDcAP8A2wD/ANoA/wDaAP8A2gD/AP8GGQX/BxYE/wQ
UBP8AFgT/AB8D/wApA/8ANQP/AEIC/wBQAv8AXAH8AGgB+QByAPUAewDyAIMA8ACKAO0Ak
QDrAJcA6QCdAOgAowDmAKoA5QCxAOMAUwDhAMYA3wDWAN0A5wDaAPQA1QP/ANIF/wD
QBv8Azgb/AM0H/wDNB/8AzQf/AP8LFQX/DBAE/woNA/8NEwP/Dh0D/wslA/8IMAP/BT0C+wNKA
vYCVwHyAmMB7wNtAOWDdgDqA34A5wSFAOQEjADiBJMA4ASZAN4FoADcBqcA2wauANkHtwD

WCMQA1AjUANEL5wDMD/cAyBH/AMUS/wHCE/8BwRP/AcAT/wHAE/8BwBP/Af8REAT/EgoD/xl
HAv8YDwL/GhgC/xkiAv8VKwL3ETcC8BBEAesOUQHmDI0A4g9nAN8QcQDdEHkA2hGBANgRiA
DWEo4A1BOVANIUnADQFaMAzharAMwXtADKGMAAxxnQAMMb4gC/HPMBvB3/Arke/wK3Hv8
CtR7/A7Ue/wO1Hv8DtR7/A/8WCgP/FwMC/x0EAf8iCgH/lxIB/yIcAfUfJQHsGzEB5Rk9Ad4ZSwHZ
GlcA1RtiANEdawHOHnMBzB97AcogggHIIYkBxiKQAcQilgHCl54BwCSmAb4lrwK8JboCuibJArcm
3AKzJ+4DsCj8BK4o/wSrKf8FqSn/Bagp/wWoKf8FqCn/Bf8aBQL/HQAB/yUAAP8qBQD/KwsA9SkT
AOomHQDgIlgA2CI2ANekRAHMJIACxyhbA8QpZQPbK0Dvyt1BL0sfAS7LIMEuS2KBLgtkQS2L
pkFtC6hBblvqgWwL7QFrjDCBaww1QapMekGpjH5B6ly/wehMf8loDH/CJ8x/wifMf8lnzH/CP8fAAH/
JAAA/y0AAPgxAADtMQIA6C4JAN8pEgDVKiAAzC4vAcUwPQPAMUoEuzJVBrgzXwa1NGcHszR
vCLE1dgivNX0lRjaECKw2iwmrNpMJqTecCac3pQmlN68Kozi8CqE4zgqeOeQLmzn1C5k4/wuXOP
8Llj/C5U3/wuVN/8LITf/C/8jAAH/KgAA+DMAAOQ3AADcOgAA1jkCANMyCADKNhoAwTkqAbs6O
AS1OkQGsTtPCK47WQqrO2ILqTtqDKc8cQyIPHgNozx/Dal9hg6gPY4Onj2WDpw+oA+aPqoPmD
63D5Y+yBCUPt8QkT7xEI8+/w+OPv8PjD3/D4w9/w6MPf8OjD3/Dv8nAAD/MAAA6DkAAN1BAADT
RAAAzUMAAMg+BADAPxUAuUIIAbJDMwStQj8lqEJKC6VCVA2hQlWpN0JkEJ1CaxGbQnIRmUJ
5EpdCgRKWQokTIEKSE5JCmxSRQqYUj0KyFlxCwxWKQtkViELuFYZC/RSEQ/8Tg0P/EoJC/xKC
Qv8SgkL/Ev8qAAD3NQAA4kAAANZIAADMTAAAxU0AAMBIAAC5RxAAsUohAKtLLwSISjsloEIG
DJxITw+ZSFgRIkdfE5RHZxSSRm0VKEZ0Fo5GfBeMRoQYi0aNGIIGlxmHRqIzhUauGYNGvhqBR
tMafkbrGX1H+xh7R/8Xe0b/FnpG/xV6Rv8Vekb/Ff8tAADrOgAA3UcAANBPAAADGVAAAv1QAALIR
AACyTgwAq1EeAKVSKwOfUTclmU9CDJVOSxCRTVMTjkbxFoxLYheJSmkZh0pwGoVKdxuDSn8
cgkqJHIBKkx1+Sp4efEqrHnpKuh53Ss8fdUrnHnRK+RxxSv8ac0r/GXJK/xhySv8Yckr/GP8xAADnQ
AAA2EwAAMtVAADCWgAAulsAALNYAACsVQgAplcaAJ9YKAKZVzQHIFU+DI9TSBCLUVAUh1B
XF4RPXhmCTmUbf05sHX1Ocx57TXsfeU2FIHdNjyF1TZsic02oInFNtyJvTcsjbU3klmxN9x9rTv8da
07/HGtO/xtrTv8ba07/G/w1AADjRQAA1FEAAMdaAAC+XwAAtmAAAK9eAACnWgQAoFwXAJpdJ
QGUXDEGj1o8C4IYRRCFV0UgVVUGH1UWxt6U2IdeFJoH3VRcCFzUXgicVCBJG9QjCRtUZgla
1GIJmIRtCZnUMgmZVDhJmRR9SJKUf8gZFH/HmRR/x1kUf8dZFH/HfI5AADgSgAA0FYAAMRfAA
C6ZAAAsmUAAKtkAACiXwAAM2EUAJZilwGQYS8Fil85CoRdQg9/W0oUe1IRGHdYWBt0VI4ecV
ZIIIG5VbCNsVHQIalR+JmhUiCdmVJUoY1SiKGFUsSIfVMUpXVTeKF1U8yVdVf8iXVX/IF1V/x9dVf
8fXVX/H+09AADcTgAAzFoAAMBjAAC3aAAAr2oAAKdpAACeZAAAI2UQAJFmIACLZiwEhWQ2C
YBiQA57YEgTdl5PF3JcVRtuW1wea1pilWhZaiRIWHImY1h7KGFYhilfWJlqXVigKltYrytZWMIqV1fc
KldY8idXWP8kV1j/IhY/yFYWP8hWFj/lepBAADYUgAAyV8AAL1nAAC1bAAArG4AAKRuAACZaQ
AAkmkMAIxrHQCHaioDgWk0B3tnPQ12ZUUScWNMFm1hUxtpYFoeZV9glmJeZyVfXW8nXFX4K
VpcgytYXJAsVlyeLFRcrSxSXMARUVzaK1Fc8ShRXP8IUlZ/I1Jc/yJSXP8iUlZ/IuZGAADUVwAAxW
MAALtrAACycAAAqnMAAKFyAACVbQAAjW4JAIdvGgCCbycCfG4yBndsOgtxakMQbGhKFWHnU
BpkZVceYGRell1jZSVZYm0oVmF2KIRhgStSYY0sUGGcLE5hqyxNYb4rS2HYK0th7yhMYP4ITGD
/JExg/yJMYP8ITGD/IuNLAADQWwAAwmgAALhwAACvdQAAP3cAAJ13AACQcgAAiHMFAlJzFgB
9dCQBeHMvBHJyOAltEAPaG9HFGNtThhfbFQdW2pbIFdpYiRUaGonUWdzKU5nfitMZ4srS2eZK
0lnqStlZ7wqRmfVKUZn7idGZv0IR2X/I0dl/yJHZf8iR2X/It5RAADMYQAAvmwAALV0AACsegAApH
wAAJp9AACKdwAAgngAAHt5EgB3eiEAcnksA214NQdod4MY3VFEV50TBZac1IaVnFZHIJwYCJ
Pb2giTG9xJ0lufChHbokpRm6YKURuqChDbronQW7UJkFt7SVBbfwjQWz/IkFr/yFBa/8hQWv/IIdX
AADHZgAAu3EAALJ5AACpfwAAoIIAAJeCAACIfwAAfH4AAHV+DQBwfx0AbIAPAmh/MgVjfsKXn
1CD1I7SRNVek8XUXIWG014XR5Kd2UhR3dvI0R2eiVDd4clQXEWJEB3pyQ/d7kjPXfTIj117CE8d
PwgPHP/Hzxz/x88c/8fPHP/H9JeAADCbQAAt3cAAK5/AACmhAAAnYgAAJOJAACGhwAAAdUAA

G2FBwBohhcAZYckAGGHLgNdhjcHWIU/C1SERg9Qg0wTTIJTF0iBWxpFgWMcQoBtHkCAeB8+
glUfPYCVHzyAph47gLgdOYDSDHd+7Bs3ffwbN3z/Gzd7/xs3e/8bN3v/G8xIAAC+cwAAs30AAKuF
AACiwwAAmY4AAI+PAACCjgAAcY0AAGWNAABfjhEAXI4eAFmPKQFVjllEUY46B06NQgtKjEkOR
otQEUOLWBRAimEWPoprGDyKdhg6ioQZOYqUGDiKpRc2irgWNYrRFTSI7BUyh/wWMYX/FjGE/
xYxhP8WMYT/FsVuAAC5ewAAr4QAAKeMAACfkQAAIZQAAIqVAAB8lQAAAbZUAAF+VAABVlggA
UpYXAE+XlwBNly0BSZc1BEaWPQZDIkUJQJVND2VvQ47IV4POZVoETeVdBE1IYIRNJWSED
OVpBAxlbcpMJXQDi+T7A4tkfwPLJD/ECuP/xErj/8RK4//Eb92AAC0ggAAq4wAAKOTAACbmAAAk
JoAAIOcAAB1nAAAZ50AAAFieAABMnwAARaAMAEogGgBBosQAP6EuAT2hNgI6oT4EOKFHBja
gUAczoFkJMaBkCTCgcQouoH8KLaGPCSyhoQgqoLQIKaDOByie6wcmnPwJJZv/CiSa/wskmv8LJ
Jr/C7iAAACviwAAp5QAAJ+aAACVngAAiKAAAHuiAABtowAAX6QAAFKIAABFpWAAO6kAADWq
DQAzqxkAMasjADCrLAAvrDUBLaw+ASusSAIqrFIDKKxeAyesawMlrHkDJKyKAyOsnQMhrLECH6
zLAh6q6QldqPsDHKf/BBum/wUbpv8FG6b/BbKJAACqIAAAo5wAAJmhAACNpAAAF6cAAHGoAA
BjqQAAVqsAAEmtAAA+rwAANLEAACq0AAAJtgoAlbcVACC3HwAfuCgAHrgyAB24PAAcuUcAG7I
TABq5YQAZuXAAF7qCABW6lgATuasAEbnEABC35QAQtfkAELP/ARCy/wEQsv8BELL/Aa2TAAC
mnAAAnKMAAJGnAACDqgAAda0AAGauAABYsAAAS7IAAEC0AAA2twAALLoAACO9AAAAbwAA
AE8MEAA3FDAAMxhcAC8YhAARHKwAJxzYACMhDAafiUQAGyGEABchzAATlhWACyJ0AAMez
AADF0AAAxOwAAMP6AADc/wAAwv8AAML/AKidAACgpAAAIKkAAlatAAB4sAAAaLIAAFq1AAB
MtwAAQLoAADa9AAAswAAAI8MAABrGAAASygAAC8wAAATQAQAA0QoAANITAADTHQAA1S
YAANYyAADYPwAA2k4AANpgAADadAAA24oAANqgAADZtwAA19MAANbqAADV9wAA1fcAAN
X3AKKIAACWqwAAia8AAHqzAABrtgAAW7kAAE28AABAvwAANMMAACrGAAAhgyAAGM4AAB
HRAAAJ1AAAAtYAAADaAAAA3AAAAAN4FAADgDQAA4hcAAOQhAADmLAAA6TkAAOtKAADsX
QAA7HIAAO2JAADtoAAA7bQAAOzJAADr3gAA694AAOveAJmsAACMsQAAfbUAAG25AABdvAA
ATcAAAEDEAAAZyAAAKc0AAB/RAAAW1QAADdkAAAAbcAAAA3gAAAOEAAADjAAAA5QAAAOg
AAADqAAAA7AYAAO8PAADxGgAA9CUAAPczAAD6RAAA+1kAAPxvAAD9hQAA/ZoAAP2pAAD
9tgAA/bYAAP22AI+yAACAtwAAb7wAAF+/AABOxAAAQMkAADPOAAAm1AAAHNoAABPeAAAK
4QAAAUQAAADmAAAA6AAAAOoAADsAAAA7gAAAPAAAADyAAAA9QAAAPcAAAD6BgAA/RI
AAP8eAAD/LAAA/z4AAP9TAAD/aAAA/3wAAP+LAAD/mAAA/5gAAP+YAP8AIAx/AB0E/wAcBP8
AHwP/ACQC/wAxAv8APwH/AEwA/wBaAP8AZwD/AHIA/wB8AP8AhAD/AIsA/wCSAP8AmAD/AJ0
A/gCjAP0AqQD8AK8A+wC3APoAwQD5AM0A9wDdAPQA6wDxAPYA7wD/AO0A/wDsAP8A6wD/
AOoA/wDqAP8A6gD/AP8AHQX/ABoE/wAZA/8AGgP/ACIC/wAuAv8AOwH/AEKa/wBWAP8AYwD/
AG4A/gB4AP0AgAD7AlcA+gCOAPkAlAD4AJ0A9wCfAPYApQD1AKsA9ACzAPMAvADyAMgA7w
DWAOWA5gDoAPMA5gD+AOQA/wDjAP8A4gD/AOEa/wDhAP8A4QD/AP8AGQT/ABYE/wAVA/8
AFQP/AB8C/wAqAv8ANwH/AEQA/wBSAP8AXgD7AGkA9wBzAPUAewD0AIMA8wCKAPEAkADw
AJYA7wCbAO4AoQDtAKgA6wCvAOoAtwDoAMIA5QDQAOEA4QDeAO8A2wD7ANkA/wDYAP8A
1gD/ANUA/wDVAP8A1QD/AP8AFgT/ABID/wAQA/8AEwL/ABwC/wAmAf8AMgH/AD8A/gBNAPgA
WQDzAGQA7wBuAO0AdgDrAH4A6QCFAOgAiwDmAJEa5QCXAOQAnQDiAKQA4ACrAN4Asw
DcAL0A2QDLANUA3ADRAOWAzwD4AM0A/wDLAP8AyAD/AMcA/wDHAP8AxwD/AP8GEQP/BQ
sC/wAIAv8EEQH/BBkB/wAIAf8ALQH9ADoA9gBHAO8AUwDpAF8A5QBoAOMAcQDhAHkA3wCA
AN0AhgDbAlwA2QCTANgAmQDVAKAA0wCnANEARwDPALkAzADHAMkB2QDGA+oAwwX5AM
AH/wC9Cf8AvAn/ALsJ/wC7Cf8Auwn/AP8LCwP/CgQC/wsEAf8QCwH/EBQA/w0eAPoIKADyBTM
A6wJAAOQBTQDeAVgA2gNjANgEawDUBXMA0gZ7ANAHgQDOB4gAzQiOAMsIIQDJCZwAxwqk
AMULrADDDLcAwQ7FAL4P2AC6EusAtxT6ALQV/wCyFf8BsBb/Aa4X/wGtF/8BrRf/Af8QBAL/EAA

B/xYAAP8ZBgD/GQ0A+hYXAO8SIQDmDiwA3gw5ANgMRgDTDIIAzhFcAMsTZgDIFG4AxxZ1AM
QXfADCGIMAwBmJAL4akAC8GpcAuxufALkcqAC3HLIArR3AALie0QCvH+YBrCD2Aakg/wKmlv8
CpCH/AqMh/wOJlf8DoyH/A/8VAAH/FwAA/x4AAP0hAAD1IAUA7xwNAOQYFwDaFCMA0RgxAMs
cPwDFHkwAwSBXAL4iYAC7I2gBuSNwAbckdwG1JX0BtCWEAbImiwGxJ5IBryebAa0opAGrKK0C
qSm6AqcpywKkKuEDoSryA54r/wScKv8Emyr/BZop/wWZKf8FmSn/Bf8ZAAD/HgAA+iUAAOcoAA
DfKQAA2iUDANcdCwDNIRwAxSYrAL8pOQG6KkYBtitRARmsWgOwLWMDri5qA6wucQSql3gEq
S9/BKcwhgSmMI0EpDGWBalxnwWgMqkFnjK1BZwyxQWaMtsGlzLvB5Qy/QeSMv8HkTH/B5Ax/
weQMf8HkDH/B/8dAAD/JQAA6S0AAN40AADVNgaAzzMAAMotBgDDLRYAuzEmALUzNAGwNE
ADrDVLBKk1VQWmNI0GpDZIB6I2bAegN3MInjd5CJ03gQibOlglmTiRCZg4mgmWOKUJIDmwCpl
5wAqQOdQKjTnrC4o4+wulOP8Lhzj/CoY4/wqGOP8Khjj/Cv8gAADzKwAA4zYAANY9AADNQAAA
xj8AAMA6AQC6NxEAjsHAKw8LwKnPDsEozxGBp88UAicPVgJmj1gCpg9ZwuWPW4MID10DJM
9ewyRPYMNjz2MDY49IQ2MPaAOij6sDog+uw+GPc8PgZ3nD4E++A9/Pf8Ofj3/Dn09/w19Pf8Nft3/
Df8IAADrMgAA3D4AANBFAADGSQAavkgAALhDAACyQAwAq0IdAKVEKwKgRDcFm0NBCJdDS
wqUQIMMkUJbDY9BYg6NQWkPi0FwEiIBdxCIQX8RhkGHEoRCkRKCQpwSgEKoE35Ctn8Qso
UeULjFHhC9hN2Qv8SdUH/EXVB/xB0Qf8QdEH/EP0pAADmOQAA1kQAAMpMAADAUAAAuVAA
ALJLAACrRwcApUkZAJ9LJwGZSjMFIEk9CJBIRwuMR08OiUdXEIdGXhGERmQSgkVrE4BFchR/
RXoVfUWDFntFjRZ5RZgXd0aIF3VFsxhzRcYYcEXfGG9F8xZuRf8VbUX/FG1F/xNsRf8TbEX/E/lu
AADhPwAA0koAAMVSAAC8VgAAtFYAAK1TAACmTgMAN08VAJIRJAGTUDAEjk86CIINQwyFTE
sPgktTEX9LWhN9SmAVeklnFnHjhd2SXYydeI/GXJJiRpxSZUbb0miG2xJsBtqScIcaEnbHGdJ8R
pmSf8YZUn/F2VJ/xZISf8WZUn/Fu4zAADdRAAAzU8AAMFXAAC4WwAAsFsAAKIZAACgUwAAM
VQSAJRWIQCOVS0DiVQ3B4RTQAt/UUgPFBPEnhPVhR1TI0Wc01kGHFNaxpvTHMbbUx8HGt
Mhh1pTJleZ02fHmRMrR9iTL8fYEzYH19M7x1ftf4bX0z/GV5M/xheTP8YXkz/GOs4AADZSAAAYl
QAAL5cAAC1YAAArWAAAKVeAACbWAAAIgOAI9aHgCJWioChFk0Bn5XPQt6VkUPdlRMEnJT
UxVvUloXbFFgGWpRaBtUHAAdZVB4H2NQgyBhUI8gX1CcIV1QqyFbUL0hWVDVIVhQ7R9YUP0
dWFD/G1hQ/xpYUP8aWFD/Guc9AADVTQAAxIkAALtgAACyZAAAqmUAAKFjAACXXQAAj10LAI
peGwCEXycCf14yBXlcOwp1WkMOcFIKEWxXUBVpVlcXZIVeGmNVZR1hVG0fX1R2IFxUgCJaVI
wiWVSa1dUqSNVVL0jU1TSI1JU7CFSVPweUIT/HFJT/xtSU/8bUIP/G+RCAADSUQAAw10AALhk
AACvaAAAp2oAAJ5oAACTYgAAi2IHAIrIGAB/YyUBemlvBHVhOAhwX0ANa15HEWdcThRkW1U
XYFpcGI5ZYh1bWWogWFhzIVZYfSNUWlokUliYJFFYpyRPWLgkTVjQI0xY6iFNWPsfTVf/HU1X/x
xNV/8cTVf/HOBGAADOVgAAwGEAALZoaACtbAAApG4AAJttAACOZwAAhmYDAH9nFAB6aCIA
dWctA3BmNgdrZT4LZ2NFEGNiTBRfYVMXW2BZGlfYB1VXmggU11xIBdeyNOXYckTV2WJEtdp
SRJXbYjSF3OI0dd6SFHXPofR1z/Hkdc/xxHW/8cR1v/HNxLAADKWwAAvWUAALNsAACqcQAAo
nMAAJhzAACKbAAAAGwAAHIsEQB1bR8AcG0qAmtsMwZnazsKYmIDDI5oSRJaZ1AWVmZXGV
NIXhxQZGYfTWNulUtjeSJY4UjR2OUI0ZjoyJEY7UiQ2PMIUIJi6CBCYvQmH/HUJh/xxCYP8cQ
mD/HNhRAADGYAAAumoAALBxAACndgAAAn3gAAJV4AACGcwAAenEAAHNxDABuchsAa3MnA
WZyMARicTkIXXBADFlvRxBVbk0UUm1UF05sWxpLa2MdSGpsH0ZqdyBEaoMhQmqSIEFqoiBAa
rMfPmrLHj1p5x09aPkcPWf/Gz1m/xs9Zv8bPWb/G9JXAADCZQAAtm8AAK12AAClewAAAnH4AAJJ
+AAEEgAAAngAAG14BwBneBcAZHkjAGB5LQJceDUGWHc9CIR2RA1QdUsRTXRRFEI0WRdG
c2EZQ3JqG0FydRw/coEdPnKQHD1zoRw7c7IbOnLKGjlx5hk4cPgZOG//GTdu/xk3bv8YN27/GMx
eAAC9awAAs3UAAKp8AACigQAAMlQAAI6EAACBggAAcoAAAGV/AQBffxEAXIAeAFmAKQFWg
DIEUn85B05/QQpLfkGNR31PEER8VhNBfF8VP3xoFj17cxc7e4AYOnyPFzh8nxY3fLEVNNvJFTV6
5RQzefgVMnf/FTJ2/xUydv8VMnb/FcZIAAC5cgAAr3sAAKeCAACehwAAIYoAAluKAAB9igAAbYgA

AGCHAABXhwoAU4gYAFGIllwBOiC0BS4g1BEilPQZEh0QQJQYdLCz+GUw48hlwPOoVmETiFcRE
2hX4RNYaNETSGnhAyhrAPMYXIDjCE5Q4ugvcPLYH/ECx//xEsf/8RLH//EcBtAAC0eQAAq4IAAK
OJAACbjgAAkpAAAIWRAAB4kQAAaJAAAFqPAABOkAEASJAQAEaRHABEkSYAQpEvAUCRNw
M9kT8FO5BHBjiQTWg2kFkJNJBjCjKQbgswkHwLL5CLCi6QnQoskK8JK5DGCCqP5AgojPcJJ4v/
CyaK/wslif8MJYn/DLp1AACwgQAAP4oAAKCQAACXIQAAjJYAAH+XAABxIwAAYpcAAFOXAABI
mAAAPpkGADqaEwA4mh4AN5snADWbMAAzmgBMZtBAi+bSgMtm1MELJteBSqbagUpm3gFJ5
ulBCabmgQIm6wDI5vEAyKZ4gMgl/cEH5b/BR6U/wYeIP8HHpT/B7R/AACriQAAPJEAAJyXAACR
mwAAhZ0AAHeeAABpngAAWp8AAE2gAABBoQAAN6MAAC2kBgAppRIAKKUcACeIJQAmpI4AJ
aY3ACSmQQAipksAlaZXASCMYwEep3IBHaeCARunlQAapqgAGKa/ABal3wAWo/YBFaH/ARSg/
wIUn/8CFJ//Aq+IAACnkgAAoJkAAJaeAACJoQAafKMAAG2kAABfpQAAUaYAAEWoAAA6qQAAL
6sAACatAAAerwIAGLANABewGAAWsSEAFbEqABSxMwATsj4AErJKABGyWAAPsmcADbJ4AAu
yiWJsqAAB7G1AAaw0gAFr+0ABq39AAas/wAGrP8ABqz/AKqSAACjmwAAmaEAAI2kAACApwA
AcagAAGKqAABUrAAAR60AADyvAAAXsQAAJ7QAAB62AAAWuAAAD7oAAAe8CQADvRMAA70
dAAK9JgABvjAAAL48AAC/SgAAv1kAAL9qAAC/fQAAPvIAAL6oAAC9vwAAvN0AALvxAAC6/QAA
uv8AALr/AKWcAACdogAAkacAAIOqAAB0rQAAZa4AAFAxAABJswAAPLUAADK3AAAnugAAHrw
AABa/AAAOwQAABsQAAADHAAAAYAgAAMkQAADKGQAAYyIAAMwsAADOOAAAz0cAAM9ZA
ADPbAAAz4EAAM+XAADPrQAAsYAAM3gAADN8AAAzPMAAMzzAJ+kAACUqQAahq0AAHew
AABnsgAAV7UAAEm4AAA8uwAAMb4AACfAAAAAdwwAAFMYAAAZJAAAEzAAAAM8AADRAAA
A0wAAANUCAADYCGAA2hMAANwcAADfJwAA4jMAAOVCAADIVgAA5WwAAOWDAADkmwAA5
K8AAOTEAADj2QAA494AAOPeAJaqAACJrwAAerlAAGq1AABZuQAASrwAADzAAAwxAAAjcc
AABvLAAASzgAACdIAAAHVAAAA2AAAANoAAADdAAAA3wAAAOIAAADkAAAA5wIAAOkLAADt
FQAA8CEAAPMtAAD3PQAA+FIAAPhpAAD4gQAA95gAAPeqAAD2ugAA9r4AAPa+AlywAAB9tA
AAbLkAAfY8AABLwAAAPcUAAC/JAAAkzgaAGtMAABDYAAAG3AAAAN8AADhAAAA4wAAAO
UAAADnAAAA6QAAAOwAAADuAAAA8QAAAPQAAAD3AgAA+g0AAP4aAAD/JwAA/zgAAP9OA
AD/YwAA/3kAAP+MAAD/mwAA/54AAP+eAP8AHAT/ABkD/wAZAv8AHAL/ACIB/wAvAP8APAD/A
EoA/wBYAP8AZAD/AG4A/wB3AP8AfwD/AIcA/wCNAP8AkWd+AJkA/gCeAP0ApAD8AKoA+gCxA
PcAuQD0AMUA8gDTAPEA5QDwAPMA7gD/AO0A/wDrAP8A6QD/AOcA/wDmAP8A5gD/AP8AG
QP/ABYD/wAUAv8AFgH/AB8B/wArAP8AOAD/AEYA/wBUAP8AYAD/AGoA/QBzAPwAewD6AIMA
+QCJAPgAjwD3AJUA9gCaAPUAoADzAKYA8QCtAO4AtQDrAL8A6gDNAOgA3wDmAO8A5QD7
AOMA/wDgAP8A3QD/ANwA/wDbAP8A2wD/AP8AFQP/ABIC/wAPAv8AEgH/ABwB/wAoAP8ANA
D/AEIA/gBPAPoAWwD3AGUA9QBvAPMAAdwDyAH4A8ACEAO8AigDuAJAA7QCWAOSAnADpAK
IA5gCoAOMAsADhALoA3wDHAN0A2QDbAOoA2QD3ANQA/wDSAP8A0AD/AM8A/wDOAP8Azg
D/AP8AEQP/AA0C/wAKAf8AEAH/ABkA/wAkAP8ALwD7ADwA9QBJAPIAVQDuAGAA7ABpAOoA
cgDoAHkA5gB/AOQAhgDjAlS4QCRAN4AlwDcAJ0A2QCkANYArADUALUA0gDBAM8A0gDNAO
QAygD0AMcA/wDFAP8AwwD/AMIA/wDCAP8AwgD/AP8ACwL/AAYB/wAFaf8ADAD/ABUA/wAfA
PwAKgDyADYA6wBDAOgATwDkAFoA4QBkAN4AbADbAHMA2QB6ANYAgADUAIYA0gCMANA
AkgDOAJkAywCgAMkAqADHALEAxQC9AMMAzADAAOEAvQDwALsA/AC4AP8AtwD/ALYA/wC0
AP8AtAD/AP8FBQL/AQAB/wQAAP8GBwD/BRAA/wAZAPMAJADmADAA4AA8ANwASQDYAFQ
A0wBeANAAZgDNAG4AywB1AMkAewDHAIEAxQCHAMMAjgDBAZUAwAGcAL4CpAC8Aq4AuQ
O5ALcFyQC1B94AsgnxAK8L/wCsDP8Aqg7/AKkO/wCoDv8AqA7/AP8JAAH/BwAA/w0AAP8QAA
D9DQgA9AkRAOkCHADcASgA1QI1ANAEQgDLBk0AxwdXAMQJYADBCmgAvwxvAL0Ndgc7DX
wAug6DALgPigC3EJEAtRGZALMSogCxESArxO3AK0UxwCqFdwApxfwAKMZ/gChGf8Anxn/AZ4

Z/wGeGf8Bnhn/Af8OAADE/EQAA/hYAAO4XAADjFAAA3w0GANoHEQDSC8Ayg4tAMQTOgC/FU
cAuxdRALgZWwC1GmMAstxtqALEbcQCwHHcArh1+AK0ehQCrHowAqR+UAKgfnQCmIKcApCGy
AKIhwgCgltYAnCPrAZkj+wGXl/8CISP/ApQi/wKUIv8DlCL/A/8SAAD/GAAA6x4AAOEKAAADZJAAA
0iAAAM4XCADGGBkAvh0nALkgNQC0lKEAsCRMAK0lVQCrJl4BqSZIAacnbAGIKHlBpCh5AalpgA
GgKYcBnyqPAZ0qmQKbK6MCMsUuApcrvQKVk9ACkivnA48r+ASNK/8Eiyv/Bloq/wWKKv8Fiir/Bf
8VAADyHwAA5CoAANgwAADOMgAAyC8AAMloAwC8JRMAtSgiAK8rLwCqLTwBpi5HAqMvUAK
hL1gDnjBgA5wwZwObMW0EmTF0BJgxewSWMYIEIDKLBjMyIAWRMp8FjzKqBY0zuAWLM8sGi
DPjBoUz9QeDMv8HgTL/B4Ay/weAMf8HgDH/B/8bAADrKQAA3TQAANA6AADGPAAAvzsAALk1
AACzLw0ArDIdAKY1KwCiNjcCnTZCA5o3SwSXN1MFITdbBpM3YgaRN2kHjzdvB443dgeMN34Iji
GClk4kAiHOJoJhTimCYM4tAmBOcYKfjjeCnw48wp6OP8KeDf/Cnc3/wp3N/8Kdzf/CvchAADIMQA
A1jwAAMICAAC/RQAAuEMAALI+AAcROqCpTsZAJ89JgCaPTICIT09BJI9RgaPPU8HjDxWClo8
XQmIPGQKhjxrC4Q8cguDPHkMgT2CDH89iwx9PZYNez2iDXk9sA53PclOdT3aDnI98A5xPP8Nc
Dz/DW88/w1vPP8Mbzz/DPAnAADgOAAA0EIAAMRJAAC6TAAAs0sAAKxGAACIQQIANkEUAJhD
lwCTRC4CjkM5BYpCQgeHQkoJhEFSC4FBWQx/QWANfUFmDntBbQ56QXUPeEF+EHZBhxBOQ
ZIRckGfEXBBRJuQb4SbEHVEmpB7RJoQP0RZ0D/EGdA/w9nQP8PZ0D/D+wuAADbPQAAy0gA
AL9PAAC2UgAArIEAAKdNAACfSAAAmEcRAJJHwCNSSsCiEk1BYRIPwiAR0cKfUZODHpGVQ
53RVwPdUViEHNEaRFxRHESb0R6E25EhBRsRI8UakWcFWWhFqhVIRboVY0TSFmJE6xVhRPwT
YET/EmBE/xJfRP8RX0T/EegzAADWQwAAx00AALxUAACzVwAAq1cAAKJUAACaTgAAkkwMAIx
OHACHTigBgk4yBH5NOwd6TEMKdktLDXNKUQ9wSVgQbklfEmxIzhNqSG4VaEh2FmZlgBZkSlw
XYkiZGGBIpxheSLgYXEjOGFpl6RdaSPoWWUj/FFIH/xRZR/8TWUf/E+U5AADSRwAAw1IAALIYA
ACwWwAAp1wAAJ9ZAACVUwAAjVEJAIdSGQCCUyUBfVMvBHhROAd0UEAKcE9IDW1OTg9qT
VURZ01cE2VMYxvITGsWYExzGF9MfRldTIkaW0yWGIImpRpXTLUaVUzMGIRM5hITS/kXU0v/FI
NL/xVTS/8VU0v/FeE9AADPTAAAwFYAALZcAACTYAAApWEAAJteAACRWAAAIiFYFAIJWFgB9V
yIAeFctA3NWNgZuVT4JalNFDGdSSw9jUVIRYVFZFF5QYBZcUGgYWiBwGVhPextWUIYbVFCU
HFNQoxXRULMcT1DJHE1P5RtNT/gZTU//F01P/xZNTv8WTU7/Ft5CAADLUAAAvVoAALNhAACq
ZAAAomUAAJhJAACNXQAAg1sCAHxbEgB4XCAAc1wqAm5bMwVpWjslZVhCDGJXSQ9eVIARX
FZWFFIVXhZWVGUZVFRuGIJUeBxQVIQcTISSHU1UoR1LVLEcSVTHHEhU4xtIU/cZSFP/GEhT/
xdIUv8XSFL/F9pGAADHVQAAu14AALFIAACoaQAAAn2oAAJZoAACJYgAAfI8AAHdfDgByYB0Ab
mAnAWlgMQRIXzkHYV5AC11dRw5aXE4RV1tUFFRaWxZRWmMZTllsGkxZdhxKWYlCvSvmQHU
dZnxxFWbAcRFnFG0NZ4RtCWPYZQlj/GEJX/xdCV/8XQlf/F9VLAADeWQAAuGMAAK5pAACmb
QAAAnW8AAJNuAAcEzWAAeWUAAHFICwBsZRkAaGYkAWRmLgNgZTYGXGQ+CVhjRQ1VYks
QUmFSE09gWRVMX2EYSV9gQUdfdBtFX4AbQ1+OG0JfnhtBX64aP1/EGj5f4Bk9XvUYPV3/Fz1c
/xc9XP8WPVz/FtBQAADAXgAAfWcAAKtuAACjcgAAmnQAAJBzAACCBwAAAdWwAAGtqBgBmax
UAYmwhAF9sKwJbazMEV2o7CFNqQgtQaUkOTWhPEUpnVxNHZl4WRGZnF0JmchIAZn4ZP2a
MGT1mnBk8Zq0YO2bCFzlm3xY5ZPQWOGP/Fjhi/xU4Yv8VOGL/FctXAAC8YwAAsm0AAKlzAAC
geAAAl3oAAI15AAB/dgAAcXMAAGVxAQBecRAAW3ldAFhyJwFVcjADUnl4Bk5xPwhLcEYLSHB
NDkVvVBBCblwSP25IFD1ucBU8bnwVom6KFTlumhU4bqwUNm7BEzVu3hI0bPQTM2r/EzJp/xMy
af8TMmn/E8VdAAC4aQAArnlAAKZ5AACdfgAAIIAAAlqAAAB8fgAAbXsAAGB5AABXEAAsAU3kYA
FF6lwBOeiwBS3o0A0h5PAZFeUMIQnhKCKB4UQ09d1oOO3djEDI3bhA3d3oRNXeJEDR3mRAzd
6sPMXfADjB23Q0vdPMOLXP/Dy1y/xAscF8QLHH/EMBKAAC0cAAAq3kAAKOAACahAAAKYYAA
leGAAB5hQAAalMAAFuCAABPgQMASoESAEiCHQBGgicARIlvAUGCNwM/gj8FPIFGbjqBTgg3g
VcKNYFgCzOBbAsygXgLMIGHCy+BmAotgakKLIG/CSqA3AgpfvMJKHz/Cyd7/wsnev8MJ3r/DLts

AACwdwAAqIAAAKCGAACXiwAAJo0AAIGNAABzjAAAY4sAAFaKAABJigAAQloJAD2KFgA8iyAA
OospADiLMQE2izkCNltBAzKLSgQwi1MFLotdBSyLaAYri3UGKouFBiiLIQUni6gEJYu9BCOK2gMii
PIFIYb/BiCF/wcghP8HIIT/B7Z0AACsfwAAPlcAAJyOAAACUkQAAiJMAAHuTAABskwAAXZIAAE+S
AABDkgAAOZMAADGUDAAvIBcALpUgACyVKQArITEAKpU6ACiVQwEnIU0BJZVYASSWYwliInE
CIZaBAR+WkgEelaUBHJW6ARqU1wAZkvEBGJH/AhiP/wMYj/8DGI//A7B+AACoiAAAoY8AAJmV
AACOlwAAgZkAAHOZAABkmQAAVpoAAEmaAAA9mwAAMpwAACmeAAAhnwsAH58VAB6gHg
AdoCcAHKAvaBugOQAaoEMAGKBOABehWwAWoWkAFKF5ABOhiwARoJ8ADqC0AAuf0AAMn
e0ADJv+AAya/wAMmf8ADJn/AKuHAACkkAAAnZcAAJObAACGngAAeJ8AAGmgAABaoAAATaE
AAEGiAAA1pAAAK6UAACKnAAAZqAAAEqpHAA2qEQAMqxoAC6sjAAqrLAAJqzcACKtCAAerUA
AFq14ABKtvAAKrgQAAq5UAAKqqaACpwQAAqOAAAKfzAACm/gAApv8AAKb/AKeRAACgmQAA
I54AAIqhAAB8pAAABAUAFA6mAABQpwAAQ6kAADiqAAAttrAAAI64AABqwAAASsQAACrMAAA
O0BwAAtRAAALUZAAC2IQAAtisAALc1AAC3QgAAAt1EAALdhAAC3cwAAAt4gAALedAAC2swAAAtc
4AALTtoAACz9gAAsvwAALL8AKOaAACaoAAAJqQAAICnAABWqQAAYasAAFKsAABErgAAOLEA
AC2yAAAJtQAAGrcAABK5AAAJugAAArwAAAC+AAAAvwYAAMANAADBFgAAwh4AAMMoAADF
MwAAx0AAAMdRAADHYwAAyHcAAMiNAADHowAAxroAAMbUAADF6AAAxPEAAMTxAJ2iAAC
RpwAAg6oAAHStAABjrwAAVLEAAEWzAAA5tgAALbkAACO7AAAZvgAAEMAAAAjCAAAAxAAA
AMcAAADJAAAAywAAAMwAAADOCAAA0BAAANIZAADVlWAA2C8AANs+AADcUAAA3WUAAN
56AADekQAA3qcAAN67AADezgAA3dsAAN3bAJOOAACGrAAAd7AAAGayAABWtQAARrgAADm
8AAAsvwAAIsIAABjFAAAOyAAABcsAAADOAAAA0AAAAANMAAADWAAAA2QAAANsAAADeAAA
A4QAAAOQHAADnEgAA6h0AAO0qAADwOgAA8E8AAPFIAADyfAAA8plAAPOmAADzswAA870
AAP09AlmuAAB6sgAAabUAAFi5AABlvQAAOcEAACzFAAAhyQAAFs0AAAzRAAAC1QAAANkA
AADcAAAA3gAAAOAAAADiAAAA5QAAAOcAAADqAAAA7QAAAPAAAAD0AAAA9wkAAPsWAA
D/lwAA/zUAAP9MAAD/YwAA/3oAAP+OAAD/nAAA/6QAAP+kAP8AGAP/ABYC/wAWAf8AGAH/A
CAA/wAsAP8AOgD/AEGa/wBVAP8AYAD/AGoA/wBzAP8AewD/AIIA/wCJAP8AjdD9AJQA+wCZA
PkAnwD3AKQA9QCrAPQAswDzAL4A8gDNAPAA3wDvAO8A7AD8AOkA/wDoAP8A5wD/AOcA/w
DmAP8A5gD/AP8AFAL/ABIC/wARAF8AEgD/AB0A/wApAP8ANgD/AEQA/wBQAP8AXAD+AGYA/
ABvAPoAdwD5AH4A+ACEAPYAigD0Al8A8gCVAO8AmgDuAKAA7QCnAOsArwDqALkA6ADGA
OYA2ADkAOoA4QD4AN4A/wDcAP8A2wD/ANoA/wDYAP8A2AD/AP8AEQL/AAwB/wAKAf8ADwD
/ABoA/wAIAP8AMgD+AD8A/ABLAPgAVwD2AGEA8wBqAPEAcgDwAHkA7QB/AOsAhQDpAlO5
gCQAOUAlgDjAJwA4gCjAOAAqgDeALQA3ADAANoA0ADVAOMA0gDzANAA/wDOAP8AywD/A
MoA/wDJAP8AyQD/AP8ADAL/AAcB/wAEAP8ADAD/ABUA/wAgAPkALAD1ADkA8gBGAO8AUQ
DrAFwA6ABIAOYAbQDjAHQA4AB6AN4AgADbAIUA2QCLANgAkQDWAJcA1ACeANIAPgDPAK8
AzQC6AMoAyQDHANwAxADuAMIA+wC/AP8AvgD/AL0A/wC8AP8AvAD/AP8ABgH/AAAA/wAAA
P8ACAD/ABEA+wAbAO8AJwDrADMA5wBAAOMASwDfAFYA3ABfANgAZwDUAG4A0QB0AM8A
egDNAIAAywCGAMoAjADIAJMAxgCaAMQAogDCAKoAvwC1ALwAwwC6ANUAtwDpALUA+ACz
AP8AsgD/AK8A/wCuAP8ArgD/AP8AAAH/AAAA/wAAAP8AAwD/AAsA8AAUAOUAIADgAC0A2gA
5ANUARQDRAE8AzABZAMkAYQDGAGgAxABvAMIAAdQDAAHsAvwCBAL0AhwC7AI4AuQCVAl
cAngC1AKcAswCxALEAvgCuANAArADmAKoA9QCnA/8ApQT/AKQE/wCjBP8AowT/AP8BAAD/A
AAA/wUAAPgEAADwAAIA4wALANsAGQDUACYAzgAyAMgAPgDEAEkAwABTAL0AWwC6AWM
AuAJqALYDcAC1BHYAswR8ALIFgwCwBooArgeSAKwImgCrCaQAqQmuAKcKvACkC88Aog3nA
J4Q+ACcEf8AmhH/AJkR/wCYEf8AmBH/AP8FAAD/CQAA7w0AAOURAADdDwAA2AgDANEBEA
DKAx0AwwYqAL0JNgC5DEIAtQ1MALIPVQCwEV0ArRJkAKwSawCqE3EAqBR4AKcVfgCIFYYAp

BaNAKIXlgCgF6AAnhirAJwZuQCaGcsAlxrxAJQb9gCSG/8AkBv/AY8b/wGOG/8Bjhv/Af8JAAD0EQ
AA5hsAANsgAADRIAAyhsAAMyTBQC/EBUAuBQjALIXMACuGjwAqhtHAKcdUACIHlgAox5fAK
EfZgCfIG0AniBzAJwhcgCbloEAmSKJAJcikgCWI5wAlCOoAJlktQCQJMcAjSTeAYok8gGHJP8Chi
T/AoQk/wKEJP8DhCT/A/8QAADrHQAAs3icAANetAADHLgAAwCoAALojAAC1HA4AriAeAKkjKwC
kJTcAoSZCAJ4nSwCbKFMBmSlbAZcpYQGVKWgBICpuAZlqdQGRKn0BjyuFAo0rjgKLK5gCiiykA
ogssQKFLMIDgyzZA4As7wR9LP8EfCz/BHsr/wR6K/8Feiv/BfQXAADIJwAA1jEAAMk3AAC/OAAA
uDYAALlwAACsKAgApioZAKAtJgCcLzIAmC89AZUwRgKSME8CkDBWA44xXQOMMWMDijFqBI
gxcQSHMXgEhTKABIMyigSBMpQFgDOgBX0zrQV7M74GeTPUBnYz7Ad0Mv0HczL/B3lx/wdxMf
8HcTH/B+8gAADfLwAAzzkAAMI/AAC5QAAAsj8AAKs6AACkMwMANjMUAJg1lgCUNi4BkDc4Aow
3QgOJN0oEhzdRBYQ3WAWCN18GgTdIBn83bAd9N3QHezd8B3o4hgh4OJAldjicCXQ4qglyOLoJ
cDjQCm046QprN/sKajf/CWk2/wlpNv8JaTb/CesnAADZNgAAyUAAAL5GAAC1SAAArUYAAKVBA
ACePAAAlzkQAJE7HgCNPSoBiD00AoU9PQSBPEYFjxNBnw8VAd6PFsleDxhCXy7aAp003AKc
jx4C3E8ggvtPI0MbTyZDGs8pwxpPLYNZzzMDWQ85g1jPPkMYjv/DGE7/wxhOv8LYTr/C+YtAADU
PAAAxUYAALpLAACxTgAAqU0AAKFIAACYQwAAKEALAltBGgCGQiYBgkIxAn5COgR6QUIGd0F
JCHRAUAlyQFcKcEBdC24/ZAXsP2wNaj90DmhAfg5nQlkPZUCWD2NAPBBhQLMQXkDJEFxA4x
BbP/cPWj//Dlo//w5aPv8NWj7/DelzAADPQQAawUsAALZQAACtUwAApVIAAJxPAACTSQAAi0UH
AIVGFwCARyMAe0ctAndHNgrR0Rj4GcEVGCG1FTQprRFMLaERaDWZDYQ5kQ2kPYkNxEGFDe
xFfRIYRXUSTEItEoRJZRLESV0TGE1VD4RJUQ/YRVEP/EFND/w9TQv8PU0L/D944AADLRgAAv
k8AALNVAACrWAAAolcAAJIVAACPTwAAhkoEAH9KFAB6TCAAdkwqAnFLMwRtSjsGakpDCWd
JSQpkSFAMYUhXDI9HXg9dR2YRXEduElpHeBNYR4MUVkeQFFRInxRSSK8UUEfDFE9H3hRO
R/QTTkf/Ek5G/xFNRv8QTUb/ENo9AADISgAAu1MAALFZAACoXAAAn1wAAJZZAACLVAAAgU8
AAHpPEAB1UB0AcFAoAWxQMqNoTzgGZE5ACGFNRgpeTE0MXExUDlIMWxBXS2MSVUtrE1N
LdRRRS4EVUEuOFU5MnRZMTK0WSkzBFUIL3BVIS/MUSEr/E0hK/xJISv8RSEr/EdZBAADETwA
AuFcAAK5dAACmYQAANWEAAJNeAACIWQAAfQAAHRUDABvVB0Aa1UIAWdULgNjVDYFX1
M9CFxSRAPZUUsMVIFRDIRQWBFSUGAST1BpFE1QcxVMUH4WSICMFkhQmxZHUKsWRVC/F
kNQ2hVDT/IUQ0//E0NO/xJDTv8SQ07/EtJGAADBUwAAtlwAAKxiAACjZQAAMmYAAJFkAAcEXg
AAAd1oAAG9ZCQBqWRcAZloiAGJaKwJeWTMEWlg7B1dXQglUV0gMUVZPDk9WVhBMVV4SSIV
nFEhVcBVGVXwWRFWKfKvNVmRZBVakVQFW9FT5V2RQ+VPEUPVP/Ez1T/xl9Uv8SPVL/Es1L
AAC+VwAAs2AAAKpmAAChagAAmGsAAI5pAACAYwAAc2AAAGleBQBjXhMAYF8fAF1fKQFZXz
EDVI44BIJdPwhPXUylTFxNDUpCVA9HW1wRRVtIE0NabhRBWnoVP1ulFT5blxQ9W6gUO1u8Ez
pb1xM5WvASOFj/EjhY/xl4V/8ROff/EclQAAC7XAAAsGUAAKdrAACebwAAIXAAAltAAB9agAAc
GcAAGRkaABdZA8AWWUbaFdlJQBUZS4CUGU1BE1kPQdKZEQJR2NKC0ViUg5CYloPQGFJE
T5hbRI8YXgSO2GGEjliih4YqcRNmK6ETVh1RA0YO8QM1/+EDNe/xAzXf8QM13/EMRWAAC3Y
gAARWoAAKVwAACcdAAAK3YAAI1AAB7cgAAbW8AAGBsAABWawoAUmsXAFBslQBNbCoBS2
wyA0hsOQVFaoAHQmtlCUBqTws9alcNO2lgDjlpag83aXYPNmmEDzRqlA4zaqYOMmq5DTBp1A
wvaO4NLmb+DS1l/w4tZP8OLWT/DsBcAAC0aAAAqnAAAKJ2AACZegAAkHwAAIZ8AAB4eQAAa
XYAAFx0AABPcgQASnlSAEhzHQBgdCYARHQuAUF0Ngl/cz0EPXNEBjPzTAg4clQJNnJeCjRya
AsycnULMXKDCy9ykwoucQJLHK4CSty0ggqcO0JKG79Ciht/wsnbP8LJ2z/C7tjAACwbGAAp3cAA
J99AACXgQAAjoMAAIODAAB0gQAAZH4AAFd8AABLewAAQnsLAD97FwA9fCAAPHwpADp8M
QE4fDkCNnxAAzR8SAQyfFEFMHxbBi58ZgYsfHIGK3yBBil8kQYofKMFJny2BCV70QQkee0FInf9
BiF2/wchdf8IIXX/CLZrAACsdgAApH4AAJyEAACUiAAAiokAAH2JAABvhwAAX4YAAFgFAABFhA
AAOoQDADSEDgAyhRkAMYUiAC+FKgAuhjlALYY7ASuGQwEphkwCKIZXAiaGYgllhm8CI4Z9Ai

KGjglghqABH4W0ARyFzwEcg+sBG4H8AhqA/wMaf/8EGn//BLFzAACofgAAoYUAAJmLAACRjgA
AhY8AAHePAABojgAAWY4AAEuNAAA/jQAANY0AACqOBQAljhAAJl8aACOPlgAijyoAIZAyACCCQ
OwAfkEUAHZBQABYQXAAbkGkAGZB4ABeQigAWkJwAE4+wABGOywARjekAEYv7ABGK/wEQif
8BEIn/Aax9AAClHgAAnY0AAJaSAACLIAAfZUAAG+VAABglQAAUpUAAESVAAA5lQAALpYAA
CSXAAAbmAQAFpkOABWZGAAUmiAAE5opABKaMQARmjsAEJpGAA6aUwAMmmEAC5pwAA
maggAHmpUABZmpAAOYwAACl94AApbzAAKV/wADIP8AA5T/AKiGAAChjwAAmpUAAJCYYAAC
DmgAAdJsAAGWbAABWnAAASZ0AADydAAAxngAAJp8AAB2hAAAVogAADaMEAAakDAACpB
YAAaUeAAGlJwAApTAAAKU7AACISAAApVYAAKVmAACldwAAPlsAAKSfAACjtQAAotAAAKHrA
ACf+AAAn/8AAJ//AKSQAACelwAAIJwAAleeAAB5oAAAaaEAAfqiAABMowAAP6QAADOIAAAop
wAAHqgAABaqAAANqwAABawAAACtBQAAArg0AAK4VAACvHQAAsCYAALAvAACxOwAAAsUkAA
LFZAACxawAAAsX4AALCUAACwqQAAr8EAAK7dAACt7wAArfoAAK36AKGZAACXngAAi6lAAH2k
AABtpgAAXacAAE6pAABAqgAANKwAACmuAAAFrwAAFBAAAyzAAAFtAAAAALYAAAC3AAAAu
AMAALkKAAC6EgAAuxoAALwjAAC+LQAawDkAAMBJAADBWWAAwW8AAMCEAADAmwAAwL
EAAL/KAAC+4QAAvu8AAL7vAJqgAACOpAAAgKcAAHCqAABGqwAAUK0AAEKwAAA0sgAAKbQ
AAB62AAAVuAAAC7oAAAAO8AAAAvgAAAMAAAADBAAAAwwAAAMUAAADGBQAAyA0AAMoV
AADNHwAA0CoAANQ3AADVSQAA1lwAANZyAADXiQAA16EAANa1AADVyQAA1d0AANXdAJG
mAACDqgAAc60AAGOVAABSsgAAQ7UAADW4AAAouwAAHr0AABTAAAAKwgAAAcUAAADHA
AAAYQAAAMsAAADOAAAA0AAAAANIAAADVAAAA2AAAAANsGAADeEAAA4hoAAOYmAADqNQ
AA7EgAAO1eAADudAAA7owAAO+gAADvsAAA774AAO++AlasAAB3sAAAZrIAAFW1AABEuQA
ANr0AACjBAAAdxAAAE8gAAAJLAAAAzgAAANEAAADTAAAA1gAAANoAAADcAAAA3wAAAOIA
AADIAAAA6AAAAOsAADuAAAA8QgAAPUVAAD4lwAA/DMAAP1IAAD+XgAA/3UAAP+JAAD/m
QAA/6YAAP+mAP8AFQL/ABIB/wASAP8AFAD/AB4A/wArAP8AOAD/AEUA/wBRAP8AXQD/AGY
A/wBvAP8AdwD+AH0A/ACDAPoAiQD4Al4A9wCUAPcAmgD2AKAA9ACnAPMARwDyALkA8QDG
AO4A2QDrAOsA6QD6AOgA/wDmAP8A5QD/AOUA/wDkAP8A5AD/AP8AEQH/AA0B/wAMAP8AD
gD/ABoA/wAnAP8AMwD/AEAA/wBNAP8AWAD9AGIA+wBrAPgAcgD1AHkA8gB/APEAhADvAlO
7gCQAO0AlQDsAJwA6wCjAOkAqgDoALMA5gC/AOIA0ADfAOQA3QD1ANsA/wDZAP8A2AD/AN
UA/wDTAP8A0wD/AP8ACwH/AAgB/wAFAP8ACwD/ABYA/wAjAP8ALwD8ADsA+gBIAPYAUwDz
AF0A8ABmAO0AbQDqAHQA5wB6AOYAgADkAlUA4wCLAOlAkQDgAJcA3gCeAN0ApgDaAK4A
1gC5ANMAyADran0AzgDvAMwA/QDKAP8AxwD/AMQA/wDCAP8AwQD/AP8ABgH/AAEA/wAA
AP8ACAD/ABIA+gAeAPUAKgDyADYA7wBCAOwATQDnAFcA4wBgAN8AaADcAG8A2gB1ANkA
ewDWAIAA1QCGANMAjADRAJIAzwCZAM0AoQDKAKkAxwCzAMUAwQDCANQAwADpAL4A+A
C6AP8AtgD/ALUA/wC0AP8AtAD/AP8AAAD/AAAA/wAAAP8AAwD8AAwA8AAYAOSAJADmADAA
4gA8AN4ARwDZAFEA0wBaANAAyGDOAGkAywBvAMoAdQDIAHsAxgCBAMQAhwDCAI0AwAC
UAL4AnAC8AKQAuQCuALcAuWc1AMwAsgDhAK8A8gCsAP8AqgD/AKkA/wCpAP8AqAD/AP8AA
AD/AAAA/wAAAP8AAADvAAUA5gASAOAAHgDaACoA1AA2AM4AQQDJAEsAxgBUAMMAXADA
AGQAvGbqALwAcAC6AHYAUB7ALcAgQC1AlgAswCPALEAlwCvAKAArQCqAKsAtgCoAMYAp
gDbAKMA7wChAPwAoAD/AJ4A/wCeAP8AnQD/AP8AAAD/AAAA9wAAAOkAAADjAAAA2wAKAN
MAFwDNACMAxgAvAMEAOgC9AEUAUGBOALcAVgC0AF4AsgBkALAAawCuAHAArQB2AKsAfQ
CqAIMAqACLAKYAkWckAJwAogCnAKACswCeA8lAmwTZAjKF7QCXB/wAIQf/AJMH/wCTB/8Ak
gf/AP8AAAD2AAAA6QkAAN4MAADVCGAAZwMAAMkADgDCABsAvAAANALcAMwCyAz4ArwVIAK
wGUACpB1gApwhfAKYJZQCKCmsAogtyAKELeACfDH8Ang2HAJwOkACaDpoAmA+IAJYQsQCU
EMIakhZAI4S7wCME/8AihP/AIKT/wCIE/8AiBP/AP8CAADtDwAA4BgAANMcAADJHAAAwhcAA

L0OAgC4BxIAsgsgAK0OLACoETcApRNCaKIUSwCfFVMAAnRZaAJsXYACZF2cAmBhtAJYYcwC
VGXsAkxmDAJEaiwCQGpYAjhuhAlwbrQCJHL4AhxzUAIQd6wCCHfwAgB3/AX4d/wF+Hf8BfRz/A
fUMAADmGwAA1iUAAMkqAADAQQAAsUAALMfAACuFQoAqBgaAKMbJwCeHTIAmx88AJgfRg
CVIE4AkyFVAJEhXACPImlAjiJpAlwibwCLI3YAiSN+AlckiACGJJIAhCWdAlIlqgB/JroBfSbPAXom6
AF4JfoCdIX/AnUk/wJ0JP8DdCT/A+8XAADfJQAAzi8AAMI0AAC5NAAAsjEAAKsrAACIlwQAnyIVA
JklIgCVJi0Akig4AI8oQQCMKUKAiilRAYgqVwGGKI4BhCpkAYMrawGBK3IBfyt6An0sgwJ8LI4Ceiy
aAngtpwJ2LbYDcyzLA3Es5QNuLPgEbSv/BGwr/wRrKv8Eayr/BOkgAADYLQAAYDcAAL08AAC0P
QAARdsAAKU1AACdLwAAIyoQAJEsHQCnLiAiS8zAIYwPQGDMEUcGTBMAn4xUwJ8MVoDezF
gA3kxZwN3MW4EdjF2BHqygARyMooFcDKWBW4ypAVsMrMFajLHBmgY4QZIMvYGZDH/BmMx/
wZjMP8GYjD/BuQnAADSNAQAawz4AALhCAACvRAAAp0IAAJ89AACXOAAAKDILAlOzGQCGNS
UAgjYvAX42OAJ7NkEDeDZIA3Y2TwR0NIYFcjZcBXA2YwZvNmoGbTZyB2s2fAdpN4cHaDeTCG
Y3oQhjN7AIYtFDCV833gldN/QJXDb/CVs1/whbNf8IWzX/COAtAADNOwAAv0MAALRIAACsSgAA
o0kAAJtEAACSPwAAiTkGAIM5FQB/OyEAezwsAXc8NQJ0PD0DcTtEBG47SwVsO1IGajtYB2g6X
whmOmclZDtvCWM7eAphO4MKXzuQCI07ngtbO60LWTvAC1c72wxWO/ILVTr/C1Q6/wpUOf8KV
Dn/CtszAADIQAAAvEgAALFNAAcCoTwAAoE4AAJdKAACORQAaHd8CAH0/EgB4QB4AdEAoAX
BAMQJtQDkEakBBBWc/SAZIP04HYj9VCGA/XAlfPmMKXT9sC1s/dQxaP4AMWD+NDVY/mw1UP
6sNUj+9DIA/2A5PP/ANTj7/DE4+/wxOPf8LTj3/C9c4AADFRQAauU0AAK5SAACmVAAAnVQAAJ
RQAACKSgAAf0UAAHdDDgByRBsAbkUIAGpFLgJnRDYDY0Q9BWFERAdEQ0sIXENRCVpDWQ
tYQmAMVkJpDVRDcg5TQ30OUUOKD09DmQ9OQ6kPTEO7D0pD1Q9JQ+4OSEL/DkhB/w1IQf8
NSEH/Ddl8AADCSQAAtIEAAKxWAACjWAAAmIlgAAJFVAACGUAAAE0sAAHJICwBtSRgAaUkjA
GVJLAFhSTMDXkg6BVtlQQZYR0glVkdOCIRHVgtSR10NUEdmDk5HcA9NR3sQS0eIEEIHlxBIR6
cQRke5EERH0xBDR+0PQ0b+DkNF/w5CRf8NQkX/Dc5BAAC/TQAAs1UAAKpaAAChXQAAMF0
AAI5aAACCVAAAd1AAAG1NBwBnTRUAY04gAF9OKQFcTjECWU04BFZNPwZTTEUIUUXmCk5
LUwtMS1sNSktkDKILbQ9HS3kQRUuGEERMIRBCTKUQUy3ED9M0RA+S+wPPkr9Dz1J/w49Sf
8OPUn/DspGAAC8UQAAsVkaAKhfAACfYgAAImIAAIfAAB/WgAAc1YAAGhSAwBhUHEAXVMdA
FpTJgBXUy4CVFM1BFFSPAZOukMITFFKCUIRUQtHUVkNRVBhDkNQaw9BUHYQQFCEED9R
kxA9UaMQPFG2DzpRzw85UOsOOE/8DjhO/w44Tf8OOE3/DsdKAAC5VQAARl4AAKVjAACdZgAA
IGcAAIIIAAB7XwAAb1wAAGNYAABbVw0AV1gZAFVYIwBSWSsBT1gzA0xYOgVJWEEHR1dHCU
RXTwpCVIYMQFZfDT5WaQ48VnUPO1aCDzpXkQ84V6ION1e0DjVXzQ00VukNM1T7DTNT/w0z
Uv8NM1L/DcNPAAC2WgAArGMAAKNoAACabAAAKWwAAIdrAAB5ZwAAbGMAAGBgAABVXQk
AUV4WAE5eIABMXygASV8wAkdeNwNEXj4FQI5FB0BdTak9XVQKO11dCzldZww3XXMNNI2AD
TVdkAwzXaEMMI2zCzBdzAsvXOGLLIr7Cy1Z/wwtWP8MLVj/DL9VAACzYAAAqWgAAKBuAACyC
QAAj3IAAIRxAAB3bgAAaWsAAFXoAABPZAQASmQRAEdIGwBFZiQAQ2YsAUfMNAI/ZjsDPGV
CBTPlSgY4ZVIIInmVbCTRIZQkyZXEKMwV/Ci9ljgkuZZ8JLWWyCCtlygcqY+cHKWH6CShg/wknX/
8KJ1//CrtbAACvZgAApm4AAJ50AACWdwAAjHgAAIJ4AAB0dQAAZXIAAFhwaABLbQAAQmwLA
D9sFgA9bSAAPG0oADpuMAE4bjcBNm4+AzRuRgQybU8FMG1YBS5tYwYtbW8GK218BiptjAYob
Z4FJ22wBCVtyQQkbOYEI2r5BiJo/wciZ/8HIWb/B7ZiAACsbQAAo3QAAJt6AACTfgAAin8AAH9/AA
BwfAAAYHoAAFN4AABHdgAAO3UEADZ1EAA0dRoAMnYiADF2KgAwdjIAlN6ASx2QgErdksCK
XdVAih3YAMmd2wDJXd6AiN3iglid5wCIHavAR52xwEddOUBHHL5Ahtx/wMbcP8EG2//BLJqAACp
dAAAoHwAAJmBAACRhQAaH4YAAHqFAABrgwAAW4EAAE6AAABBfwAANn4AACx+CAApfhM
AJ38cACZ/JAAIfysAJIAzACOAPAAigEUAlIBQAB+AWwAdgGgAHIB2ABqAhwAZgJkAF4CsABV/x
AATfuMAFHz4ABN6/wETef8BE3n/Aq1yAACIfAAAnYMAAJaJAACOjAAAgYwAAHOLAABkigAAV

YkAAEelAAA8iAAAMlgAACaIAAAeiAkAG4gTABqJGwAZiSMAGlkrABeKNAAWij0AFIplABOKVA
ASimEAEIpwAA6KgQAMipMAComnAAilvQAHh9oAB4byAAeE/wAlg/8ACIP/AKI8AACihAAAmosA
AJOQAACIkQAAepIAAGuRAABckQAATpAAAEQCQAAA0kAAAKpEAACCRAAAAXkgAAD5MIAAuT
EQAKkxoACZQIAAiUKwAHIDQABpQ/AASUSwADIFkAAZRoAACUeQAAk4sAAJOfAACStAAKc4
AAI/qAACP+AAAjv8AAI7/AKWFAACejQAAMJMAAI2WAACAlwAAcZgAAGGYAABTmAAARZgAA
DiYAAAasmQAAlpoAABmbAAARmwAACJwCAAGdCgAAnhIAAJ4aAACelgAAAnysAAJ81AACfQQA
An04AAJ9eAACfbwAAAnIAAJ6XAACdqwAAAnMQAAJvhAACa8gAAmfWAAJn/AKGPAACblgAAkZ
kAAIScAAB1nQAAZp4AAFAeAABInwAAO6AAAC+hAAAkogAAGqMAABGkAAAjPQAAAKYAAA
CnAwAApwoAAKgRAACpGQAAqSEAAKoqAACrNAAAq0IAAKtRAACrYwAAq3UAAKuKAACqoQ
AAqbgAAKjTAAcN6gAAp/YAAKb5AJ6YAACVnAAAIj8AAHmhAABqowAAWqQAAEulAAA9pgAA
MKgAACWpAAAaqwAAEawAAAitAAAAArgAAAK8AAACxAAAAsQAAALIGAACzDQAAtBUAALYe
AAC3JwAAuTMAALICAAC6VAAAumcAALp8AAC6kwAAuakAALnAAAC42gAAt+sAALfwAJefAAC
LogAAfaUAAG2nAABdqAAATKoAAD6sAAAwrgAAJbAAABqyAAARswAAB7UAAAC3AAAAuAAA
ALkAAAC7AAAAvAAAAAL4AAADAAQAAwQgAAMMRAADGGgAAySQAAM0wAADOQQAAziYAA
M9rAADPggAAz5oAAM+uAADPwgAAz9UAAM/dAl6IAACAqAAAcKoAAGCsAABPrwAAQLEAAD
G0AAAltgAAGrKAAA+7AAAFvQAAAL8AAADBAAAAwwAAAMQAAADGAAAAyAAAAMsAADNA
AAA0AAAAANMCAADXCwAA2xUAAOAhAADILgAA50EAAOhXAADpbGAA6YYAAOmCAADprgAA
6bwAAOnCAIOqAAB0rQAAY7AAAFKyAABBtgAAMrKAACS9AAAZwAAADsMAAATGAAAAyQAA
AMsAADMAAAAzwAAANEAAADTAAAA1gAAANoAAADdAAAA4AAAAOMAAADnAAAA6wUAA
PARAAD1HgAA+S0AAPpCAAD7WAAA/G8AAP2FAAD+lwAA/qQAAP6pAP8AEQH/AA4A/wANA
P8AEAD/ABwA/wAoAP8ANQD/AEEA/wBOAP8AWQD/AGIA/wBqAPwAcgD6AHgA+QB+APgAhA
D3AlOa9gCPAPUAIQD0AJsa8wCiAPIAqgDwALMA7gC/AOsA0ADpAOYA6AD3AOYA/wDIAP8A4
AD/ANwA/wDaAP8A1gD/AP8ADAH/AAgA/wAGAP8ACwD/ABgA/wAkAP8AMQD/AD0A/wBJAP4A
VAD6AF4A9gBmAPMAbQDxAHQa8AB6AO8AgADuAIUA7ACLAOsAkQDqAJcA6QCeAOYApQD
kAK4A4QC5AN8AyQDdAN8A2gDxANgA/wDSAP8AzQD/AMwA/wDMAP8AywD/AP8ABgD/AAIA/
wAAAP8ACAD/ABQA/wAgAP0ALAD6ADgA+ABEAPMATwDuAFgA6gBhAOgAaADmAG8A5AB1
AOMAewDhAIAA4ACGAN4AjADdAJIA2gCZANcAoADUAKkA0gCzANAAwQDNANUAygDqAMUA
+gDBAP8AvwD/AL8A/wC+AP8AvgD/AP8AAQD/AAAA/wAAAP8ABAD8AA8A9gAbAPIAJwDvAD
MA6wA+AOYASQDhAFMA3QBcANsAYwDYAGoA1QBwANQAdgDSAHsA0ACBAM4AhwDMAI0
AyQCTAMcAmwDFAKQAwCuAMAAuWC+AMwAuQDiALYA9AC0AP8AswD/ALIA/wCxAP8AsA
D/AP8AAAD/AAAA/wAAAP0AAADxAAkA6wAWAOYAIGDhAC0A3AA4ANYAQwDSAE0AzgBWA
MsAXgDJAGQAxgBrAMQAcADCAHYAwAB7AL4AgQC9AlcAuWCOALkAlgC3AJ8AtQCoALEAtA
CuAMQArADZAKoA7gCoAPwApgD/AKQA/wCkAP8AowD/AP8AAAD/AAAA/wAAAPAAAADnAAM
A3wAQANkAHADRACcAzAAyAMgAPQDEAEcAwABQAL0AWAC6AF8AuABIALYAawC1AHAAsw
B2ALEAfACwAlIArgCJAKwAkQCqAJApwCkAKUARwCjAL4AoADSAJ4A6ACcAPkAmgD/AJkA/w
CYAP8AmAD/AP8AAAD8AAAA7QAAAOMAAADcAAAA0wAIAAMsAFQDEACAAvwAsALsANgC3A
EEAtABKALEAUgCuAFkArABfAKoAZQCpAGsApwBxAKYAdwCkAH0AogCEAKAAjACeAJYAnA
CgAJJoAqwCYALoAIQDMAJMA5QCRAPUAjwD/AI4A/wCNAP8AjAD/AP8AAADvAAAA4wYAANgJ
AADOBgAAxwAAAMAADQC6ABkAtAAkALAAALwCsADoAqQBDAKYATACjAFMAoQBAAJ8BYAC
eAmYAnAJsAJsDcgCZA3kAlwSAAJYFiQCUBZIAkgadAJAHqQCNB7cAiwlAlkAJ4wCGCvYAhAv/
AIIl/wCBC/8AgQv/APYAAADnDQAA2RYAAMwaAADCGAAAuxMAALYJAgCwARAAqWldAKYgK
ACiCDMAngo9AJsMRgCZDU0Alw1UAJUOWwCTDmEAkhBnAJAQbQCPEHQAJRF8AlSshQCJE

o8AhxOaAIUTpwCDFLUAgRTKAH4V4wB7FvcAehb/AHgW/wB4Ff8AdxX/AO8LAADfGgAAzyMAA
MMmAAC6JQAAsyEAAKwaAACnEQYAoQ8WAJwTlgCYFS0AIBY3AJIXQACPGEGAjRIPAlSvVg
CJGIlwAhxpjAIYbaQCEG3AAgxx4AIEcgQB/HYsAfR2XAHsepAB5HrIAdx7GAHQe3wByHvUAcB7/
AW8e/wFuHf8BbR3/AekWAADYJAAAXy0AALwwAACzMAAArC0AAKUnAACeHwAAmBoQAJMc
HQCOHigAiyAyAlghPACGIUQAgyJLAIEiUgB/I1gAfiNfAHwkZQB6JGwAeSR0AHclfQB1JYgAdCW
TAHlmoAFwJq8BbSbCAWsm3AFoJvICZyX/AmUI/wJIJP8DZCT/A+MfAADQLAAAwjQAALc4AAC
uOQAApjCAAJ8xAACXKwAAkCMLAllokGQCGJiQAagiguAH8pNwB9KT8AeipHAHggTgF2KIQBdSp
bAXMqYQFxFK2gBcCtwAm4reQJsLIQCaiyQAmgsnQJmLKwDZCy+A2Is2ANgLPaEXiv/BF0r/wRd
Kv8EXCr/BN4mAADKNAAAvtTsAALM/AACqQAAAoj4AAJk6AACRNAAAiSwGAIMsFAB+LSAAei8
qAHcvMwB1MDsBcjBDAXAwSgJuMFACbDBXAmowXQNoMGUDZzBtA2UxdgRjMYEEYjGNBGA
ymgVeMqkFXDK7BVkx1AVYMe4GVjH+BIUw/wZVL/8GVS//BtkS AADGOQAAukEAAK9FAACnRg
AAAnUAAJVAACMOwAAgzQBAHwyEAB3MxwAczQnAHA1LwBtNTgBajU/Amg1RgNmNUwDZD
VTBGI1WgRgNWEFXzVpBV02cwZbNn0GWjaKB1g2mAdWN6cHVDe4B1I20QhQNusITzX9B041/
wdONP8HTJT/B9MyAADDPgAAtkYAAKxKAACKTAAAm0sAAJJGAACIQQAafjsAAHU4DABwORk
AbTkjAGk6LAFmOjQBYzo7AmE6QgNeOkkEXDIPBV5sVgZZOV4GVzpmB1Y6cAhUOnolUjqHCV
E7IQUIPO6UJTtu2CUs6zglJOukJSTn8CUg5/wlIOP8ISDj/CM84AAC/QwAAtEoAAKpPAACHUQAA
mFAAAI9MAACERgAAekEAAHA9CQBqPRYAZj4gAGM/KQBGPzEBXT44AlO+PwRYPkUFVj5MBI
Q+UwdSPisIUT5jCE8+bQINPngKTD6ECko/kwplP6MKRz+0CkU/zApDPugKQz37CkI9/wpCPP8J
Qjz/Ccs8AAC8RwAAsU4AAKhTAACfVQAAlIUAAlxRAACBTAAAdkcAAGtCBQBkQhMAYUMdAF1
DJgBaQy4BV0M1AIRCPANSQKIFUEJJBk5CUAdMQlglS0JhCUICagpHQnULRkKCC0RDkAtDQ6
ELQUOyCz9Dygs+QuYLPUH6Cj1B/wo9QP8KPUD/CsdAAAC6SwAAr1IAAKVXAACdWgAAIFkA
AlpWAAB9UQAACKwAAGZHAkBfRg8AW0caAFhllwBVSCsBUkcyAk9HOQNNR0AFS0dHBkIHTg
dHR1YJRUZeCkNHaaAtCR3MLQEeACz9Hjws9R58LPEixCzpHyAs5R+ULOeb5CzhF/wo4RP8KO
ET/CsRFAAC3TwAArFcAAKNcAACaXgAAkV4AAIdbAAB6VgAABlIAAGJNAABZTAsAVUwXAFJ
MIQBPTSkATU0wAUpNNwNITD0ERkxEBkRMSwdCTFMIQExcCT5MZgo9THELO0x+CzpMjQs4
TZ0LN02vCjVMxgo0TOMKM0v4CjNK/woySf8KMkj/CsFJAAC0VAAAqlsAAKFgAACYYwAAj2MAA
IVhAAB3XAAAa1gAAF9VAABUUCAT1EUAEExSHQBKUiYASFItAUVSNAJDUsDQVJCbt9SSQ
Y9UIEHO1FaCDIRZak3Um8KNIJ8CjVSiwozUpwJMIKuCTBSxQgvUeIlID3CS1P/wktTv8JLU3/Cr
1OAAcXWQAAqGAAAJ9IAACWaaaaJWkAAINnAAB1YwAAaF8AAFXcAABPWAMASFcQAEZYg
gBEWCIAQIkqAEBZMQE+WTgCPFg/AzpYRwU4WE8GNlhYBzRYYgcyWG0IMVh6CC9ZigguWZ
oHLFmtBytZwwYqWOEGKFb2ByhV/wgoVP8IJ1P/CLpUAACuXgAApWYAAJ1rAACUbgAAi28AAI
BuAABzagAAZmcAAfhKAABMYAAQI4LAD9eFgA9Xx8AO18mADlgLgA4YDUBNmA8AjRgRAM
yYEwEMGBVBS5gYAUtYGsFK2B4BSpgiAUoYJkEJ2CrBCVgwgQkX98DI131BSJb/wYiWv8Gllr/B
7ZaAACrZAAAomwAAJpxAACSDAAAiXUAAH50AABxcgAAYm4AAFRrAABlaQAAPGYFADdmEA
A1ZhoAM2ciADFnKQAWaDEALmg4AS1oQAEraEkCKmhSAihoXQMmaGgDJWh2AyRohgliaJcCl
WipAh9owAEeZ94BHWX0Ahxj/wMcYv8EG2H/BLJhAACoawAAoHIAAJh4AACQewAAhnwAAht7
AABseAAAXXYAAFBzAABDcQAAOHAAAC5vCgArbxQAKW8cAChwJAAncCsAJXAzACRwOwAj
cUQAlnFOACBxWQAfcWUAHnFzABxxgwAbcZQAGXGnABdwwQAVb9sAFW3zABVs/wEUa/8CF
Gr/Aq5pAAClcgAAAnXoAAJZ/AACNggAAhIMAAHaBAABnfwAAV30AAEp7AAA+egAAM3kAACh4
AgAgeAwAHngVAB15HQAceSUAG3ktABp5NQAzeJ4AGHplABZ6UwAVemAAFHpuABJ6fgAQep
EADnqkAAx5uQAKenQACnbvAAAt1/gALdP8AC3P/AKpxAACiegAAmoEAAJOGAACLiQAAfokAA
G+IAABghgAAUYUAAEOEAAA4gwAALIIAACKCAAZggIAEoIMABGDFQAPgx0ADoMIAA2DLQ

AMhDYAC4RAAAmETAAIhFkAB4RoAAWEeAADg4oAAYOdAACCSQAAGcsAAH/nAAB/9wAAfv8
AAH3/AKZ6AACfgwAAmIkAAJGNAACEjgAAdo8AAGeOAAByjQAASowAADyMAAAxjAAAJowAA
ByMAAAUjAAAC4wFAASNDQAAjRYAAI0eAACOJQAAji4AAI44AACORAAAjIEAAI5gAACOCAAAj
oMAAI2WAACMqwwAAi8IAAIrfAACJ8gAAiPwAAIj/AKKEAACcjAAAIZEAAIqTAAB8IAAAbZQAAF6
UAABPIAAAQZQAADSUAAAoIAAAHpUAABWVAAAMlgAABJYAAACXCAAAIw8AAJgXAACYHg
AAmCYAAJkvAACZOgAAmUcAAJIWAACZZwAAmXkAAJiOAACYowAAI7kAAJbVAACV7AAAIpC
AAJT9AJ+NAACZIAAAjpcAAIGZAABymgAAYpoAAFObAABEmwAAN5wAACucAAAgnQAAFp4A
AAyfAAAEenwAAAKAAAACHAAAAogcAAKINAACjFQAAoxwAAKQkAACILgAApjoAAKZJAACIWw
AApW4AAKWDAACkmQAApK8AAKPIAACi4wAAovEAAKH4AJyWAACSmgAAhZ0AAHafAABmn
wAAVqAAAEehAAA5owAALKMAACGIAAAWpgAADKcAAAOoAAAAqQAAAKoAAACrAAAArAAA
AKwDAACtCgAArhEAAK8ZAACxlgAAsi0AALM6AAC0SwAAtF4AALRzAAC0igAAs6MAALK6AAC
x1AAasecAALDwAJWdAACJoAAAEqMAAGqkAABZpQAASacAADuoAAAsqgAAIasAABatAAAMr
wAAArAAAACxAAAAsgAAALQAAAC1AAAAtgAAALgAAAC5AAAAuwQAAL0MAAC/FQAAwR8AA
MQRADGOWAAx04AAMdjAADlegAAyJIAAMmoAADlvQAAX9EAAMfGAlYjAAB9pgAAbagAAF2pA
ABMrAAPK4AAC6wAAAhsgAAFrQAAAU3AAABuQAAALoAAAC7AAAAvQAAAL8AAADAAAAA
wgAAAMQAAADHAAAAyQAAAMwAAADQBGA1BAAANobAADfKQAA4DsAAOFRAADiaAAA43
8AAOOWAADkqgAA5LkAAOTDAICoAABxqwAAYK0AAE+wAAA+sgAAL7YAACG5AAAWvAAAC
r8AAADBAAAAxAAAAMYAADHAAAAyQAAAMsAAADNAAAA0AAAAANIAADWAAAA2gAAAN4
AAADiAAAA5gAAAOsLAADwGAAA9ScAAPc7AAD4UgAA+GoAAPmBAAD5lgAA+aMAAPqsAP8
ADAD/AAkA/wAIAP8ADAD/ABkA/wAIAP8AMgD/AD4A/wBKAP8AVAD/AF4A/ABmAPsAbQD5AH
QA+AB6APcAgAD2AIUA9QCLAPQAKQDzAJcA8QCeAO8ApQDtAK4A6wC6AOkAygDoAOAA5g
D0AOIA/wDcAP8A2QD/ANgA/wDTAP8AzWd/AP8ABwD/AAIA/wAAAP8ACQD/ABUA/wAhAP8AL
QD/ADkA/wBFAPkAUAD1AFkA8wBhAPEAaQDvAHAA7gB2AO0AewDrAIEA6gCGAOKAjADnAJI
A5ACZAIOIAoQDgAKkA3gC0ANwAwwDaANcA1QDtAM4A/QDLAP8AygD/AMkA/wDIAP8AxQD/A
P8AAAD/AAAA/wAAAP8ABQD/ABEA/gAdAPoAKQD3ADQA8wA/AO0ASgDqAFQA5wBdAOUAZ
ADjAGsA4QBxAN8AdgDeAHwA2wCBANKahwDWAI0A1ACUANMAmwDQAKQAzgCuAMwAuWd
HAM0AwgDIAL8A9wC+AP8AvAD/ALsA/wC6AP8AugD/AP8AAAD/AAAA/wAAAP8AAAD4AAwA8
wAYAO4AJADpAC8A5AA6AOEARQDdAE8A2QBXANYAXwDTAGUA0QBrAM4AcQDMAHYAyg
B8AMgAgQDHAIgAxQCOAMMAIgDBAJ8AvgCpALoAtAC2AMQAtADcALIA8ACwAP8ArgD/AK0A/
wCtAP8ArAD/AP8AAAD/AAAA/wAAAPQAAADtAAcA5gATAOAAHwDZACkA1AA0ANAAPwDNA
EkAyQBSAMYAWQDDAGAAwABmAL4AawC9AHEAuWB2ALoAfAC4AIItgCJALQAKQCxAJkArg
CjAKwArgCqAL0AqADSAKYA6QCjAPoAogD/AKEA/wCgAP8AnwD/AP8AAAD/AAAA8wAAAOoA
AADhAAIA2QANAM8AGQDJACQAxQAuAMEAOQC+AEMAUGBLALcAUwC1AFoAswBgALEAZg
CvAGsArgBxAKwAdgCqAH0AqACDAKYAiWckAJQAogCeAKAAqQCeALcAnADKAJkA4gCXAPQ
AlgD/AJMA/wCSAP8AkGd/AP8AAADzAAAA6AAAAAN0AAADUAAAAygAHAMMAEgC9AB4AuAAo
ALQAMgCxAdwArgBFAKsATQCpAFQApwBaAKUAYACjAGYAoQBrAKAAcQCeAHcAnAB+AJsA
hgCZAI8AIwCaAJQApQCSALIAkADEAI4A2wCLAPAAiQD+AlcA/wCGAP8AhgD/APgAAADqAAA
A3QMAANAGAADGAwAAvwAAALkACwCzABcArQAiAKkALACmADYAowA/AKAARwCdAE4Am
wBVAJoAWwCYAGEAlgBmAJUAbACTAHMAkQB6AJAAggCOAlsAiWcWAlkAogCHAK8AhQDAA
IIB2ACAAu0AfgP7AHwD/wB7A/8AewP/APAAAADhCwAA0RQAAMUWAAC8FAAAAtQ4AAK8FAQ
CpAA8ApAAaAJ8AJQCbAC8AmAI4AJUDQQCTBEkAkQVPAl8GVQCNBIsAjAdhAlOHaACICG4A
hwh2AIUJfgCDCogAgQqTAH8LoAB8C60Aegy/AHgM1wB1De8Acw7+AHIO/wBxDf8AcA3/AOKLA

ADZGAAAYCAAAL0jAAC0lgAArR0AAKYWAACgDAUAmgcTAJUKHwCRDCkAjg0yAIsPOwCJEE
MAhxFKAIQSUACDEIYAgRNdAH8TYwB+FGoAfBRyAHoUegB4FYUAdhWQAHQWnQByFqsAcB
a9AG4W1QBrF+4AaRf/AGgX/wBnF/8AZxb/AOMWAADPlwAAwSoAALctAACuLQAApikAAJ8jAA
CYGwAAkRMMAlwUGQCHFQAhBgtAIEZNgB/Gj4AfBtFAHobTAB5G1IAdxxYAHUcXwB0HGYAc
h1uAHAddwBvHoEAbR6NAGsfmgBpH6gAZx+5AGQf0QBiH+sAYB/9AV8f/wFeHv8BXh7/AdwfAA
DJKwAAvDIAALI1AACpNgAAoTMAAJktAACRJgAAiR4GAImdFAB/Hx8AeyApAHghMgB2ljoAdCJ
BAHEjSABwI04AbiNVAGwkWwBrJGIAaSRqAGclwBmJX4AZCaKAWImlwFgJqYBXia2AVwmzgF
aJukCWCX7AIcl/wJWJP8CViT/AtUmAADEMgAAuDkAAK48AACIPQAAnTsAAJQ2AACLLwAAgyg
BAHskEAB3JhsAcyclAHAoLgBtKDYAayk9AGkpRABnKUoBZSpRAWMqWAFiKl8BYCpnAV8rcAJ
dK3oCWyyGAloslAJYLKMCViy0A1QsywNSLOYDUCv6A08q/wRPKv8ETyn/BNAsAADAOAAAAtT4
AAKtCAACiQwAAmUEAAJA8AACHNwAAfTEAAHQrCwBvLBcAay0hAGguKgBmLjIAYy85AGEvQ
AFfL0cBXS9NAIwvVAJaL1sCWDBkA1cwbQNVMHcDVDGDBFIxkQRQMaEETjGxBEwxyARKMe
QFSTD4BUgv/wVIL/8FSC7/BcwyAAC9PAAAskMAAKhHAACfSAAAlkcAAI1CAACDPQAAeDgAA
G4yBwBpMRQAZTleAGEzJwBfNC8AXDQ2AVo0PQFYNEMCVjRKAIQ0UQNTNFgDUTRhBFA1a
gRONXQFTTWBBUs2jwVJNp4GRzavBkU2xQZENeIGQzX3Bkl0/wZCM/8GQjP/Bsg3AAC6QQAA
r0gAAKVMAACcTQAAIEwAAIpIAAB/QwAAAdD4AAGk4AwBiNhEAXjcbAFs4JABYOCwAVjgzAVM
4OQFROEACTzhGA044TQRMOVUESzleBUk5ZwZIOXIGRjp+BkU6jAdDOpwHQTqtBz86wwc+O
uAHPTn2Bz04/wc8N/8HPDf/B8U7AAC3RQAARewAAKNQAACaUgAAkVEAAIdNAAB8SAAAcUM
AAGU+AABdOw0AWDwYAFU8IQBTPSkAUD0wAU09NgFLPDwCST1DA0g9SwRGPVIFRT1bBk
M9ZQZCPm8HQD58Bz8+igc9PpoHPD+sBzo+wQc4Pt4HOD31Bzc8/wc3O/8HNzv/B8I/AAC1SQA
AqlAAAKFUAACYVgAAj1YAAIVTAAB5TQAAbUKAAGFDAABYQQKAU0AVAFBBHgBNQSYAS0It
AEhBMwFGQTtoCREFBA0NBSARBQIAFQEJZBj5CYgc8Qm0HO0J6BzpDiAc4Q5kHN0OqBzVDv
wczQ90GMkH0BzJA/wcyQP8HMj//B79EAACyTQAAqFQAAJ9ZAACWWwAAjVoAAINYAAB2UgA
Aak4AAF1KAABSRgYATUURAEpGGwBIRiMARUcqAENHMQFBRzgCQEc/Az5HRgQ8R04FOkd
WBjIHYAY3R2sHNkd4BzRIhwczSJcHMUiPbJBlvgYuSNsGLUzbBi1F/wcsRP8HLEP/B7tIAACwUg
AAplkAAJ1eAACUYAAAI2AAAIBdAABzWAAAZ1UAAfTRAABPTAIAR0sNAERLGABCTCAAQEW
nAD5MLgA8TTUBOk08AjINQwM3TUsENU1UBTNNXgUyTWkGME12Bi9NhQYUtpYFLE6oBStO
vAQpTdkEKEzyBSdK/wYnSf8GJ0j/BrhNAACtVwAApF4AAJtjAACSZQAAiWUAAH5kAABxXwAAZ
VwAAFIYAABMVAAAQVEJAD5RFAA8Uh0AOIlkADhTKwA2UzIBNVM5ATNTQQIxU0kDMFNsAy
5TXAQsU2cEK1N0BCpUgwQoVJQEJ1SmAyVUuwMjU9gC11LxAyJQ/wQiT/8Flk7/BbVTAACqXA
AAoWMAAJloAACQawAAh2sAAHxqAABvZwAAy2QAAfVgAABIXAAAPFkFADZYEAA0WRkAMl
kgADFZKAaVWi8ALlo2ACxaPQErWkYBKVPaihaWQImW2UCJVtyAiRbgQliW5ICIVulAR9buQE
dWtUBHVjwAhxX/wMcVf8DHFX/BLFZAACnYgAAAn2oAAJduAACOcQAAhXIAAHpxAABtbGAAxm
sAAFFnAABEZQAAOWIAAC9gCgAsYBQAKmEcACHhlwAnYSoAJmEyACViOQAKYkIAImJLACFi
VgAgYmIAHmNvAB1jfwAbY5AAGmKjABhitwAWYdMAFmDuABZe/gEVXf8CFVz/Aq5gAACkaQA
AnHAAAJV1AACMeAAAg3kAAHh4AABpdQAAWXIAAEVvAABAbQAANGsAAClpAwAiaQ0AIGkV
AB9pHgAeaiUAHWosABxqNAAaaj0AGWpGABhrUQAXa10AFWtrABRrewASa40AEGugAA5qtAA
Mac4ADGjrAA1m/AANZf8ADWT/AapoAACChcQAAMngAAJJ8AAckfwAAgYAAAHN+AABjfAAAVH
kAAEZ3AAA6dQAAL3QAACRzAAAbcwUAFXIOABRzFwATcx4AEnMmABFzLgAQdDYADnRAAA
10SwALdFgACnRmAAl0dgAHdlcABXOaAANyrgACcsYAAxDjAAfV9QACbv8AAm3/AKZwAACee
QAAI38AAJCEAAClHgAAeoYAAGyEAABcggAAToEAAEB/AAA0fgAAKX0AAB99AAAWfAAADXw
HAAAd9EAAGfRgABX0gAAN9JwACfjAAAX46AAB+RQAAfIEAAH5gAAB9bwAAfYEAAH2UAAB8q

QAAe78AAHrcAAB48QAAePwAAHf/AKN5AACcgQAAIYcAAI6LAACBjAAAc4sAAGSKAABUiQAA
RogAADmHAAAthwAAIocAABiHAAAQhgAAB4cDAACHCwAAhxIAAlgaAACIIQAAiCkAAIgyAACI
PQAAiEkAAIhYAAcIaAAAIHoAAleNAACHogAAhrgAAITUAACD7AAAq/gAAIH/AJ+DAACZigAAk4
8AAleRAAB5kQAAapEAAfQRAABLkAAAPZAAADGQAAAIkAAAG5AAABGQAAAIkAAAAJEA
CRBQAAkgsAAJITAACSGgAAkyEAAJMpAACTMwAAk0AAAJNOAACTXgAAk3AAAJOFACSm
gAAkbAAAJDKAACQ5QAAj/QAAI78AJyMAACXkgAAjJUAAH6WAABVlwAAX5cAAE+XAABBlwA
AM5gAACeYAAAcMQAAPkAAaIAAAAMgAAAJsAAACbAAAAAnQAAJ0KAACdEQAAnhgAAJ4
gAACfKQAAoDQAAKBDAACgUwAAoGUAJ95AACfkAAAnqYAAJ6+AACd2wAAAnO0AAJz4AJqV
AACQmAAAgpsAAHOcAABJnAAAU50AAESeAAA1nwAAKaAAAB2hAAATogAACKMAAACjAAAA
pAAAAKUAAACmAAAApgAAAKcAAACoBgAAqQ0AAKoUAACrHQAArScAAK4zAACuRAAArlcA
AK1sAACtgwAArZsAAK2xAACsygAArOEAAKvvAJObAACGngAAAd6AAAGehAABWogAARqQAA
DelAAAppwAAHqgAABKpAAAlqgAAAKwAAACsAAAArQAAAK4AAACvAAAAsQAAALIAAACzAA
AAtQAAALYIAAC4EAAAuxoAAL4IAADANAAAwEYAAMFbAADCCQAAwYsAAMCkAAC/uQAAv80
AAL/gAlmhAAB7owAAa6UAAfqnAABlqQAAOaoAACqtAAAergAAErAAAAeyAAAAtAAAAALUAAA
C2AAAAuAAAAALkAAAC7AAAAvQAAAL4AAADBAAAAwwAAAMYAAADJAgAAzAsAANEXAADW
JAAA2TUAANpJAADB YAAA3HgAAN2QAADepAAA3rUAAN3FAH6mAABuqQAAXaoAAEytAAA8r
wAALLIAAB21AAASuAAABroAAAC8AAAAvwAAAMEAAADBAAAAwwAAAMUAAADHAAAAygAA
AMwAAADPAAAA0wAAANcAAADcAAAA4QAAAOYGAADrEwAA8SIAAPI2AADyTQAA9GUAAP
V8AAD1kQAA9qAAAPasAP8ABwD/AAMA/wADAP8ACgD/ABYA/wAjAP8ALgD/ADoA/wBFAP8A
UAD9AFkA+wBiAPkAaQD4AHAA9wB2APYAewD1AIEA8wCGAPEAjADvAJIA7gCZAOWAoQDrA
KoA6QC1AOcAxADIAnoA4ADvANkA/wDXAP8A1QD/ANQA/wDOAP8AygD/AP8AAQD/AAAA/wA
AAP8ABgD/ABMA/wAfAP8AKgD+ADUA+QBAAPYASwDzAFUA8QBdAO8AZQDtAGsA6wBxAOo
AdwDoAHwA5gCBAOQAhwDiAlO4QCUAN8AnADdAKUA2wCvANkAvQDSAM8AzADnAMoA+g
DIAP8AxwD/AMYA/wDDAP8AwAD/AP8AAAD/AAAA/wAAAP8AAgD/AA4A+wAbAPYAJgDxADAA
7gA7AOoARgDmAFAA5ABYAOEAYADfAGYA3QBsANoAcgDYAHcA1gB8ANQAggDSAlgA0ACP
AM4AlwDMAJ8AyQCpAMQAtQDAAMYAvgDfALsA9AC6AP8AuAD/ALcA/wC2AP8AtQD/AP8AAA
D/AAAA/wAAAPsAAAD1AAkA7wAWAOcAlQDiACsA3wA2ANsAQADXAEOA0wBTANAAWgDNA
GEaywBnAMkAbADHAHIAxQB3AMQAFADCAIMAwACJAL4AkQC7AJJoAtwCjALQArwCyAL4AsA
DUAK4A7QCsAP4AqwD/AKkA/wCoAP8AqAD/AP8AAAD/AAAA+QAAAPAAAAADoAAUA4AAQAN
cAGwDSACYAZQAxAMoAOgDGAEEQAwgBNAMAAVAC9AFsAuwbhALoAZwC4AGwAtgBxALUA
dwCzAH0AsACEAK4AiWcsAJQAqgCeAKgAqQCmALcAowDLAKEA5QCfAPcAnQD/AJwA/wCbA
P8AmwD/AP8AAAD4AAAA7gAAAOQAAADaAAAAzwAKAMgAFgDDACEAvwArALsANQC3AD4A
tABHALIATgCwAFUArgBbAKwAYQCqAGYAqABsAKYAcQCIAHcAowB+AKEAhgCfAl8AnQCZAJ
sApACZALEAlgDDAJQA2wCSAPEAkAD/Al8A/wCNAP8AjAD/APsAAADuAAAA4gAAANYAAADM
AAAAwwAFALwAEAC3ABsAsgAlAK4ALwCrADgAqABBAKUASACjAE8AoQBWAJ8AWwCeAGE
AnABmAJJoAbACZAHIAlwB5AJUAgQCTAlOAKQCUI8AnwCNAKwAigC8AlgA0gCFAOsAhAD6AI
EA/wCAAP8AgAD/APMAAADkAAAA1QEAMgDAAC/AAAAuQAAALIAACgCsABQApwAfAKMAKA
CfADIAnAA6AJJoAQgCYAEoAlgBQAJQAVgCSAFsAkQBhAl8AZwCNAG0Aiwb0AlOAFACIAIUhQ
CPAIMAmwCAAKgAfgC3AHwAzQB6AOYAdwD4AHYA/wB1AP8AdAD/AOsAAADbCgAAyhIAAL8
TAAC2EQAArwoAAKkAAQCjAA0AnQAYAJkAlgCVACsAkgA0Al8APACNAEQAiwbKAlkAUACHA
FYAhQBcAlQAYgCCAGgAgAFwAH4BeAB8AoEAegKMAHgDmAB2A6YAcwS1AHEEygBvBeQAb
Qb1AGsG/wBqBv8AagX/AOMLAADQGAAAwH4AALcgAACvHgAApxoAAKASAAACZCAQAIAERA

8BGwCLAyUAhwUuAlQHNgCCCD4AgAlFAH4KSwB8CIEAegtXAHkLXQB3C2QAdQxsAHQMdA
ByDX4AcA2JAG4OlgBrDqUAaQ61AGcOywBkEOYAYxD5AGEQ/wBhEP8AYBD/ANsWAADJIQA
AvCcAALlqAACpKQAAoSUA AJkgAACSFwAAig4IAIUMFQCADh8AfRAoAHoRMB3EjgAdRM/A
HQRgByFEwAcBRTAG4VWQBtFWAAaxVoAGkWcABoF3oAZheGAGQYkwBiGKIAYBiyAF0Yy
ABbGOMAWhn4AFkY/wBYGP8AWBf/ANMfAADDKQAAty8AAK0zAACkMgAAAnC8AAJMpAACLIw
AAgxsBAHwWEAB3FxsAdBkkAHEaLABuGjQAbBs7AGocQgBoHEgAZxxPAGUdVQBjHVwAYh1k
AGAebQBfHncAXR+DAFsfbABZIJ8AVyCvAFUgxQBTIOEAUId2AVEf/wFQHv8BUB7/Ac4mAAC/
MAAAszYAAKk5AACgOgAAmDcAAI8yAACGLAAAFSUA AHQeCwBvHhYAayAgAGghKABmITAA
ZCI3AGliPgBgl0UAXiNLAf0jUgBbJfKAWSRhAFgkagBWJXQAVSWAAFMmjgFRJp0BTyatAU0m
wgFLJt4BSib1Akkl/wJJJP8CSCT/AsksAAC7NgAAsDwAAKY/AACdQAAAI0AAIs5AACBMwAAe
C0AAG4mBgBnJRM AZCYcAGEnJQBeJywAXCg0AFooOgBYKUEAVilHAFUpTgBTKVUBUipeAV
AqZwFPKNEBTSt9AkwwiJKLJoCSCyrAkYsvwJEK9sCQyvzA0lq/wNCKf8DQin/A8YxAAC4OwAAr
UEAAKREAAcARQA AKkMAAlg/AAB9OgAAczQAAGgtAgBhKg4AXSsZAFoslgBXLskAVS0wAFM
tNwBRLj0ATy5EAU4uSwFMLIBSy9bAkkvZAJIL24CRjB7A0UwiQNDMZgDQjGpA0AxvQM+MNk
DPTDyAzww/wQ8Lv8EPC3/BMI2AAC1PwAAqkUAAKFJAACYsGAAj0gAAIVEAAB6QAAAbzoAA
GQ0AABbMAsAVjAWAFMxHwBRMiYATjltAEwyMwBKMjoBSTJBAUczSAFGM08CRTNYAkM0Y
QNCNGwDQDV4BD81hgQ9NZYEPDWnBD01uwQ4NdYENzTwBDcz/wQ2Mv8ENjL/BL86AACzQ
wAAqEoAAJ9OAAcWTwAAjU0AAINKAAB3RQAAbEAAAGA7AABWNgcAUDUTAE02GwBLNiMA
SDcqAEY3MABENzcBQzc+AUI3RQJAOE0CPzhVAz04XwM8OWoEOzl2BDk5hAQ4OpQENjqmB
DQ6uQQyOdUEMTnvBDE4/wQxN/8FMTb/Bbw+AACwSAAApk4AAJ1SAACUuWAAi1IAAIBPAAB
0SgAAaUUA AFxAAABROwQASzoPAEg7GABFOyAAQzsnAEE8LgA/PDQBPjw7ATw8Qgl7PUoC
Oj1TAzg9XQM3PWcENT50BDQ+ggQyPpIEMT+kBC8/uAQtpTMDLD3uBCw8/gQsO/8FKzr/BbIDA
ACuTAAApFIAAJtWAACSWAAAIvCAAH5UAABYTwAAZksAAFpHAABOQgAARkALAEJAFQBAQ
B0APkAIADxBKwA6QTIAOUE5ATdBQAE2QkgCNEJRAzNCWgMxQmUDMENyBC5DgAQtQ5ED
K0OjAypDtgMoQ9EDJ0LtAydB/gQmQP8EJj//BLdHAACrUAAAolcAAJlbAACQXQAAh1wAAHxaA
ABvVQAAY1EAAFhOAABMSQA AQEUIADxFEgA6RhoAOEYiADZGKQA1Ry8AM0c2ADJHPQEw
R0UBL0dOAi1IWAIsSGMDKkhwAyllfwMoSY8CJkmhAiVJtQljSM8BlkfsAiFG/QMhRf8EIUT/BLNM
AACpVQAAoFwAAJdgAACPYgAAhWIAAHpgAABtXAAAYVkaAFZVAABJUQAAPU0DADZLDgAz
TBcAMUweADB MJQAuTSwALU0zACxNOgAqTkMBKU5LASdOVgEmTmEBJU5uASNPfAEiT40B
IE+gAR9PswEdTs0AHE3qARxL/AlcSv8CHEn/A7BSAACmWwAAAnmEAAJVM AACNaAAAg2gAA
HlnAABsYwAAX2AAAFJcAABFWQAAOIUAAC9SCQArUhMAKIMaAChTlgAnVCgAJlQvACRUNw
AjVD8AIIVIA CFVUwAfV4AHIVrAB1VegAbVYsAGIWeABhVsQAWVMsAFIPpABZS+wEVUf8BFV
D/Aq1YAAcK YQAAM2cAAJNsAACLbgAAgm8AAHduAABqawAAW2cAAE5kAABBYAAANV0AAC
pbBAAjWg0AIVoWACBbHQAfWyQAHlsrABxcMgAbXDsAGlxEABlcTwAXXFsAFIxoABVddwATXY
gAElybABBcrwANW8gADVrmAA1Z+gAOV/8ADIf/AapfAAChaAAAmW4AAJFyAACJdQAAGHYAA
HV1AABmcgAAVm4AAElrAAA9aQAAMWYAACZIAAAAdYwcAGGMQABZjGAAVZB8AFGQmABN
kLQASZDYAEWQ/AA9kSgANZVYADGVjAAtlcwAJZYQAB2SXAAV kqgAEY8EAA2LeAANh8wAE
X/8ABV7/AKZnAACebwAAI3UAAI96AACHfAAAFX0AAG97AABgeAAAUHUA AENzAAA3cQAAK3
AAACFuAAAYbQAAD20IAAptEQAJbRkACG0gAAduKAAGbjAABW46AARuRAACbIAAAW5eAAB
ubQAAbX4AAG2RAABspQAAa7sAAGvWAABp7wAAaPoAAGj/AKNvAACbdwAAIX0AAI2BAACFh
AAAd4MAAGiBAABZfwAASn0AADx7AAAwegAAJXkAABt4AAASdwAACXcFAAJ3DAAAdxQAAH
cbAAB4lwAAeCoAAHgzaAB4PgAAeEoAAHhYAAB3ZwAAAd3gAAHeMAAB2oAAAdbUAAHTQAA

Bz6gAAcvgAAHH/AKB4AACZgAAAk4UAAIuJAAB+iQAAclgAAGCHAABRhQAAQ4QAADWDAAA
pggAAH4IAABWCAAALgQAAA4EBAACBCAAAagg4AAIIWAACCHQAAgiQAAIMsAACDNgAAg0IA
AIJQAACCYAAAgnEAAIGFAACBmgAAgK8AAH/JAAB+5QAAffUAAHz9AJ2CAACXiAAAkY0AAIS
OAB2jwAAZ44AAFaNAABljAAAOWAAC2LAAhiwAAF4sAAA2MAAAEiwAAAlwAAACMAgAAj
AgAAI0OAAACNFQAAjRwAAI4kAACOLQAAjjkAAI5HAACOVwAAjmgAAI18AACNkgAAjKcAAIvAA
ACK3gAAIfEAAIj6AJqLAACVkJAAiZMAAHuUAABsIAAAXJQAAEYUAAA9IAAAMJQAACOUAAA
YIAAADpUAAASVAAAAIQAAAJYAAACWAAAAIwEAAJcHAACYDQAAmBQAAJkbAACalwAAmi4
AAJs7AACaSwAAml0AAJpxAACahwAAmZ8AAJm1AACY0AAAI+gAAJf0AJiUAACNIgAAgJgAAH
CZAABgmgAAT5oAAECaAAAXmwAAJZwAABmdAAAOnQAABJ4AAACeAAAAAnwAAAKAAAACh
AAAAoQAAAKIAAACjAgAApAgAAKUQAACmGAAApYIAAKguAACoPgAAqFAAAKhkAACoegAAq
JIAAKiqAACnwgAAp9wAAKfrAJCZAACDnAAAdJ4AAGSeAABToAAAQ6AAADSiAAAmowAAGq
QAAA6mAAADpgAAAKcAAACoAAAAqQAAAKoAAACrAAAArAAAAK0AAACuAAAAsAAAALEDA
ACzCwAAAtRUAAALggAAC6LQAAuj8AALpVAAC5bAAAUyQAALmcAAC5swAAuccAALnaAlafAAB4
oQAAaKMAAFekAABFpgAANqgAACepAAAbqwAADq0AAAOuAAAAsAAAALEAAACxAAAAswA
AALQAAAC1AAAAtwAAALkAAAC7AAAAvQAAAL8AAADCAAAAxQcAAMoSAADPHgAA0S4AAN
NCAADUWQAA1XEANSLAADTogAA0rUAANLFAHukAABrpwAAWqgAAEmqAAA5rAAAKa8AA
BuxAAAOtAAAArYAAAC4AAAAugAAALwAAAC8AAAAvgAAAMAAAADCAAAAxAAAAmCAAADK
AAAAzQAAANAAAAADUAAAA2gAAAN8CAADIDwAA6x4AAOwwAADuRgAA714AAPB2AADxjAA
A8p4AAPKqAP8AAQD/AAAA/wAAAP8ABwD/ABMA/wAgAP8AKwD/ADUA/wBBAP0ATAD7AFUA
+QBdAPcAZQD2AGsA9ABxAPIAdwDxAHwA7wCBAO4AhwDtAI0A6wCUAOoAnADpAKUA5wCw
AOUAvgDeANIA2ADqANYA/QDUAP8A0gD/AM4A/wDHAP8AxAD/AP8AAAD/AAAA/wAAAP8AA
wD/ABAA/wAcAP0AJgD5ADEA9gA8APMARwDwAFAA7QBZAOsAYADpAGYA5wBsAOUAcgDjA
HcA4gB8AOAAggDfAlkA3QCPANwAlwDaAKAA1gCqANAAtwDLAMkAyQDiAMcA9wDEAP8Aww
D/AMIA/wC9AP8AugD/AP8AAAD/AAAA/wAAAP8AAAD9AAsA9wAXAPAAIgDsACwA6gA3AOYA
QgDiAEsA3gBTANsAWwDYAGEA1gBnANQAbQDSAHA0QB3AM8AfQDOAIMAzACKAMoAkgD
GAJoAwQCkAL8AsAC8AMAAugDYALgA8AC1AP8AtAD/ALMA/wCyAP8AsAD/AP8AAAD/AAAA/
wAAAPgAAADxAAcA6AASAOEAHQDdACgA2QAYANUAPADQAEYAzABOAMkAVQDHAFwAxQ
BiAMMAZwDCAG0AwABYAL8AeAC9AH4AugCEALcAjAC1AJUAswCfALAAqgCuAlkArADNAKo
A6ACoAPwApgD/AKUA/wCkAP8ApAD/AP8AAAD+AAAA9QAAAOsAAADiAAIA2AANANEAGADL
ACMAxwAtAMMANgC/AEAAvABIALoAUAC3AFYAtgBcALQAYgCyAGcAsABsAK4AcgCsAHgAq
wB+AKkAhgCnAl8ApQCZAKMApACHALIAnwDEAJwA3wCaAPUAmQD/AJcA/wCVAP8AIQD/AP8
AAAD0AAAAA6QAAAN4AAADSAAAAyQAIAMIAEwC9AB4AuAAAnALQAMQCxADkArgBCAKwASg
CqAFAPwBWAKUAXACkAGEAowBnAKEAbACgAHIANGB5AJwAgACbAlkAmACTAJYAnwCUA
KwAkgC8AI8A1ACMAO0AigD+AlkA/wCIAP8AiAD/APYAAADpAAAA2wAAAM4AAADFAAAAvQA
DALYADQCwABgArAAiAKgAKwCIADQAogA8AJ8ARACdAEoAmwBRAJoAVgCYAFwAlwBhAJU
AZwCUAG0AKgBzAJAAewCOAIQAJACOAlkAmgCHAKYAhAC2AIAYwCAAOYAfgD4AH0A/wB7A
P8AegD/AO4AAADeAAAAzQAAAMIAAAC5AAAAsgAAAKwACACmABIAoQAcAJ0AJQCZAC4Alg
A2AJQAPgCRAEUAKABLA14AUQCMAFYAiWbCAlkAYgCHAGgAhgBuAlQAdgCBAH8AfwCJAH0
AIQB7AKIAeACxAHYAxB0AN8AcgDzAHAA/wBvAP8AbgD/AOUAAADSCQAAxA4AALkQAACw
DQAAqQYAAKMAAACcAAsAlwAWAJIAHwCOACgAiWAwAlgAOACGAD8AhABFAIIASwCBAFEA
fwBXAH0AXQB8AGMAegBqAHgAcgB2AHsAdACFAHEAkQBvAJ8AbQCuAGsAwABpANkAZgDw
AGUA/QBkAP8AYwD/ANwLAADJFgAAvRsAALIdAACpGwAAohYAAJoOAACtBAMAJQAPAlgAG

QCEACIAGAAqAH0AMgB7ADkAeQFAAHcBRgB2AkWAdANSAHIDWABxBF4AbwRIAG0FbgBrBX
cAaQaCAGcGjwBIB50AYgesAGAHvwBeB9kAXAfvAFsJ/gBaCP8AWQj/ANMWAADDHwAAtyUAA
K0nAACkKgAAmyIAAJQcAACMFAAAhAoHAH4FEgB6BhwAdggkAHMJLABxCjMabgs6AG0LQA
BrDEcAaQ1NAGgNUwBmDVoAZA5iAGIOagBhEHQAXxCAAf0RjQBbEZsAWRGrAFYRvwBUeD
oAUxLyAFIS/wBREv8AURH/AMweAAC+JwAAsi0AAKgwAACfLwAAlysAAI4mAACGHwAAfRcAA
HUQCwBwEBYAbBEfAGkSJwBnEy8AZRQ1AGMUPABhFUIAXxVJAF4WTwBcFIYAWxdeAFkXZ
wBXGHEAVhh8AFQZigBSGZkAUBqpAE4avABMGdcASxrwAEoZ/wBJGf8ASRj/AMglAAC6LgAAr
jQAAKU3AACcNwAAkzQAAIouAACAKAAAdyEAAG4aBgBnGBIAZBkbAGEalwBeGisAXBsxAFob
OABYHD4AVxxFAFUdSwBUHVMAUh1aAFeeYwBPH24ATh95AEwghwBKIJYASSCnAEcgugBFI
NQAQyDuAEMf/wFCH/8BQh7/AcMqAAC2NAAaqzoAAKI9AACZPQAAkDoAAIY1AAB8MAAAcio
AAGgjAQBgHg0AXB8XAFkgIABWICcAVCEuAFIhNABRIjsATyJBAE4jSABMI08ASyNXAEkkYQBI
JGsARyV3AEUmhABDJpQAQialAUamtW+JtEBPSxtATwl/gE8JP8BPCP/AsAwAACzOQAAqT8
AAJ9CAACWQgAAjUAAAIM7AAB4NwAAbjEAAGMqAABaJQkAVSUUAFIHHABPjiQATSYqAEsn
MQBKJzcASCg+AEcoRQBFKEwARCIUAEMpXgFBKmgBQCp0AT8rggE9K5IBOyujATortQE4K88
BNirrATYq/QI2Kf8CNij/Ar01AACwPQAAPkMAAJ1GAACURwAAi0UAAIBBAAB1PAAAajcAAF8xA
ABUKwYATyoRAEwrGQBJKyEARysnAEUsLgBDLDQAQiw6AEAtQgA/LUKBPi5SAT0uWwE7L2Y
BOi9yATkwgAl3MJACNTChAjQwtAEyMM0BMC/qAjAu/AlwLf8CMC3/Aro5AACuQgAApEcAAJtLA
ACSTAAAIUoAAH5HAABzQQAAZz0AAFw4AABQMglASS8NAEYvFgBDMB4AQTAkAD8wKwA9
MDEAPDE3ADsyPwA6MkcBODNPATczWQE2NGQCNDRwAjM0fglxNY4CMDWfAi41sglsNcsBK
zToAisz+wlqMv8CKjH/A7c9AACsRgAAokwAAJIPAAACQUAAAh08AAHxMAABwRwAAZUIAAfK9A
ABNOAAARDUJAEA1EwA9NRsAOzUiADk1KAA4Ni4ANzY1ADU3PAA0N0QBMzhNATE4VwEw
OGECLzluAi05fAlS0owCKjqeASk6sQEnOcoBJjnnASU4+gllNv8CJTb/A7VBAACqSgAAoFAAAJd
UAACOVQAAhVQAAHPRABtTAAAYkgAAFZEAABLPwAAPzsGADo6EAA4OhgANjsfADQ7JQA
yOywAMTwzADA8OgAuPEIALT1KASw9VAEqPV8BKT5sASg+egEmPooBJT+cASM/rwEiPsgBIT
3mASA8+glgO/8CIdr/ArJGAACnTgAAAnIQAAJVYAACNWQAAG1kAAHhXAABrUgAAyE8AAFRLA
ABJRwAAPUICADVADAAyQBUAMEAcAC5AlwAsQSkAK0EwACpBNwApQj8AJ0JIACZCUgAIQ1
0BI0NqASJDeAEhRIgBH0SbAB5ErgAcQ8YAG0LkABtB+QEbQP8BGz//Aq9LAACIUwAAAnFoAAJ
RdAACLXwAAgV8AAHddAABqWQAAXIYAAFNSAABGTgAAOkoAAC9GCAArRhEAKUYZACdGI
AAmRyYAJUctACRHNAaISDwAlUhFACBITwAfSVoAHUlnABxJdgAbSYyAGUmZABhJrAAWScQ
AFUjjABVH+AAVRf8BFUX/AaxRAACjWQAAMl8AAJJjAACJZQAAGGUAAHVKAABoYAAAXF0AA
E9ZAABDVQAAN1EAACtOAwAkTQwAlU0VACBNHAAfTiAHU4pABxOMAAbTjgAGk9BABIPSwA
XT1cAFk9kABVPcwATUIMAEICWABBPqgAOT8EADE7fAA1N9gAOS/8ADkr/AKIXAACgXwAAm
GUAAJBpAAClawAAfmwAAHRrAABnaAAAWGQAAEtgAAA+XQAAMloAACdXAAAEVQcAGVUQ
ABhVFwAWVR4AFVUIABRWLAATVjQAEIY9ABBWRwAPVIMADVZgAAxXbwAKV4AACVaSAAd
WpgAFVbsABFTYAART7wAFUv4ABIH/AKZeAACdZgAAImwAAI5wAACGcgAAfXMAAHJyAABibg
AAU2sAAEZoAAA6ZQAALmIAACNgAAAZXwEAEV0JAA1dEgAMXhkAC14gAApeKAAJXjAAB14
5AAZeQwAFX08AA19cAAJfawAAXnsAAF6OAABeoQAAXbYAAFzRAABb7AAAWvkAAFn/AKNIA
ACbbQAAIHMAAIx3AACEegAAe3oAAGx4AABcdQAATXIAAEbvAAA0bQAAKGsAAB5qAAAVaA
AAC2gFAARnDQABaBUAAGgcAABolwAAaCsAAGg0AABoPgAAaEoAAGhXAABoZgAAZ3YAAG
eJAABmnQAAZbEAAGTLAABj5wAAYvcAAGH/AKBuAACydgAAknsAAIt/AACCgQAAdIAAAGV+
AABWewAAR3kAADI3AAAtDQAAlnQAABhzAAAOCgAABXICAABYcGAAchEAAHIXAABYHgAAci
UAAHluAABYOAACkMAAHJRAABYyAAAcXAAAHGDAABwlwAAB6wAAG7FAABt4gAABPMAAG

z8AJ13AACWfgAAkIMAAImHAAB8hgAAbYUAAF2EAABOggAAP4EAADJ/AAAmfgAAG34AABF9
AAAlfAAAAHwAAAB8BQAafAsAAH0SAAB9GAAafR8AAH0nAAB9MAAAfTwAAH1JAAB9WQAaf
GoAAHx9AAB7kQAAeqcAAHm/AAB43QAAd/EAAHb7AJqAAACUhwAAjosAAIKMAABzjAAAY4sA
AFOKAABEiQAAANogAACmIAAAadhwaAE4cAAAmHAAAAhgAAAlcAAACHAAAAhwUAAIcLAACIE
QAAiBgAAIgfAACJKAAAiTMAAIIAAACJTWAaiGEAAIh0AACHigAAh6EAAIa3AAACE1AAAhOwAAI
P4AJiKAACTjwAAh5EAAHmRAABpkQAAWJEAAEmQAAA6kAAALJAAACCQAAAVkAAACpAAA
ACQAAAAkAAAAJEAAACRAAAAKgAAAJIDAACCTCQAAkw8AAJQWAACVHgAAISgAAJU1AACV
RAAAIVYAAJVPAAACVfwAAIJYAAJStAACTxwAAkuQAAJHzAJaSAACLIAAAfZYAAG6WAABdlwA
ATJcAAD2XAAAumAAAIzgAABWZAAAKmQAAAJoAAACaAAAAmgAAAJsAAACcAAAAAnAAAAJ
0AAACeAAAAAnwUAAJ8MAAChFAAAoh0AAKMoAACjNwAAo0kAAKNdAACjcgAAo4sAAKOjAACi
uwAAotQAAKHnAl6YAACBmgAAcpsAAGGcAABQnQAAQJ0AADCFAAAjnwAAFqAAAAqiAAAAo
wAAAKMAAACjAAAApAAAAKUAAACmAAAApWAAAKgAAACpAAAAqWAAAKwAAACuBwAAsB
AAALibAACzKQAAszoAALNOAAC0ZAAAtHwAALSVAAC0rAAAtMMAALPVAISdAAB1nwAAZaA
AAFSiAABCowAAMqUAACOMAAAXpwAACqkAAACqAAAArAAAAK0AAACtAAAAArgAAAK8AAA
CxAAAAsgAAALMAAAC1AAAAtwAAALoAAAC8AAAAvwlAAMMMMAADIGQAAYigAAMs8AADLUw
AAymwAAMuGAADLnQAAY7AAAMzBAHiiAABopAAAV6YAAEaoAAA2qgAAJawAABiuAAALsAA
AALIAAAC0AAAAtQAALcAAAC3AAAAuQAAALsAAAC9AAAAvwAAAMEAAADDAAAAxgAAAM
kAADNAAAA0gAAANgAAADeCgAA5RgAAOcqAADpQAAA6lgAAOtwAADriQAA6p4AAOmsAP8
AAAD/AAAA/wAAP8ABAD/ABAA/wAcAP8AJwD/ADIA/gA9APsARwD4AFEa9gBZAPQAYADxA
GYA7wBsAO4AcQDtAHcA7AB8AOsAggDqAlgA6ACPAOcAlwDIAKAA4gCrANwAuADXAMsA1Q
DmANMA+wDRAP8AzwD/AMkA/wDCAP8AvgD/AP8AAAD/AAAA/wAAP8AAAD/AAwA/gAYAPk
AlwD2AC0A8wA4AO8AQwDsAEwA6QBUAOYAWwDjAGEA4QBnAOAAbQDeAHIA3QB3ANwAf
QDaAIMA2QCKANYAkGDRAJsAzQCIAMoAsQDHAMMAxQDdAMMA9ADBAP8AvwD/AL0A/wC3
AP8AtAD/AP8AAAD/AAAA/wAAP8AAAD5AAgA8AAUAOWAHwDoACkA5QAZAOEAPQDcAEcA
2ABPANQAVgDSAFwA0ABiAM4AaADNAG0AywByAMoAeADIAH4AxQCFAMEAjAC+AJUAvQCg
ALsAqwC4ALsAtgDSALQA7ACyAP8AsAD/AK8A/wCtAP8AqgD/AP8AAAD/AAAA/AAAAPQAAAD
qAAMA4gAPANwAGgDWACQA0gAuAM4AOADJAEExgBJAMMAUQDBAFcAvwBdAL4AYgC8A
GcAugBtALcAcgC1AHgAswB/ALIAhwCwAJAArgCaAKwApgCqALMAqADHAKYA4wCkAPkAogD/
AKEA/wCfAP8AngD/AP8AAAD6AAAA8AAAAOYAAADaAAAA0QAKAMoAFQDFACAAwAApAL0
AMgC6ADsAtgBDALQASwCyAFEARwBXAK0AXQCrAGIAqgBnAKkAbACnAHIApgB5AKUAgQCjA
IoAoQCUAJ8AoACcAK0AmgC+AJgA2ACVAPIAkWd/AJEA/wCRAP8AkAD/APsAAADvAAAA4wA
AANYAAADLAAAAwgAGALwAEAC2ABsAsgAkAK8ALQCsADUAqAA9AKUARQCjAEsAogBRAK
AAVwCfAFwAnQBhAJwAZwCbAG0AmQB0AJcAewCWAIQAKwCOAJEAmgCPAKcAjAC2AlOAzA
CHAOkAhQD8AIQA/wCDAP8AggD/APEAAADjAAAA0wAAAMcAAAC/AAAAtgABALAACwCqABU
ApgAfAKIAJwCeADAAnAA3AJkAPwCXAEYAIQBMAJQAUQC SAFYAkQBcAI8AYQCOAGcAjABu
AlOAdgCIAH4AhgCJAImAIACBAKEAfwCwAHwAxAB6AOAAeAD2AHcA/wB2AP8AdQD/AOgAAA
DWAAAAxwAAALwAAACzAAAArAAAAKYABgCgABAAMwAZAJcAlgCTACoAkAAyAI4AOQCLAE
AAiQBGAIGASwCGAFEAhABWAIMAXACBAGIAfwBpAH4AcQB8AHkAeQCEAHcAjwB1AJwAcgC
rAHAAvgBuANgAbADwAGsA/gBpAP8AaAD/AN4AAADLBwAAvgwAALQMAACrCQAAowMAAJw
AAACWAAoAkQATAIwAHACIACQAQhQAsAIAMwCAADoAfgBAAHwARgB6AEwAeQBRahcAVw
B1AF0AdABkAHIAbABwAHUAbQB/AGsAiWbPAJgAZwCnAGQAuQBiANEAYADqAF4A+wBdAP8
AXQD/ANMLAADEFAAAtxkAAK0aAAcKfWAAAnBMAAJULAACNAQIAhwANAIIFgB9AB8AegAm

AHcALgB0ADQAcgA7AHEAQQBvAEYAbQBMAGwAUgBqAFkAaABgAGYAaABkAHEAYgB7AG
AAiABeAJUAXACIAFoAtgBYAc0AVgHoAFQB+ABTf8AUwH/AMwVAAC9HgAAsiMAAKgkAACfl
wAAIh4AAI4YAACGEAAAfgcFAHgAEABzABkAbwAhAGwCKABqAy8AaAM1AGYEOwBkBUEAY
gVHAGEGTgBfBIQAXQdcAFwHZABaCG0AWAI4AFYJhQBUCpQAUgqkAFAKtQBOCs0ATAvnAE
sL+QBKC/8ASgr/AMYdAAC4JgAArSsAAKQtAACaLAAAkigAAIkiAACAHAAAdxQAAG8LCABpCB
MAZQobAGILlwBgDCoAXQwwAFsNNgBaDTwAWA5DAFcOSQBVD1AAVBByAFIQYQBQEWoA
TxJ2AE0SgwBLE5IASROjAEcTtABFE80ARBPpAEMT/ABDE/8AQhL/AMljAAC1LAAAqjEAKA0A
ACXNAAAJjAAAIUrAAB7JQAACH4AAGgXABgEg0AXBIXAFkTHwBXFCUAVRQsAFMVMgBRFT
gATxY/AE4WRQBMF0wASxdUAEkYXQBIGGcARxIzAEUZgABDGpAAQhqqAEaasgA+GssAPBr
nADwa+wA8Gf8AOxn/AL4pAACxMgAApzgAAJ06AACUOgAAizcAAIEyAAB3LQAAbSYAAGMgA
ABZGQgAVBkTAFEZGwBPGilATRsoAEsbLwBJGzUASBw7AEYcQgBFHUkARB1RAELeWgBBH
mUAQB9wAD4ffgA9II0AOyCeADkgsAA3IMgANiDIADUg+gA1H/8ANR7/AbsvAACuNwAApD0AAJ
s/AACSPwAAiT0AAH44AAB0MwAAaS0AAF4nAABUIQQATR4OAEofFwBIIB8ARiAIAEQgKwBCI
TEAQSE3AD8iPgA+IkYAPSNOADwjWAA7JGIAOSRuADglfAA2JYsANSacADMmrwAxJcYALyXk
AC8I+QEvJP8BLyP/AbgzAACsPAAOkEAAJIEAACQRAAAhklAAHw+AABxOQAAZjQAAFSuAAB
QKAAASCULAEQkFABBJRsAPyUiAD0IKAA8Ji4AOiY0ADknOwA4J0MANyhLADYovQA0KWAA
MylsADlqeQAwKokALiubAC0rrQArKsQAKSriACkp+AEpKP8BKSj/AbU4AACqQAAAoEUAAJdIAA
COSQAAhEcAAHpDAABuPgAAYzoAAFg1AABNLwAAQyoHAD4pEQa7KhgAOSofADcqJQA1Kis
ANCsxADMsoAAyLEAAMS1JADAtUwAvLI0ALS5qACwvdwAqL4cAKTCZACcwrAAIL8MAJC/gAC
Qu9gEkLf8BJCz/AbM8AACoRAAAankoAAJVNAACMTQAAg0wAAHhJAABsRAAAYD8AAFU7AAB
KNgAAPzEEADkvDQA2LxUAMy8cADEvlgAwMCgALjAvAC0xNgAsMT4AKzJGACoyUAApM1sAK
DNnACY0dQAINIUAlzSXACI0qgAgNMEAHZTfAB8z9gAfMv8BHZH/AbBAAACmSAAAnE4AAJNR
AACLUgAAgVEAAHZOABpSQAAXkUAAFNBAAABIPQAAPTgAADM1CgAwNBIALTUZACw1IAA
qNSYAKTYtACg2MwAnNzsAJdEACQ3TgAjOFkAljhlACE5cwAfOYMAHjmVABw5qQAbOb8AGTj
dABk49QAaNv8BGjX/Aa1FAACjTQAAMlIAAJJVAACJVwAAf1YAAHRUAABoTwAAxEwAAFFIAA
BGRAAAO0AAAC87BgApOg4AJzoWACY7HQAkOyMAIzsqACI8MQAhPDgAIDxBAB49SwAdPV
YAHD1jABs+cQAZPoEAGD6TABY+pwAVPr0AEz3bABQ88wAUO/8AFDv/AatKAACHUgAAmFcA
AJBbAACHXAAAFIwAAHNaABmVgAAW1MAAFBQAABESwAAOEcAACxDAGAkQQsAIUATAB9
BGgAeQSAAHUEnABtCLgAaQjUAGUI+ABhCSAAXQ1MAFUNgABRDbgATQ38AEUSRABBDpQ
ANQ7oADELWAAxB8QANQP8ADj//AKhPAACfVwAAII0AAI5gAACGYgAAfGIAAHJgAABIXQAAWI
oAAE1WAABAUgAANE4AACILAAAfSAYAGUcOABhHFgAXSB0AFUgjABRIKgATSDIAEkk6ABFJ
RQAPSVAADUIdAAxKawALSnsACUqOAAAdJoQAGSbYABUjRAAVH6gAFRvsABkX/AKVVAACdX
QAAIWMAAI1nAAACEaQAAe2kAAHBoAABkZQAAVWEAAEhdAAA8WQAAMFYAACVTAAAbUQE
AE08JAA9PEQANUBgADFafAAtQJgAKUC4ACVA3AAAdRQQAGUUwABVFZAANRZwACUXcAA
FGJAABQnQAAULEAAE/KAABN5wAATfYAAEz/AKJcAACaZAAAk2oAAIttAACDcAAAenAAAG9v
AABfawAAUGcAAENkAAA3YQAAKI4AACBcAAAWWgAADVKGAAZYDQADWRUAAIkcaAFZlWA
AWSOAfFkzAABZPQAAWUgAAFIVAABZYwAAWXMAAFiFAABYmQAAV60AAFbGAABV4gAAV
PUAAFP+AJ9kAACyBAABkXEAAlI1AACBdwAAeHcAAGI1AABZcgAASm4AAD1sAAAXaQAAJW
cAABtIAAARZAAACGMDAABjCgAAYxEAAGMYAABiHwAAYiYAAGluAABiOAAAYkMAAGJQAAB
iXgAAYm4AAGGBAABhIQAAyKkAAf/BAABd3wAAXfIAAFz8AJxtAACWdAAAj3kAAIh9AACafwA
AcX0AAGJ6AABTeAAARHUAADZzAAAqcQAAH3AAABVvAAALbgAAAm0AAABtBwAAAbQ0AAG
0TAABtGgAABSEAAG0oAABtMgAAbT0AAGxKAABsWAAAbGkAAGt7AABrjwAAaqUAAGm8AAB

n2gAAZvAAAGX7AJp2AACUfAAAJoEAAIaEAAB5hAAAaoIAAFqBAABKfwAAPH0AAC57AAAieGAGHkAAA15AAAEeAAAAHcAAAB3AwAADwgAAHcOAAB4FAAAeBsAAHgiAAB4KwAAeDYAAHhDAAB3UQAAd2IAAHZ0AAB2iQAAdaAAAHs2AABz0wAAcuwAAHH4AJd/AACShQAAjlkAAH+KABwiQAAyIgAAfCHAABbhgAAM4UAACaEAAAagwAAEIMAAAWDAAAaggAAAIIAAACCAAAAgglAAIIHAACDDQAAGxQAAIMbAACElWAAhC0AAIQ5AACDSAAAag1oAAIJtAACCGgAAgZkAAICvAAB/ygAAfucAAH31AJaIAACQjQAAhI4AAHaPAABmjgAAVY4AAEWNAAA3jQAAKYwAABYMAAARjAAABowAAACMAAAAJAAAAIwAAACMAAAAJQAAAI0AAACOBQAAjgsAAI8SAACQGgAAkCMAAJAvAACQPQAakE8AAJBIAACQdwAAj48AAI6oAACNwAAAJd4AAIzwAJJSRAACIkWAAepQAAGuUABalAAASZQAADqUAAaqIAAAHpQAABKVAAGIQAAAjYAAACVAAAAIgAAAJYAAACXAAAAIwAAAJgAAACZAAAAmgEAAJsHAACcDwAAAnRgAAJ4jAACeMQAAAn0IAAJ5WAACeawAAAnoMAAJ6cAACdtAAAncwAAJzkAlYWAAB+mAAAb5kAAAF6ZAABNmgaAPZoAAC2bAAAFnAAAE50AAAEeAAAAAnwAAAJ8AAACfAAAAoAAAAKEAAACiAAAAowAAAKMAAACIAAAAPgAAAKcAAACoAwAAqgwAAKwXAACTlwAArjMAAK5HAACvXQAAR3UAAK+PAACupwAArr4AAK7UAIKbAABznQAAYp4AAFGfAABAoAAAL6IAACCjAAAUpAAAB6YAAACnAAAAqAAAAKkAAACpAAAAqgAAAKsAAACsAAAArgAAAK8AAACxAAAAsgAAALQAAAC3AAAAugAAAL0HAADBFAAAwiQAAMI3AADDtgAAxGYAAMR+AADFlwAAxawAAMW+AHagAABmogAAVaMAAEoIAAAypwAAIqkAABWrAAAHrQAAAK8AAACwAAAAsQAAALIAAACzAAAAtAAAALYAAAC3AAAAuQAAALsAAAC+AAAwAAAAAMMAAADHAAAAyWAAANAAAAADYBQAA3xMAAOEIAADjOQAA41MAAOJtAADihgAA45sAAOOrAP8AAAD/AAAA/wAAAP8AAQD/AA0A/wAYAP8AlwD/AC4A/AA5APgAQwD1AEwA8gBUAO8AWwDtAGEA7ABnAOoAbQDpAHIA6AB3AOcAfQDmAlMA5ACKAOIAkgDdAJsA2ACIANUAsgDTAMQA0ADgAM4A+ADMAP8AywD/AMUA/wC8AP8AuAD/AP8AAAD/AAAA/wAAAP8AAAD/AkA+gAUAPYAIADzACoA8AA0AOsAPgDnAEcA4wBPAAOAAVgDeAF0A3QBiANsAaADZAG0A2ABYANYAeADUAH4A0ACFAMwAjADJAJUAxwCgAMUArADDALwAwADVAL4A8AC8AP8AuwD/AlkA/wCyAP8ArgD/AP8AAAD/AAAA/wAAAP0AAADzAAUA7AARAOCAGwDjACYA4AAvANoAOQDVAEIA0QBKAM4AUQDMAFgAygBdAMgAYwDHAGgAxABtAMIAcgC/AHgAvQB/ALsAhwC6AJAAuACaALYApGc0ALUAsQDKAK8A5wCtAP0ArAD/AKoA/wCnAP8ApAD/AP8AAAD/AAAA+QAAAO8AAADkAAAA3AAMANUAFwDQACEAywAqAMcAMwDDADwAwABFAL0ATAC7AFIAuQBYALYAXQC0AGIASgBnALEAbQCwAHMArgB5AK0AgQCrAlOaqQCUAKgAoACmAK4AowDBAKEA3QCfAPYAnQD/AJsA/wCaAP8AlwD/AP8AAAD2AAAA6wAAAN4AAADTAAAAyWAAHAMQAEgC+ABwAugAlALcALgC0ADcAsAA/AK0ARgCrAEwAqQBSAKgAWACmAF0ApQBIAKQAZwCiAG0AoQB0AJ8AewCeAlQAnACOAJoAmgCXAKcAlQC4AJIA0ACQAO0AjgD/AlwA/wCMAp8AiW/APcAAADqAAAA3AAAAM4AAADFAAAAvAADALUADQCwABcArAAhAKKAKQCIADeAogA5AKAAQACeAecAnABMAJsAUgCZAFcAmABcAJYAYgCVAGgAkwBuAJIAdgCQAH4AjgCIAIsAlACJAKEAhgCxAlQAxgCCAOQAgAD6AH4A/wB+AP8AfQD/AOWAAADcAAAAzAAAAAMEAAAC5AAAAAsAAAAKoACACIABIAoAAcAJwAJACZACwAlgAzAJQAOgCRAEEAkABHAI4ATACMAFEAiWbXAIkAXACIAGIAhgBpAlQAcACCAHkAgACDAH4AJwB8AJwAeQCrAHcAvgB1ANsAcwDzAHEA/wBwAP8AbwD/AOEAAADPAAAAwQAAALcAAACuAAAApwAAAKAABACaAA0AlQAWAJEAHwCNACYAigAtAlcANACFADsAgwBBAIEARgCAAEwAfgBRAH0AVwB7AF0AEQBkAHcAawB1AHQAkwB+AHEAigBvAJcAbACmAGoAtwBoANEAZgDsAGQA/QBjAP8AYwD/ANUAAADGBQAAuQkAAK8JAACmBgAAngAAAJcAAACRAAgAiWARAIYAGQCCACEAfGaoAHsAlwB5ADUAdwA7AHUAQQB0AEYAcgBMAHEAUgBvAFgAbQBfAGsAZgBpAG8AZwB5AGUAhQBjAJMAYACiAF4AsgBcAMkAWgDmAFkA9wBYAP8AVgD/AM0KAAC+EGAAshYAAKgxAAACfQAAAlw8AAI8HAACIAAEAgQALAHwAFAB3AB

wAdAAjAHEAKgBuADAAbAA2AGoAPABoAEEAZwBHAGUATQBjAFMAYgBaAGAAYgBeAGsAX
AB1AFoAgQBYAI8AVgCeAFQArwBSAMQUADfAE4A9QBNAP8ATAD/AMYUAAC4HAAArSAAA
KMiAACaIAAAkhsAAIkVAACADQAAeAQEAHIADQBtABYAaQAeAGYAJABjACsAYQAxF8ANg
BdADwAXABCAFoASABYAE4AVwBWAFUAXgBTAWcAUQJyAFACfgBOA4wATAScAEoErQBIA
8IARgTeAEQE8QBEBP4AQwT/AMAcAACzJAAAqSkAAJ8qAACWKQAAjSQAAlQfAAB7GQAACH
EAAGkJBwBjAxAXwMYAFwEHwBZBSYAVwUsAFUGMgBTBzcAUgc9AFAlQwBOCEoATQISA
EsJWgBKCmQASAtvAEYLfABEDIsAQgybAEAMrQA/DMMAPQzeADwN8wA8DP8AOwz/ALwiAA
CwKgAApi8AAJwxAACTMAAAii0AAIAoAAB2lgAAbBsAAGMUAABaDQkAVQsSAFIMGgBQDSEA
TQ0nAEsOLQBKDJMASA85AEcQPwBFEEYARBFoAEIRVwBBEmEAPxJtAD4TegA8FIkAOxSaA
DkUrAA3FMIANRTfADQU9QA0FP8ANBP/ALkoAACtMAAAozUAAJk4AACQNwAAhzQAAHwwAA
ByKQAAaCMAAF4dAABUFwMATRMNAEoTFgBHFB0ARRQjAEQVKQBCFS8AQBU1AD8WOWA
+FkMAPBdLADsYVAA6GF4AOBlqADcZdwA1GocAMxqYADlaqgAvGr8ALhrcAC4a9AAtGv8ALRn
/ALYtAACqNgAAoDoAAJc9AACOPAAAhDoAAHk1AABvMAAAZCoAAFOlAABQHwAARxoJAEMZ
EgBAGRkAPhogaDwaJgA7GywAORsyADgbOAA3HD8ANh1IADQdUQAzhIwAMh5nADAfdQAvH
4QALSCWACsgqAApIL0AJx/aACcf8wAnHv8AKB7/ALMyAACoOgAAAnj8AAJVBAACMQQAAGj8A
AHc7AABsNgAAYTEAAfCrAABMJgAAQIaFAD0eDgA6HxYAOB8dADYflwA0ICgAMyAuADEgNQ
AwIT0ALyJFAC4iTgAtI1kAKyNIACokcwAoJYIAJyWUACUlpGajJbsAlSTYACIk8gAil/8Alil/ALA3AA
CmPgAAAnEMAAJNGAACKRgAAgEQAAHVAAABqPAAAXzcAAAFQyAABJLAAAPycCADckCwA0J
BMAMSQZAC8kHwAtJCUALCUrACslMgAqJJoAKSdCACgnTAAAnKFcAJihjACQpcQAJKYAAISqSA
CAqpQAeKboAHCnWABwp8AAAdKP8AHSf/AK47AACkQgAAmkgAAJFKAACISgAAf0kAAHRGAA
BoQQAAXD0AAFE4AABHNAAAPS8AADlqBwAuKRAAKykXACkPHQAoKiMAJyopACYrMAAIKzc
AlyxAACIsSQAhLVQAIC1hAB8ubgAdLn4AHC6QABouowAZLrgAFy7UABct7wAXLP8AGCv/AKw/
AACiRwAAmEwAAJBOAACHTwAAfU4AAHJLAABmRgAAWklAAFA/AABFOwAAOzYAADAYBAA
oLwwAJS8UACQvGgAiLyAAITAnACAwlQAfMTUAHJE9AB0xRwAcMIIAGjJeABkzbAAYM3wAFjO
OABUzogATM7YAETLSABEy7gASMf8AEjD/AKIEAACgSwAAI1AAAI5TAACFVAAAe1MAAHFRA
ABkTAAAWUkAAE9GAABEQgAAOT4AAC05AAAJNQkAHZURAB41FwAdNR4AGzYkABo2KwAZ
NjIAGDc6ABc3RAAVN08AFDhcABM4agASOHOAEDiMAA44oAAMOLQACjjOAAo36gALNvwADD
X/AKdJAACeUAAAIvUAAI1YAACEWQAAelkAAG9XAABjUwAAWFAAAE5NAABCSQAANKQAAC
pAAAAGPQUAGTsNABc7FAAWPBsAFTwhABQ8KAASPC8AET03ABA9QQAOPUwADT5ZAAst
ZwAKPncACT6JAAc+nAAFPbAABD3JAAQ85QAEo/cABTr/AKROAACbVgAAk1sAAIteAACCXw
AAeWAAAG5eAABiWwAAV1gAAEpUAAA+TwAAMksAACZIAAAcRQAafEMIABBCEAAOQhcAD
UMeAAtdJQAKQywACUM0AAhEPgAGREkABURWAAREZAACRHMAAESFAABEmAAAQ60AA
EPEAABC4gAAQfQAAED+AKJUAACZXAAAKWEAAIlkAACBZgAAeGYAAG1IAABhYgAAU14AA
EVaAAA5VgAALVMAACJQAAAYTgAAEEsFAAhKDAAFSxQABESbAAJLIgABSyKAAEsxAABLOW
AAS0YAAEtSAABLYAAAS3AAAEuBAABLIQAASqkAAEnAAABI3QAAR/IAAEb8AJ9bAACXYwAA
kGgAAIhrAACAbQAAd24AAGxsAABcaAAATWQAAEBhAAA0XQAAJ1sAAB1YAAAUvgAACIUDA
AJUCwAAVBEEAFQYAABUHgAAVCUAAFQTAABUNwAAVEIAAFROAABTXAAAU2wAAFN9AAB
SkQAAUqYAAFG9AABP2gAATu8AAE37AJxjAACVagAAjm8AAldzAAB/dQAAdXUAAGZyAABWb
gAAR2sAADpoAAAtZQAAlmMAABhhAAAOYAAABF8BAABeCAAAXg4AAF4UAABdGgAAXSEA
AF0pAABdMgAAXT0AAF1JAABcVwAAXGcAAFX5AABbjQAAWqlAAFM5AABY1QAAV+4AAFb6A
JlrAACTcgAAjHcAAIV7AAB9fAAAb3oAAF93AABPdQAAQHIAADNvAAAnbgAAHGwAABFrAAAH
aQAAAGkAAABoBAAAaAoAAGgQAABoFgAAaBwAAGcjAABnLAAAZzcAAGdDAABnUgAAZmlA

AGZzAABliAAAZJ4AAGO0AABi0QAAYesAAGD5AJd0AACRewAAi38AAISCAAB2gQAAZ38AAF
Z9AABHewAAOXkAACT3AAAfdgAAFHUAAAp0AAAAcWAAAHMAAABzAAAAcWUAAHMKAAABzE
AAAcxYAAHMDAABzJQAACjAAAHl8AABYswAAclSAAHFtAABxgQAACJgAAG6uAABtyQAAbOcA
AGv3AJV+AACQgwAAiocAAHyHAABthgAAXYUAAE2DAAA+ggAAMIEAACKAAAAAXfwAADH8AA
AF+AAAAfQAAAH0AAAB9AAAAfQAAAH0EAAB+CQAAfhAAAH4WAAB+HgAAfygAAH40AAB+Q
gAAfIIMAAH1IAAB9egAAfJEAAHuoAAB6wQAAeeAAAHjzAJSHAACOjAAAgowAAHONAABjjAAA
UosAAEKKAAAzigAAJokAABmJAAANiAAAAogAAACHAAAAhwAAAlgAAACIAAAAIAAAAIkAAA
CJAQAaigcAAIoNAACLFQAajB4AAIwpAACLNwAAi0gAAItbAACKcAAAIogAAImgAACluQAAh9U
AAIfAJKPAACGkQAaeJIAAGiRAABXkQAARpEAADaRAAAAnQAAG5EAAA6RAAACkQAAAjIA
AACRAAAAKgAAAJIAAACSAAAAKwAAAJQAAACUAAAAIQAAAJYDAACXCgAAmBMAAJodAAC
aKwAAmjsAAJpPAACZZAAAmXsAAJmVAACYrgAAI8oAAJbjAlmUAAB8lgAAbJcAAFuXAABKlwA
AOZcAACmYAAAcMAAAD5kAAAOaAAAAmwAAAJsAAACbAAAAAnAAAAJwAAACdAAAAngAAA
J8AAACgAAAAoQAAAKIAAACkAAAApQcAAKcRAACpHgAAqS0AAKIAAACqVgAAqm4AAKqIAA
CpoAAaqbcAAKnQAH+ZAABwmwAAX5sAAE6cAAA9nQAALJ8AAB2gAAAQoQAAA6IAAACjAA
AApAAAAKUAAACIAAAApGAAAKcAAACoAAAAqQAAAKsAAACsAAAAArgAAALAAAAcXAAAAAtA
AAALYEAAC5EAAAux8AALwxAAC9RwAAvI8AAL54AAC+kQAAvqgAAL68AHOeAABJnwAAUqE
AAECjAAAvpAAAH6YAABKoAAAEqQAAAKsAAACsAAAAArgAAAK4AAACvAAAAsAAAALEAAAC
zAAAAAtQAAALcAAAC5AAAAuWAAAL4AADDBAAAAxQAAAMoAADQAAAA2A0AANkgAADZNg
AA2U8AANpnAADcgAAA3JcAAN2oAP8AAAD/AAAA/wAAAP8AAAD/AAkA/wAVAP8AIAD9ACoA
+QA1APQAPwDwAEgA7QBQAOsAVwDpAF0A6ABjAOYAaADIAG0A5ABYAOmAeADgAH4A3AC
FANgAjADUAJUA0gCgANAArQDOAL4AzADYAMkA8wDHAP8AxgD/AMAA/wC5AP8AswD/AP8A
AAD/AAAA/wAAAP8AAAD7AAUA9gARAPMAHADvACYA6wAwAOUAOgDhAEMA3gBLANsAUg
DZAFgA1wBeANUAYwDTAGgA0ABtAM0AcwDKAHgAxwB/AMUAhwDEAJAAwgCbAMAApWC+AL
LYAvADNALkA6wC3AP8AtgD/ALQA/wCuAP8AqQD/AP8AAAD/AAAA/wAAAPcAAADuAAIA6AAN
AOMAGADdACIA2AArANMANQDPAD4AywBGAMgATQDGAFMAxABYAMEAXgC+AGMAvABoA
LoAbQC5AHMAuAB6ALYAgQC1AlOAswCVALEAoQCvAK8ArQDDAKoA4QCpAPoApwD/AKUa/
wChAP8AnQD/AP8AAAD+AAAA9QAAAOgAAADfAAAA1gAJAM8AFADJAB4AxQAnAMEALwC9
ADgAugBAALcARwC0AE0AsQBTALAAWACvAF0ArQBjAKwAaACrAG4AqQB0AKgAfACmAIUAp
QCPAKMAmwCgAKkAngC6AJwA1ACZAPEAlwD/AJYA/wCUAP8AkQD/AP0AAADyAAAA5QAAA
NgAAADNAAAAxQAEAL0ADwC4ABkAtAAiALEAKgCtADIAqgA6AKcAQQCmAEGApABNAKIAUw
ChAFgAoABdAJ4AYgCdAGgAnABuAJoadgCYAH8AlgCJAJQAIQCSAKIAjwCyAlwAyQCKAOgAi
QD+AlcA/wCGAP8AhQD/APIAADjAAAA1AAAAMgAAAC/AAAAtgAAALAACgCrABQApGAdAKIA
JQCgAC0AnQA0AJOAOWCYAEIAlwBIAJUATQCUAFIAkgBXAJEAXQCPAGIAjgBpAlwAcACKAH
kAiACDAIUajgCDAJwAgQCrAH4AvwB8AN0AegD3AHgA/wB3AP8AdwD/AOYAAADUAAAAxgAA
ALsAAACzAAAAqwAAAKQABgCfABAAMgAYAJYAIACtACgAKAAvAI4ANgCLADwAigBCAlgAR
wCGAEwAhQBSAIMAVwCCAF0AgABjAH4AawB8AHMAegB9AHgAiQB1AJYAcwCIAHEAtwBvA
NIAbADvAGsA/wBqAP8AaQD/ANkAAADJAAAAvAAAALEAAACpAAAAoQAAAJJoAAQCUAAsAjw
ATAIsAGwCHACMAhAApAIEAMAB/ADYAfQA8AHsAQQB5AEcAeABMAHYAUgB1AFcAcwBeAH
EAZgBvAG4AbQB4AGsAhABoAJEAZgCgAGQAsQBiAMgAYADnAF4A+wBdAP8AXAD/AM8AAA
DAAwAAtAcAAKoGAACHAwAAmQAAAJEAAACLAAYAhQAOAIAAFgB8AB4AeAAKAHUAKwBzA
DEAcQA3AG8APABtAEEAbABHAGoATABoAFIAZwBZAGUAYQBjAGkAYQBzAF4AfwBcAlwAW
gCcAFgArABWAMIAVADfAFMA9QBSAP8AUQD/AMYIAAC5EAAARQAACKQUAACaEgAAkgsAAI

oEAACCAAAAwAJAHYAEQBxABkAbQAFAGoAJgBnACwAZQAxAGMANwBiADwAYABCAF4AR
wBdAE4AWwBUAFkAXABXAGUAVQBvAFMAewBRAIkATwCYAE0AqQBLAL0ASgDYAEgA7wB
HAP0ARgD/AMATAACzGgAAqB4AAJ8fAACWHQAAjRgAAIQSAAB7CgAAcwACAGwACwBnAB
MAYwAbAGAAIQBdACcAWgAsAFkAMgBXADcAVQA9AFQAQwBSAEkAUABQAE8AWABNAGE
ASwBrAEkAeABHAIUARQCVAEMApGBCALkAQADSAD8A7AA9APsAPQD/ALsbAACvlgAApCYA
AJsoAACSJgAAiCEAAH8cAAB2FgAAbA4AAGMGBQBdAA0AWQAVAFYAHABTACIAUAAoAE8
ALQBNADIASwA4AEoBPgBIAUUARgJMAEUCVABDA14AQQRoAEAFdQA+BYMAPAaTADoGp
QA5BrgANwXRADUG6gA0BvkANAX/ALchAACrKAAAoS0AAJgvAACOLQAaHSoAAHsIAABxHw
AAZxgAAF4RAABVCgcATwYPAEWFFgBBh0ARwcjAEUHKABDCC4AQgg0AEAJOGA/CUAAPQ
pIADwKUQA6C1sAOQxmADcMcwA1DYIAMw2SADENpQAvDbgALQ3SACwN7AAsDfsALA3/AL
QnAACpLgAAnzMAAJU1AACMNAAAgjEAAHgsAABuJgAAyYAAAFkaAABQFAAARw4JAEMME
QBBDRgAPg4eADwOJAA7DioAOQ8vADgQNgA2ED0ANRFFADQRTgAyElgAMRNjAC8TcQAtFI
AALBSRACoUowAoFLcAJhTSACYU7QAmFP0AJhP/ALEsAACmNAAAnDgAAJM6AACKOQAAg
DcAAHUyAABrLQAAYCgAAFYiAABMHAAQxcEADwUDAA5FBQANxQbADUUIQAZFSYAMhUs
ADAVMgAvFjkALhdBAC0XSwArGFUAKhhACgZbgAnGn0AJRqPACMaoQAIgrUAIBnQAB8Z7A
AgGf0AIBj/AK4xAACkOAAAmj0AAJE/AACIPgAAfjwAAHM4AABoMwAAXS4AAFMpAABIlwAAPx
4AADcaCQAYGREAMBkXAC4ZHQAsGiMAKxopACobLwApGzYAJxw/ACYcSAIHVIAJB5eACle
bAAhH3sAHx+NAB4fnwAch7MAGh7OABoe6gAaHvwAGx3/AKw2AACiPQAAMEEAAI9DAACGQ
wAAfEEAAHE9AABmOQAAWzQAFAvAABGKgAAPCUAADlhBQAsHg0AKR4UACceGgAlHiAAJ
R8mACQgLAajIDQAliE8ACAhRQAFIIAAHiNcAB0jagAbI3kAGiSLABgkngAWJLIAFSPMABQj6QA
VlvwAFSL/AKo6AACgQQAAl0UAAI5HAACFSAAAE0YAAHBDAABkPgAAWToAAE42AABEMQA
AOi0AADAoAgAnJAoAJCMRACIkFwAgJB0AHyQjAB4IKgAdJTEAHCY5ABsmQwAaJ04AGCdaA
BcoaAAWKHcAFCmJABMpnAARKLAADijKAA4n5wAQJ/sAECb/AKg+AAceRQAAlUoAAIxMAAC
DTAAAEUsAAG5IAABiRAAAV0AAAE08AABDOQAAOTQAAC8wAAAKKwYAHikOABwpFQAbKhs
AGiohABgqJwAXKy8AFis3ABUsQAAULEsAEy1YABEtZgAQLXUADi6HAAwumgALLa4ACS3GAA
gs4gAJK/cACir/AKZCAACcSQAak04AAItQAACCUQAAEFEEAG1OAAABhSgAAVkyAAEExDAABC
QAAANzwAACw3AAAhMwMAGTAKABYvEgAVMBgAFDAeABlwJQARMSwAEDE0AA4xPgANMk
kADDJVAAozYwAJM3IACDOEAAYzlwAEM6sAAZLBAAlx3wADMfIABDD/AKNHAACaTgAAklMA
AllWAACAVwAAdlcAAGxUAABgUQAAVU4AAExLAABARgAANEIAACg+AAAEogAAFTcHABA2
DgANNhUADDccAAs3lgAKNyOACTgyAAc4OwAGOEYABThSAAM5YAACOW8AADmBAAA5IAA
AOKgAADi+AAA32wAANvAAADX7AKFNAAcYVAAAKFgAAIhbAAB/XQAAdV0AAGtbAABfWAAA
VvYAAEHRAAA8TAAAL0gAACRFAAAaQgAAET8EAAK+CwAGPhMABD4ZAAM+IAACPicAAT8v
AAA/OAAAP0MAAD9PAAA/XQAAP2wAAD99AAA/kQAAPqUAAD27AAA81gAAO+8AADv7AJ5T
AACWWgAAjI8AAIZiAAB+ZAAAdWQAAGpiAABeYAAAUFSAAENXAAA3UwAAKIAAAB9NAAAW
SgAADEgDAARGCgAARhEAAEYXAABGHQAARiQAAEYsAABGNQAARj8AAEZMAABGWQAAR
mkAAEZ6AABFjQAARaIAAES4AABD0wAAQuwAAEH6AJxaAACUYQAAjWUAAIVpAAB9awAAD
GsAAGlqAABZZgAAS2EAAD5eAAAxWgAAJVcAABpVAAARUgAAB1EBAABQCAAATw4AAE8U
AABPGgAATyAAAE8oAABPMQAATjsAAE5IAABOVQAATmUAAE52AABNigAATJ8AAEu1AABK
0AAASesAAEj4AJIiAACSaAAAI20AAIRxAAB8cwAAcnMAAGNvAABUawAARGgAADdkAAAqYgA
AH18AABVdAAAKWwAAAVoAAABaBQAAWQsAAfKQAABZFgAAWBwAAFGkAABYLQAAWDcA
AFdDAABXUQAAV2AAAFZyAABWhQAAVZsAAFSxAABSzQAAUekAAFD4AJZqAACQcAAAIInU
AAIN4AAB7egAABHcAAfX1AABNcgAAPm8AADBBSAAAKagAAGGgAAA5nAAAEZQAAAGQAAAB

kAgAAYwcAAGMMAABjEgAAYxgAAGIfAABiJwAAYjEAAGI9AABhSwAAYVsAAGBtAABggAAAX
5YAAF6tAABdxwAAW+YAAFr3AJVzAACPeQAAiX4AAIKAAABzfzAAZH0AAFN6AABEeAAANXY
AACh0AAAaccgAAEXEAAAZwAAAAbwAAAG8AAABuAAAAbgIAAG4HAABuDAAAabhIAAG0ZAABt
IAAAbSoAAG02AABtRAAAbFQAAGxmAABregAAapAAAGmoAABowgAAZ+EAAGX1AJN8AACO
ggAAh4UAAHqFAABrhAAAWoIAAEqAAAA7fwAALX0AAB98AAAUewAACHoAAAB5AAAAeQAA
AHkAAAB5AAAAeAAAAHkAAAB5BQAAeQsAAHkSAAB5GQAAeilAAHkuAAB5PAAeUwAAHhe
AAB4cgAAd4kAAHahAAB1ugAAdNkAAHPwAJGGAACMigAAf4oAAHCKAABgiQAAT4gAAD+HA
AAwhgAAIoUAABaFAAAKhAAAAIQAAACDAAAAAgwAAAIMAAACDAAAAhAAAAIQAAACEAAAA
hQIAAIUJAACGEAAAhxkAAIckAACHMQAAhkEAAIZUAACFaQAAhYAAAIISZAACDsQAAGs4AAI
LpAJCOAAACEjwAAdZAAAGWPAABUjwAAQ44AADOOAAAljQAAGI0AAAuOAAAAjgAAAI4AAAC
NAAAAjQAAAI4AAACOOAAAAjwAAAI8AAACQAAAAkQAAAJIAAACSBgAAkw4AAJUYYAACVJQA
AITUAAJVIAACVXQAAIHQAAJSPAACtqAAAKslAAJLfAlE TAAB5IAAAapQAAFiUAABGIQAANpU
AACaVAAAZIQAADJYAAACWAAAAIwAAAJcAAACXAAAAIwAAAJgAAACZAAAAmgAAAJJoAAA
CcAAAAAnAAAAJ4AAACfAAAAoQIAAKMMAACkGAAApCgAAKU6AACIUAAApWcAAKWAAACIm
gAApLEAAKTLAH2YAABtmQAAXZkAAEuaAAA6mwAAKZWAAABqdAAANngAAAJ8AAACgAAAAo
AAAAKAAAAChAAAAogAAAKMAAACkAAAApQAAAKYAAACnAAAAqQAAAKoAAACsAAAArgA
AALAAAC0CwAAtHoAALYrAAC3QAAAUfGAAALhxAAC4jAAAUkQAALi4AHGcAABgnQAAT58AA
D2gAAAsogAAHKMAAA6IAAAApGAAAKgAAACpAAAAqgAAAKoAAACrAAAArAAAAK0AAACvAA
AAsAAAALIAAAC0AAAAtgAAALkAAAC8AAAAvwAAAMQAAADJAAAAzQoAAM4dAADQMAAA0
UgAANNhAADUegAA1ZIAANWmAP8AAAD/AAAA/wAAAP8AAAD/AAAYA/wASAP0AHQD6ACcA9
QAxAPAAOgDsAEMA6gBLAOcAUgDIAFgA4wBeAOIAYwDgAGgA3QBtANoAcwDWAHkA0gB/A
NEAhwDPAJAAzQCbAMsAqADJALcAxwDQAMUA7gDCAP8AwQD/ALsA/wC0AP8ArwD/AP8AA
AD/AAAA/wAAAP8AAAD4AAIA8wAOAO8AGQDqACMA5QAsAOAANgDcAD4A2QBGANUATQD
TAFMA0ABZAM0AXgDKAGMAxwBoAMUAbQDDAHMAwgB6AMAAggC/AlSvQCVALsAogC5AL
AAtwDGALUA5QCyAP4AsQD/AK8A/wCpAP8ApAD/AP8AAAD/AAAA/QAAAPIAADqAAAA4wAK
ANwAFQDWAB8A0gAnAM0AMQDJADkAxQBBAMIASAC+AE4AuWBUALkAWQC4AF4AtwBjALU
AaAC0AG4AswB1ALEfACvAIUArgCPAKwAnACqAKoAqAC8AKYA2QCjAPYAoQD/AJ8A/wCcA
P8AmAD/AP8AAAD7AAAA7gAAAOIAAADZAAAA0AAGAMgAEQDDABoAvwAjALsAKwC3ADQA
swA8ALAAQwCuAEkArABOAKsAUwCqAFkAqABeAKcAYwCmAGkApABvAKMAdgChAH8AnwC
JAJ0AIQCbAKMAMQC0AJYAzACTAO0AkGd/AJEA/wCPAP8AjAD/APkAAADtAAAA3gAAANEAA
ADHAAAAvgABALgADACzABYArwAeAKsAJgCnAC4ApAA2AKIAPQCgAEMAnwBJAJ0ATgCcAF
MAmgBYAJkAXQCYAGMAIgBpAJQAcQCTAHkAkQCDAl4AjwCMAJ0AiQCsAlcAwgCFAOIAGwD
7AIIA/wCAAP8AfgD/AOWAAADcAAAAzgAAAMIAAAC4AAAAAsQAAAKoABwCIABEAoAAaAJ0Alg
CaACkAlwAwAJUANwCTAD0AkQBDAI8ASACOAEOAjABSAIsAVwCJAF0AiABkAIYAawCEAHM
AggB9AIAAiQB9AJYAewCmAHkAuQB2ANYAdADzAHMA/wBxAP8AcQD/AN4AAADOAAAAwAA
AALYAAACtAAAApgAAAJ8AAwCZAAwAlQAVAJEAHQCNACQAigArAlgAMQCFADcAhAA9AIIA
QgCAAEcAfwBMAH0AUgB8AFgAegBeAHgAZQB2AG4AdAB4AHIAgwBvAJEAbQCgAGsAsgBoA
MsAZgDqAGQA/wBjAP8AYwD/ANIAAADDDAAAAtgAAAKwAAACkAAAAAnAAAAJUAAACPAAGig
ARAIUAGACBAB8AfgAIAHsALAB4ADIAdwA3AHUAPQBzAEIAcgBHAHAATABuAFIAbQBZAGsA
YABpAGkAZwByAGQAfgBiAlSAYACbAF4ArABbAMIAWgDhAFgA+QBXAP8AVgD/AMgAAAC6A
QAARwQAAKUEAACcAAAAIAAAAIwAAACFAAMAFwAMAHoAEwB2ABoAcgAhAG8AJwBsACwAa
gAyAGkANwBnADwAZQBCAGQARwBiAE0AYABUAF4AWwBcAGQAWgBuAFgAeQBWAICAVA

CWAFIApwBQALsATgDZAE0A8gBMAP8ASwD/AMEHAACzDgAAqRIAAJ8SAACWDgAAjQgAAI
UAAAB8AAAAdgAHAHAADgBrABYAZwAcAGQAlgBhACcAXwAtAF0AMgBbADcAWQA9AFgAQg
BWAEGAVQBPAFMAVwBRAf8ATwBpAE0AdQBLAIISQCSAEcAowBFALYAQwDQAEIA7ABBA
PwAQQD/ALsSAACuGAAApBwAAJocAACRGQAAiBUAAH8OAAAB2BgAAbQABAGcACQBhABE
AXQAXAFkAHQBxACMAVAAoAFIALQBQADIATwA4AE0APQBMAEQASgBLAEgAUgBGAFsAR
ABmAEMAcQBBAH8APwCPAD0AoAA8ALIAOgDKADkA5gA4APcANwD/ALYaAACqIAAAoCQAA
JcIAACNIwAAhB4AAHoZAABxEgAAZwsAAF4DAwBYAAsAVAASAFAAGQBNAB4ASgAjAEgAKQ
BHAC4ARQAzaEMAOQBCAD8AQABGAD8ATgA9AFgAOwBiADkAbgA4AHwANgCMADQAnQA
yALAAMQDHAC8A4gAuAPUALQD/ALlgAACnJwAAAnSsAAJQsAACKKgAAgCcAAHYiAABsHAAA
YhUAAfKOAABQCAUASgIMAEcAFABDABoAQQAfAD8BJAA9ASoAPAIvADoCNQA5AzsANwNC
ADUESwA0BVQAMgVfADAGbAAvB3oALQeKACsHnAApB68AKAfFACYH4QAIB/MAJQb/AK8mA
ACkLQAAMzEAAJEyAACIMQAAfi4AAHMPAABPlwAAXx4AAFUYAABLEgAAQwsGAD0IDQA7Bx
QAOAcaADYIIAA1CCUAMwkrADEJMQAwCjcALgo/AC0LRwArDFEAKQxdACgNagAmDXkAJA6
KACIONAAhDq8AHw3HAB4N4wAeDfUAHhg3/AKwrAACiMgAAmDYAAI83AACGNwAAfDQAAHEv
AABmKgAAWYUAAFEfAABIGgAAPxQBADcQCAAYDQ8ALw4WAC0OGwAsDiEAKg8nACkQLQA
oEDMAJhE7ACURRAAkEk8AIhNaACETaAAfFHcAHhSIABwUmgAaFK4AGBTGABgT4wAYE/gA
GRP/AKowAACgNwAAIjsAAI08AACEPAAAeJoAAG81AABkMQAAWswAAE8mAABFIQAAPBwA
ADIXAwAsFAsAKBQSACYUGAAIFB4AlxQjACIVKgAhFjAAIBY4AB8XQgAeF0wAHBhYABsZZQA
ZGXQAGBmGABYZmQAVGawAExnEABIY4gATGPcAExj/AKg0AACeOwAAIT8AAIxBAACCQAA
AeD8AAG07AABiNgAAVzIAAE0tAABCKAAAOSMAADAFAAAnGwcAIhkOACAZFQAeGRoAHRkg
ABwaJwAbGy4AGhs2ABkcPwAYHEoAFx1WABUdYwAUHnMAEH6EABEelwAOHqsADB7CAAsd
3gANHfUADRz/AYK4AACcPwAAk0MAAIpFAACBRQAAd0QAAGxAAABgPAAAVTgAAEsZaABB
LwAAOCsAAC4mAAAlIgQAHR8LABoeEgAZHxgAGB8eABcfJAAWICsAFCAzABMhPQASIUCAE
SJTAAsiYQANI3AADCOCAAojIAAJI6gABY0+AAYi2gAHIfEACCD/AKQ9AACaQwAAkkcAAIIJAA
B/SgAAdkkAAGtGAABfQQAaVD4AAEo6AABANgAANzIAAC0uAAAJKgAAGSYIABQkDgATJBUA
EiUCABEIlgAQJSkADiYxAA0mOgAMJ0UACidRAAkoXwAHKG4ABih/AAQokgADKKUAASi7AAEn
1gABJu0AAiX8AKJBAACYSAaAkEwAAIdOAAB+TwAADE4AAGpLAABdRwAAU0QAAEpBAABA
PgAANJoAACo1AAAgMQAAFi0FAA8rCwAMKxMACysZAAorIAAJLCCACCwvAAYsOAAFLUIABC
1OAAluXAABLmsAAC58AAAUjwAALaMAAC24AAAs0gAAK+wAACr5AKBGAAACWTQAAjIEAAIZT
AAB9VAAAc1QAAGISAABdTgAAU0sAAEIJAAA+RAAAMkAAACY7AAAcOAAAEzUCAAoyCgAG
MhEABDIXAAMyHgACMiUAADMsaAAzNQAAm0AAADNMAAA0WQAANGgAADR5AAA0jAAAM
6AAADK2AAAx0AAAMOoAAC/5AJ1LAACVUGAAjVYAAIRZAAB8WgAAcloAAGhYAABcVgAAU1
MAAEZPAAA5SgAALUYAACJCAAAYPwAADJwCAAY6CQAAOg8AADoVAAA6GwAAOIlAADoqA
AA6MgAAOj0AADpJAAA6VgAAOmUAADp2AAA6iQAAOZ4AADizAAA3zQAANugAADX4AJtSAA
CTWAAAi10AAINgAAB7YQAAcmEAAGhgAABcXQAATIkAAEFUAAA0UAAAKE0AAB1JAAATRW
AACUQBAAFDCAAAQg0AAEITAABCGQAAQR8AAEEnAABBLwAAQTkAAEFFAABBUwAAQWI
AAEFzAABAhgAAQJsAAD+wAAA+ywAAPOcAADz2AJhZAACRXwAAimMAAIJnAAB6aQAAcWk
AAGdnAABXYwAASF4AADtaAAAUvWAAIIQAABhRAAANTwAABE0AAABMBgAASwsAAEsQAA
BKFGAAShwAAEojAABKLAAASTYAAEIBAABJTwaASV4AAEHvAABlgwAAR5gAAEauAABFyAA
AROUAAEL2AJZgAACPZgAAiGsAAIFuAAB5cQAAChAAAGftAABRaQAAQmUAADVhAAAOXgA
AHFwAABJaAAAHWAAAAFYAAABWAwAAVQcAAAFQMAABUEgAAUxgAAFMfAABTJwAAUzEA
AFI9AABSSgAAUloAAFFrAABRfgAAUJQAAE+qAABOxAAATOMAAEv2AJRpAACObwAAh3MAA

IF2AAB4eAAAaXUAAFlyAABKbwAAO2wAAC1pAAAhZgAAFWQAAApjAAAAAYQAAAGAAAAABgA
AAAXwQAAF4IAABeDQAAAXhMAAF0aAABdlgAAXSwAAF03AABcRQAAXFQAAfTmAABbeQAA
Wo8AAFmmAABXvwAAVt8AAFX1AJJxAACMdwAAhnwAAH9+AABxfAAAYXoAAFF3AABBdQAA
MnIAACVwAAAZbwAADW0AAANsAAAAawAAAGoAAABqAAAAagAAAGkDAABpCAAAaQ0AAG
kUAABoGwAAaCUAAGgwAABoPgAAZ04AAGdfAABmcwAAZYoAAGShAABjugAAYdoAAGDxAJ
F7AACLgAAAhYQAAHeDAABogQAAV4AAAEd+AAA4fAAAKXoAABx5AAAQdwAABXcAAAB1AA
AAdQAAAHUAAAB0AAAAAdAAAAHQAAAB0AQAAAdAcAAHQNAAB0FAAAAdR0AAHQoAAB0NQAA
AdEUAAHNXAABzawAAcolAAHGaAABwsgAABtEAAG3tAl+EAACKiAAAfYkAAG6IAABdhwaATI
UAADyEAAAsgwAAH4IAABKBAAAGgQAAAIAAAAB/AAAAfwAAAH8AAAB/AAAAfwAAAH8AAAC
AAAAAgAAAAIEEAACBCwAAAgRQAAlfAACCLAAAgTsAAIFOAACBYgAAgHkAAH+SAAB+qgAAf
scAAHzlAl6MAACBjQAAC40AAGKNAABRjAAAQIsAADCLAAAIigAAFIoAAAeKAAAAigAAAIoAAA
CJAAAAiQAAAIoAAACKAAAAigAAAIIsAAACMAAAAJAAAAI0AAACOAQAajwkAAJATAACQIAAA
kC8AAJBBAACQVwAAj24AAI+IAACOoQAajbsAAI3aAIWRAAB3kgAAZ5IAAFWSAABEkgAAM5I
AACOSAAAWkgAACJMAAACTAAAAkwAAAJMAAACTAAAAIAAAAJQAAACVAAAAIQAAAJYAA
ACXAAAAmAAAAJoAACbAAAAAnAAAAJ4HAACfEwAAoCMAAKA0AACgSQAAoGAAAKB5AAC
glAAAn64AAJ7JAHqWAABrlwAAWpcAAEiXAAA3mAAAJpkAABiZAAAJmgAAAJwAAACcAAAAAn
QAAAJwAAACdAAAAngAAAJ8AAACgAAAAoAAAAKEAAACjAAAApAAAAKYAAACnAAAAqgAA
AKwAAACvBgAAsRQAALeMAACyOgAAsIIAALJrAACzhQAAs54AALKzAG6bAABemwAATJwAA
DqeAAApnwAAGaEAAAuIAAAAwAAAKQAAACmAAAApWAAAKYAAACnAAAAqAAAAKoAAACr
AAAArAAAAK4AAACwAAAAsgAAALQAAAC2AAAAuQAAALwAAADAAAAAxQUAAMgXAADJKw
AAyklAAMxbAADNdAAAzY4AAM2kAP8AAAD/AAAA/wAAAP8AAAD/AAMA/QAOAPoAGQD2AC
MA8AAAtAOwANgDoAD8A5QBHAOIATgDgAFQA3gBZANoAXwDWAGMA0wBoANAAbgDPAHQAA
zQB6AMwAggDKAIsAyACWAMYAowDEALIAwQDIAMAA6QC+AP8AvAD/ALYA/wCvAP8AqgD/A
P8AAAD/AAAA/wAAAPsAAAD1AAAA7wAKAOoAFgDkAB8A4AAoANsAMgDWADoA0gBCAM8A
SQDKAE8AxwBUAMQAWQDCAF4AwQBjAMAAaQC+AG4AvQB1ALsAfQC6AIYAUAACQALYAnA
C0AKsAsgC/ALAA3gCuAPoAqwD/AKkA/wCjAP8AnwD/AP8AAAD/AAAA9wAAAO0AAADIAAAA3
QAHANUAEQDQABsAzAAkAMcALQDDADUAvQA9ALoARAC3AEoAtQBPALQAVACzAFkAsQB
eALAAYwCvAGkArgBwAKwAdwCqAIAAqQCKAKcAlgCIAKQAowC2AKAA0QCdAPEAmwD/AJoA
/wCXAP8AkWd/AP8AAAD2AAAA6AAAAAN0AAADSAAAAyQACAMIADQC9ABcAuQAfALQAJwCw
ADAARQA3AKsAPgCpAEQApwBKAKYATwCIAFQAowBZAKIAXgChAGQAnwBqAJ4AcQCCaHoA
mgCEAJgAkACWAJ4AkWCUAJAAxQCOAOcAjAD/AlSA/wCKAP8AhwD/APUAAADmAAAA2AAAA
MsAAADBAAAAuAAAAALACQCtABIAqAAbAKUAlwCiACoAnwAyAJ0AOACbAD4AmQBEAJgASQ
CWAE4AIQBTAJQAWACSAF4AkQBkAI8AawCNAHQAIwB+AlgAigCGAJcAhACnAIEAuWB/ANsA
fQD4AHwA/wB7AP8AeQD/AOUAAADUAAAAxwAAALwAAACzAAAAqWAAAKUABQCfAA0AmwA
WAJcAHgCUACUAKQAsAI8AMgCNADgAiW+A+AlOAwCIAEgAhwBNAlUAUgCDAFgAggBeAlAA
ZgB+AG4AfAB4AHoAgwB3AJEAdQChAHMAswBwAM4AbgDvAG0A/wBsAP8AagD/ANcAAADHA
AAAugAAALAAACoAAAAoAAAAJkAAQCUAAkAjwASAlSAGQCHACAAhAAmAlIALQB/ADMAfQ
A4AHwAPQB6AEIAeQBHAHcATQB2AFMAAdABZAHIAAYABwAGkAbgByAGwAfgBpAlSAZwCbAG
QArABiAMQAYADmAF8A/QBdAP8AXQD/AMsAAAC9AAAAAsQAAAKgAAACeAAAAIwAAAJAAAA
CJAAUUhAANAHA8AFQB7ABwAdwAiAHUAKABYAC0AcAAzAG8AOABtAD0AawBCAGoARwBoA
E0AZgBUAGUAWwBjAGMAYABtAF4AeABcAlYAWgCVAFcApwBVALwAVADbAFIA9gBRAP8AU
AD/AMMAAAC1AAAAqglAAKABAACXAAAAJwAAAlcAAACAAAEAEQAIAHQAEQBwABcAbAAAd

AGkAlwBmACgAZAAuAGIAMwBgADgAXwA9AF0AQgBbAEgAWgBPAFgAVgBWAf4AVABoAFI
AdABQAIEATgCQAEwAogBKALUASADRAEYA7wBFAP8ARQD/ALsGAACvDAAApA8AAJsPAA
CRCwAAiAUAAH8AAAB3AAAAcAAEAGoADABIABMAYQAYAF0AHgBaACMAWAApAFYALgBV
ADMAUwA4AFIAPQBQAEMATgBKAEwAUQBLAFoASQBkAEcAbwBFAH0AQwCMAEEAnQA/AL
AAPgDJADwA6AA8APoAOwD/ALYQAACqFwAAoBoAAJYaAACNFgAAgxIAAHoLAABxAwAAaA
AAAGEABwBcAA4AVwAUAFQAGgBQAB8ATgAkAEwAKQBKAC4ASQAzAEcAOABFAD4ARAB
FAEIATQBAAFYAPgBgAD0AawA7AHkAOQCIADcAmgA2AKwANADEADMA4QAYAPUAMQD/AL
EZAACmHwAAncIAAJMiAACJIAAAfxsAAHUWAAABsEAAAYggAAfKAAQBTAAkATgAQAEoAFQ
BHABsARAAfAEIAJQBAACoAPwAvAD0ANAA8ADoAOgBBADgASQA3AFIANQBcADMAaAAxA
HYALwCFAC4AlwAsAKkAKwC/ACkA2wApAPAAKAD+AK4fAACjJQAAmSkAAJApAACGJwAAfC
QAAHIfAABoGQAAXhMAAFQMAABMBQMARQAKAEeAEQA+ABYAOwAbADkAIAA3ACUANgA
qADQAMAAyADYAMQA9AC8ARQAtAE4ALABZACoAZQAoAHMAJwGDACUBIQAjAagAlgC8AC
EA1gAfAO8AHwD7AKsAACgKwAAIy8AAI0vAACELgAAeisAAG8mAABIIQAAWHSAAFEVAABH
DwAAPwKFADKECwA1AhEAMwEXADACHAAuAiEALQMmACsDLAAqAzIAKAQ5ACcFQgAlBksA
lwZWACIHwAgCHEAHwiCAB0IIAAbCKcAGge7ABgH1gAXB+0AFwf6AKgqAACeMAAAITQAAIs
1AACCNAAAdzEAAG0sAABiKAAAVyMAAE0dAABEGAAAOxIAADMNBQAsCQsAKggSACglFwA
mCR0AJAKiACMJKAhCi4AIAo2AB8LPgAdDEgAHAXUABoNYQAYDnAAFW6BABUOkwATDqcA
Eg69ABAN2AARDe4AEQz8AKYuAACcNQAAkzgAAIo6AACAOQAAdjcAAGsyAABgLgAAVSkAA
EskAABBHwAAOBoAAC8VAAAnEQYAlg4MAB8OEwAdDhgAHA4eABsPJAAaECsAGREyABgR
OwAWEkYAFRJSABMTXwASE24AEBR/AA4UkgANFKUACxS6AAKT1QAKE+8ACxL+AKQzAAC
aOQAAkT0AAIg+AAB/PgAAAdDwAAGk4AABeNAAAVC8AAEkqAAA/JgAANiEAAC0dAAAIGQEAH
RUIABgUDgAXFBUAfHqBAbUVIQAUFsGAEYwABIWOQAQF0MADxdPAA0YXQAMGWwAChI
8AAgajwAHGaMABRm3AAQY0QAEGoABRf7AKI3AACYPQAAkEEAAIdDAAB9QwAAc0EAAGg
9AABcOQAAUjUAAEGxAAA/LQAANSKAAcWIAAAkIQAAGx0FABQaCwARGhIAEBoYAA4aHwAN
GyYADBsuAAscNgAKHEEACR1NAAceWgAGHmkABB96AAIfjAABHqAAAB61AAAdzgAAHOgAA
Bv3AKA7AACXQgAAjkUAAIVHAAB8RwAAckYAAGdDAABbPwAAUTsAAEc4AAA+NAAANTEAA
CwtAAAIKAAAGCQCAA8hCQALIBAACiAXAAggHQAHisQABiEsAAUiNAAElj8AAiNLAAEjWAAAI
2cAACR3AAAkigAAI54AAcOyAAAIyWAAIecAACD2AJ5AAACVRgAAjUoAAIRMAAB7TQAACUsA
AGZJAABaRQAAUUIAAEc/AAA+PAAANDgAACkzAAAElwAAFCsBAAsOCAAGJw4AAycVAAInG
wABJyIAACcqAAAOmgAAKDwAAChIAAApVQAACKWQAACI1AAApiAAAKZwAACiwAAAnyQAAJu
QAACX2AJxFAACTSwAAi08AAIJRAAB6UgAAcFEAAGZPAABaTAAAUekAAEdHAAA8QgAAMD
4AACQ5AAAaNQAAETIBAAcvCAABLg0AAC4TAAAUgQAALiAAAC4nAAAUMAAALjoAAC9GAA
AvUwAAL2IAAC9yAAAvhQAALpoAAC2uAAAsxwAAK+MAACr0AJpKAACSUAAAIiQAAIFXAAB4
WAAAb1gAAGVWAABaUwAAUVEAAERMAAA3RwAAK0MAACBAAAWPAAADDkAAAM3BwA
ANgwAADYRAAA1FwAANR4AADUIAAA1LQAANTcAADVDAAA1UAAANV8AADVwAAA1ggAAN
JcAAD0sAAAYxQAAMeIAADD0AJdQAACQVgAAiFoAAIBdAAB4XwAAb18AAGVeAABaWwAAT
FYAAD9SAAAYTQAAJkoAABtGAAARQwAAB0EAAAA/BQAAPwoAAD4PAAA9FAAAPRoAAD0iA
AA9KgAAPTQAAD0/AAA9TQAAPFwAADxsAAA8fwAAO5QAADqqAAA5wwAAOOEAADb0AJVX
AACOXQAAh2EAAH9IAAB3ZgAAb2cAAGRIAABVYAAARlwAADIXAAAsVAAAI FEAAABVOAAALS
wAAAUkAAABIAwAARwgAAEYMAABGEgAARRcAAEUeAABFJwAARTAAAEQ8AABESQAARFg
AAERpAABDfAAAQpEAAEGnAABAwAAAP98AAD70AJNfAACMZQAAhmkAAH9sAAB3bgAAbm
4AAF5qAABOZgAAP2IAADJeAAAIWwAAGVgAAA5WAAAEVAAAAFMAAABSAAAAUQQAFAJ

AABPDQAATxMAAE4aAABOIgAATiwAAE43AABNRAAATVMAAExkAABMdwAAS40AAEqkAABl
vQAAR9wAAEbZAJFnAACLBQAahXEAAH51AAB2dgAAZ3MAAFZvAABHbAAAOGgAACplAAAE
YwAAE2EAAAdfAAAAXQAAAFwAAABbAAAAWwAAAFoFAABaCQAAWQ4AAFgVAABYHQAAW
CcAAFGyAABXPwAAV04AAFZfAABWcgAAVYgAAFSgAABSuAAAUdgAAFDxAJBwAAACKdgAAh
HoAAH18AABuegAAXncAAE51AAA/cgAAMG8AACJtAAAWawAACmoAABBoAAAAZwAAAGYA
AABmAAAAZQAAAGUAAABkBAAAZAkAAGQQAABjFgAAYyAAAGMrAABjOAAAYkcAAGJZAAB
hbAAAYIIAAF+aAABeswAAXNIAAFvuAl56AACJfwAAg4IAAHWBAABIfwAAVH0AAER7AAA1eQA
AJncAABl1AAANDAAAAAXMAAABxAAAAcQAAHEAAABwAAAAcAAAAHAAABwAAAAcAIAAH
AIAABwEAAAbxgAAG8jAABvMAAAbz8AAG5RAABuZQAAbXsAAGyTAABqrAAAacoAAGjpAl6D
AAClhwAAe4cAAGuFAABahAAASYIAADmBAAAqgAAAHH8AAA5+AAADfQAAAH0AAAB7AAAA
ewAAAHsAAAB7AAAAewAAAHsAAAB7AAAAfAAAAHwAAAB8BwAAfQ8AAH0aAAB9JgAAfDUA
AHxHAAB8WwAAe3IAAHqLAAB5pAAAE8AAHjgAlYLAAB/iwAAcIsAAGCKAABOIgAAPYkAAC2I
AAAfhwAAEYcAAASGAAAAhgAAAlcAAACFAAAAhQAAAIYAAACGAAAAhgAAAIYAAACHAAAAi
AAAAIkAAACJAAAAigQAAlS0AACLGwAAIyoAAIs7AACLUAAaimcAAAlqAAACJmwAAibQAAljSAI
OPAAB0kAAAZJAAAF0QAABBJwAAMI8AACCCQAAATjwAABZAAAACQAAAAkAAAAI8AAACPA
AAAKAAAAJAAACRAAAAKQAAAJIAAACTAAAAIAAAAJUAAACWAAAAmAAAAJkCAACbDgAA
mx0AAJsuAACbQgAAm1oAAJtzAACajwAAmqkAAJnCAHiUAABolAAAV5UAAEWVAAA0IQAAI5Y
AABWWAAAGlwAAAJgAAACZAAAAmQAAAJgAAACZAAAAmgAAAJsAAACcAAAAAnAAAAJ0AA
ACfAAAAoAAAAKIAAACjAAAApQAAAKcAAACqAAAArA8AAKwhAACTNAAArUwAAK1IAACtfgAA
rZgAAK2vAGyZAABbmQAASpoAADibAAAmnAAAFp4AAAIifAAAAoAAAAKEAAACiAAAAowAAA
KMAAACkAAAApQAAAKYAAACnAAAAqAAAAKkAAACrAAAArQAAAK4AAACwAAAAswAAALY
AAAC6AAAAvWAAAMESAADDDJQAAXDsAAMVVAADFbwAAxokAAMagAP8AAAD/AAAA/wAAAP
8AAAD/AAAA+gALAPUAFgDxAB8A7AApAOcAMgDjADsA4ABDAN0ASQDYAE8A1ABVANAAW
gDOAF8AzABkAMsAaQDJAG8AyAB1AMYAfQDFAIYAwWCRAMEAnQC+AKwAvADBALoA4gC5
AP0AtwD/ALEA/wCpAP8ApAD/AP8AAAD/AAAA/wAAAPgAAADxAAAA6gAHAOMAEgDfABwA2g
AlANQALgDPADYAYgA+AMYARADCAEoAwABQAL4AVQC9AFoAvABfAlLoAZAC5AGoAuABwA
LYAdwC0AIAAswCLALEAlwCvAKYArAC4AKsA1QC0APUApGd/AKMA/wCeAP8AmgD/AP8AAAD
+AAAA8gAAAOkAAADgAAAA1gADAM8ADQDJABcAxQAgAMAAKQC6ADEAtgA4ALQAPwCyAE
UAsABKAK8ATwCuAFQArABZAKsAXwCqAGQAqQBrAKcAcgCIAHsApACFAKIAkQCgAJ8AnQC
wAJ0AyQCYAOwAlgD/AJUA/wCRAP8AjgD/AP4AAADvAAAA4wAAANYAAADLAAAAwwAAALwA
CQC3ABMAsgAcAK4AJACqACwAqAAZAKYAOQCkAD8AogBFAKEASgCfAE8AngBUAJ0AWQC
bAF8AmgBIAJgAbACXAHUAlQB/AJIAigCQAJgAjQCpAlSavgCJAOEAhwD9AlUA/wCEAP8AgQD/
AO8AAADfAAAA0QAAAMUAAAC7AAAAswAAAK0ABQCnAA4AowAXAJ8AHwCcACYAmgAtAJc
ANACVADoAIAA/AJIARACRAEkAjwBOAI4AUwCMAFkAiWbFAlkAZgCHAG8AhQB4AIMAhACAAJ
IAfgCiAHwAtQB5ANMAdwD0AHYA/wB1AP8AcwD/AN4AAAD0AAAAwgAAALYAAACuAAAApgA
AAJ8AAgCaAAoAlgATAJIAGgCPACEAjAAoAlkALgCHADQAQhQA5AIQAPgCCAEMAgQBIAH8AT
gB9AFMAfABZAHoAYQB4AGkAdgBzAHQAfgByAlwAbwCbAG0ArgBqAMcAaADqAGYA/wBIAP8
AZQD/ANAAAADCAAAAAtQAAAKsAAACjAAAAmwAAAJQAAACPAAYaigAOAIUAFgCBAB0AfgAi
AHsAKQB5AC4AdwA0AHYA0QB0AD4AcwBDAHEASABvAE4AbgBUAGwAWwBqAGMAaABtA
GYAeABjAIYAYQCVAF4ApwBcAL4AWgDgAFkA+gBYAP8AVwD/AMUAAAC4AAAArAAAAKMAA
ACZAAAAkgAAAl0AAACEAAMAFgAKAHkAEQB1ABgAcQAeAG4AJABsACkAagAuAGgAMwBnA
DgAZQA9AGQAQgBiAEgAYABPAF4AVgBcAF4AWgBoAFgAcwBWAIAAVACQAFEAogBPALYA

TQDVAEwA8wBLAP8ASgD/AL0AAACwAAAApQAAAJwAAACTAAAAigAAAIEAAAB6AAAAAdAAG
AG8ADQBqABMAZgAZAGIAHwBgACQAXgApAFwALgBaADMAWAA4AFcAPQBVAEMAVABJAF
IAUQBQAFkATgBjAEwAbgBKAHsARwCLAEUAnABEALAAQgDLAEEA6wA/AP8APwD/ALYFAA
CqCgAAoA0AAJYMAACNCAAAhAIAAHoAAABYAAAAawACAGUACQBgABAawwAVAFcAGgBU
AB8AUgAIAFEAKQBPAC4ATQAZAESAOABKAD4ASABFAEYATABEAFQAQgBeAEAAagA/AHc
APQCGADsAmAA5AKsAOADDADcA4wA2APkANQD/ALEPAACmFQAAMxgAAJIXAACIEwAAfw
4AAHUIAABsAAAAyWAAAFwABQBWAAAsAUgARAE4AFgBKABsASAAgAEYAJQBEACoAQgAu
AEEANAA/ADkAPgBAADwASAA6FAAOABaADYAZgA1AHMAMwCDADEAIAAvAKcALgC9AC0
A2wAsAPMAKwD/AK0XAACiHQAAMCAAAI8fAACFHQAaexkAAHETAABnDAAAXQUAAFUAAA
BOAAYASQAMAEUEgBBABcAPgAbADwAIAA7ACUAOQAqADcALwA2ADUANA8ADIAQwAw
AEwALgBXAC0AYgArAHAAKQB/ACgAkQAmAKQAJQC4ACQA0wAjAO4AlgD8AKkeAACflwAAIS
cAAlwnAACCCJQAaCEAAG4cAABjFgAAWRAAAFJAABHAgEAQQAIAD0ADQA5ABMANgAXA
DMAHAAxACEMAAAMAC4AKwAsADEAKgA4ACkAQAAAnAEkAJgBTACQAXwAiAG0AIQB9AB8A
jgAeAKEAHAC1ABsAzgAbAOgAGgD5AKcjAACcKQAakywAAIkAACALAAAdSgAAGsjAABgHgA
AVhkAAEWTAABDDQAAOwcDADUCCAAwAA4ALQATACoAGAAoAB0AJwAiACUAJwAkAC0Alg
A0ACEAPAAfAEUAHgBQABwBXQAaAWsAGQJ7ABcCjAAWAp8AFQGzABQAYwASAEYAEQD3
AKQoAACaLgAAKTEAAIgyAAB+MgAAcy8AAGkqAABeJQAAVCAAAEobAABAFQAAOBAAAC8L
AwAoBwkAJQQOACICEwAgAxkAHgMeAB0ElwAcBCoAGgUxABkFOQAYBkMAFgZOABUHWwA
TCGkAEQh5ABAliwAOCJ8ADAizAAsHywAKB+YACQf1AKItAACYMwAAjzYAAIY3AAB8NwAAcj
UAAGcwAABcKwAAUiYAAEGiAAA+HQAANRgAACwUAAAKDwQAHQsJABoJDgAYCRQAFgkaA
BUKIAAUCiYAEwstABELNgAQDEAADgxLAAwNWQALDmcACQ93AAgQigAGEJ0ABQ+xAAQOy
QACDeQAawz2AKAyAACWOOAAjjsAAIQ8AAB7PAAAcTkAAGY2AABaMQAAUC0AAEY0AAA9J
AAANB8AACsbAAAJGAAAGxQEABMQCQAODg4ADhAVAA0QHAAMESMACxEqAAoSMwAIEj0
ABxNJAAYUVgAEFGUAAxV1AAEVhwAAFZsAABSvAAAUxgAAE+MAABLzAJ42AACVPAAAJD8
AAINAAAB6QAAAcD4AAGU7AABZNwAATzMAAEUvAAA8KwAAMycAACojAAAIIAAGxwAABE
YBgAKFg0ACBYUAAcWGgAGFyEABRcoAAMyMQACGDsAARIHAAAZVAAAGmIAABpzAAAah
QAAGpkAABmtAAAYxAAAGOAAABbzAJw6AACTQAAAI0MAAIJFAAB4RQAAb0MAAGRAAABY
PQAATjkAAEU2AAA8MgAAMy8AACssAAAhJwAAFyIAAA0fBgAGHAWAAhwSAAAcGQAAHR8A
AB0nAAAdLwAAHjkAAB5FAAAfUgAAH2AAAB9xAAAFgwAAH5cAAB6sAAAdwwAAHN8AABvzAJ
o/AACSRAAAIUgAAIFKAAB3SgAAbkkAAGNGAABYQwAATkAAAEU9AAA8OgAAMzcAACcxAA
AcLQAAEyKAAAKlBgABJAsAACMQAAAJFwAAIx0AACMIAAAjLQAAIzcAACRCAAkTwAAJF4AA
CRuAAAKgQAAJJUAACOqAAAIwQAAId4AACDxAJhEAACQSQAAIE0AAH9PAAB2UAAAbU8AA
GNNAABXSgAATkcAAEVFAAA7QAAALjwAACM3AAAYMwAADi8AAQsBQAAKwoAACoOAAA
pFQAAKRsAACKjAAaqKwAAKjQAACpAAAAqTQAAKIsAACpsAAaqfgAAKZMAACioAAAnwAAAJ
t0AACXxAJZJAACOTgAAhIIAAH5VAAB1VgAAbFUAAGJUAABYUQAAT08AAEJKAAA1RQAAKU
EAAB49AAATOGAACTYAAAA0BAAAMwKAADINAAAXEgAAMBKAADAgAAAwKAAAMDIAADE9A
AAxSgAAMFKAADBpAAAwfAAAL5EAAC6mAAAtvQAALNwAACvxAJRPAACNVAAAhVgAAH1bA
AB1XQAAbFwAAGJbAABYWQAASIQAAD1PAAAwSwAAI0cAABhDAAAOQAAABD4AAAA8AwA
AOwcAADoLAAA5EAAAOBYAADgdAAA4JQAAOC8AADg6AAA4RwAAOFUAADdmAAA3eQAAN
o4AADWkAAA0uwAAMtsAADHxAJJWAACLWwAAhGAAAHxJAAB1ZAAAbGQAAGJJAABSXgAA
RFkAADZVAAApUQAAHU4AABNLAAAISAAAAEYAAABEAQAAQwUAAEIJAABBDQAAQBMAA
EAaAABAlgAAQCsaAEA2AABAQwAAP1IAD9iAAA+dQAAPosAAD2hAAA7uQAAOtGAADnxAJ

FdAACKYwAAg2cAAHxqAAB1bAAAAa2wAAFtoAABMYwAAPV8AAC9bAAAiWAAAF1UAAAtTAA
ABUAAAAE8AAABOAAAAATQEAEwFAABLCgAASg4AAEkVAABJHgAASScAAEKyAABIPgAAS
E0AAEHheAABHcQAARocAAEWWeAABEtgAAQtQAAEHwAl9mAACJawAAgnAAAHxzAAB0dAAAZ
HAAAFRtAABEAQAANWUAACdiAAAbXwAAEF0AAARbAAAAAWQAAAFkAAABYAAAAVwAAAFY
BAABVBQAAVAoAAFQQAABTGAAAUYIAAFMsAABTOQAAAUkgAAFFZAABRBAAAUIIAEE+ZAA
BOsQAATNAAAEvuAl1vAACIdAAAgnGAAHT6AABseAAAw3UAAEtyAAA8bwAALWwAAB9qAAA
TaAAB2YAAABKAAAAAYwAAAGIAAABiAAAAAYQAAAGAABBgAAAAAXwUA AF8LAABeEgAAXh
wAAF4mAABeMwAAXUEAAF1SAABcZgAAW3wAAFqUAABZRAAAV8oAAFbqAlx4ACHfQAAgY
AAAHN/AABjfQA AUxoAAEF4AAaydgAAI3MAABZyAAAKcAAAAG8AAABuAAAabQAAAG0AAB
sAAAabAAAAGwAAABsAAAAawAAAGsEAABrCgAAaxMAAGoeAABqKwAAajKAAGIKaabPxA
AahQAAGeNAABmpgAAZMIAGPkAlYCAACGHQAaeIUAGMDAABXggAAR4AADd+AAAnfQ
AAGxsAAAt6AAAAegAAAHkAAAB3AAAAAdwAAAHcAAAB3AAAAAdwAAAHcAAAB3AAAAAdwAAA
HcAAAB4AgAAeAoAAHgVAAB4IQAAeC8AAHdB AAB3VQAAdmsAAHWDAAB1nQAAC7cAAHLA
AlqJAAB9igAAbokAAF2IAABLhwAAO4YA ACqFAAAchAAADYQAAACDAAAAgwAAAIMAAACBA
AAAgQAAAIIAACC AAAaggAAAI IAACDAAAAgwAAAI QAAACF AA AhQAA AIY JAACHFgAAhi
UAAIY2AACGSgAAhmEAAIV5AACFI AA AhK0AA IPKAI COA B yj gAAY o4AAFCNAAA/j QAAL Y0AA
B6MAAAQjAAAAAYwAAACMAAAAjAAAAIsAAACL AAAAjAAAAlwAAACNAAAAjQAAAI4AAACPAA
AAKAAAAJEAAACSAAAakwAAA JQA AACWCQA AlhgAAJcpAACXPAAAILQA AJZ uAACViAAAlaIA
AJ S8AHa SA ABmkga AVZIA AEKTAA AxkwAAIJQAABKUAAADI AAAAJ UA AACWA AAAIg AAAJUAA
ACWAAAA Ig AAAJC AAACYAAAA mQAAA JkAAAC bAAAA nAAAA J0AAAC fAAAA oQAA AKMAAAC
mAAAA pwoAA KgcAACo LwAA qEUAA KheAACoeAAA qJMAA KitAGmXAABZI wAAR5gAADaYAAA
kmgaAE5sAAAWcAAA AnQAAAJ4AAACfAAAAoAAAAJ8AAACgAAAAoQAAAKIAAACjAAAAPAA
AAKUAAACmAAAAqAAAAKoAAACsAAAArgAAALEAAAC1AAAAuQAAALwMAAC9IAAAvTYAAL
5PAAC/aQA Av4MAAL+bAAAAAAAAAAAAAAAAAAAAAAAAABA WQFBggJCgsNDg8REhMU FhcYG
hscHR8glSlkJSYoKSorLS4vMDIzNDY3ODk7PD0+QEFCREVGR0IKS01OT1BSU1RVV1hZW1xd
Xm BhYmNI Zmdpamtsbm9wcXN0dXd4eXp8fX6AgYKDhYa HilqLjl6PkJGTIJWWmJmanJ2en6Gio6
Smp6iqq6ytr7Cs7S1tri5uru9vr/BwsPE xsflycvMzc/Q0dLU1dbX2drb3d7f4OLj5Obn6Onr7O3u8PH
y9PX29/n6+/z+//////////////////////////8AAAAAAAAAAAAAAAAAAAAAAAAQM EBQYICQ
oLDQ4PERITFBYXGBobHB0fICEijCUmKCkqKy0uLzAyMzQ2Nzg5Ozw9PkBBQkRFRkdJSktNTk
9QUINU VvdYWVtcXV5gYWJjZWZnaWprbG5vcHFzdHV3eHI6fh1 +gIGCG4WGh4iKi4yOj5CRk5S
VlpiZmpydnp+hoqOkpqeoqqusra+w sbO0tba4ubq7vb6/wcLDxMbHyMnLzM3P0NH S1NXW19na29
3e3+Di4+Tm5+jp6+z t7vDx8v T19vf5+v v8/v//////////////////////////////////////AAAA
AAAAAAAAEDBAUGCAkKCw0ODxESE xQWFxgaGxwdHyAhliQlJigPKistLi8wMjm0NJc4OTs8PT5
AQ UJERUZHSUpLTU5PUFJT VFVXFibXF1eYGF iY2VmZ2lqa2xub3Bxc3R1d3h5enx9foCBgoO
PhoeliouMjo+QkZOUI ZaYmZqcnZ6foaKjpKanqKqrrK2vsLGztLW2uLm6u72+v8HCw8TGx8jJy8zN
z9DR0tTV1tfZ2tvd3t/g4UPk5ufo6evs7e7w8fL09fb3+fr7/P7 //////////////////////////////////wABA
gMEBQY HCAkKCwwNDg8QERIT FBUFFSU1RVVldYWVpbXF1eX2BhYmNkZWZnaGlqa2xtbm9wcXJzdHV2d3h5ent8fX5/glGCg4SFhoeliYqlJl2Oj5CRkpOUIZaXmJmam5ydnp+goaKjpKWm
p6ipqqusra6vsLGys7S1tre4ubq7vL2+v8DBwsPExcbHyMnKy8znzs/Q0dLT1NXW19jZ2tvc3d7f4OH
Hi4+TI5uf06err7O3u7/Dx8vP09fb3+Pn6+/z9/v9tZnQxAAAAAMB IQAA AQAAAAAAAAAAAAAAAA
AAAAAEAAAAAAAAAAAAAAAAAAAAAAAAABAAAAAQIDBAUGBWqJCqsMDQ4PEBESE xQVFhcYGRo

[illegible]

Wscj8+MW3rF7FgD
kmo/+a9wRF6z/+XAIKXY7ff
7Cmrb93qp2PG+
61q2o3ADOl6vRh0IAAA1s1f/+fVl
hAABjz/drlxvGhdm1v/69fLdnia5v/
/29z/
5tj/
/86Vb3Pe/yH86CTad9f/Rhz4AAAlryP/
/eiUAMAAABEpv/9V0GgAAAAAnj/7I8AAAAAAPgfn
/5pUEwAAAAAAefX/92YWiMAAAAAAePj/p
tlVZPkFaevz/+a9p6zF5/
38bd9bWAW
D+L+v/qnlkfAABOuv/+uYBIAAAAIivP/85
3IAAAAAACZtH/502AAAAAAAASrj/4mMAAAAAAAAANKb/
/sRwAAAAAAAAlZj/aQgAAAAAAAACyV/tlkY
GAAAAAAAHAH/6Z5TFAAAAAAAHL/++djQFAAADWX/
/66d8YmByiaf5/8vD/
6dDL6dDL

////////////////////////////////6LKIZ047sv////////////////////////////////NiE0cAAAAA+D////////////////////////////////9mEOg
AAAAAAOqz////////////////////////////////59CAAAAAAAAAAGIb////////////////////////////////2WICAAAAAAAAAAGnk/////

////////////////////////////////oSEAAAAAAAAAFLLQ////////////////////////////////6WgAAAAAAAAAAD6////////////////////////////////DD
QAAAAAAAAAACqw////////////////////////////////9oAAAAAAAAAAAABai////////////////////////////////8IAAAAAAAAAAA
AAAACU////////////////////////////////9CGMAAAAAAAAAAAAAACE////////////////////////////////2pVwaAAAAAAAAAABy///

//////////////////+SURxwAAAAAAAAABI////////////////////////////////zJFrWVVgbnuL////////////////////////////////98/
T+////////////////////////////////////
////////////////////////////////////
////////////////////////////////////
////////////////////////////////////
////////////////////////////////x+f////////////////////////////////

////////MyniG9XaPT////////FhVloBgAALLD////////7IsJQAAAAAAAHn4///
////////0XQdAAAAAAAAAEzl////////li4AAAAAAAAAACah////////dUw
AAAAAAAAAAAAAOA////////+xEgAAAAAAAAAAAAABk+f////////9nAAAAAAAAAA
AAAABN4////////kLAAAAAAAAAAAAAAA60v////////5QAAAAAAAAAAAAAAAAn
w////////wlAAAAAAAAAAAAAAAAARtP////////sgAAAAAAAAAAAAAAAAAAAp////////

////////VlleAAAAAAAAAAAAAAAAAan////////U/vW2bCQAAAAAAAAAAAAAIIP////////
///7xzPhsAAAAAAAADif////////z6OEdnN3gYqRpv////////////////////////////////////
////////////////////////////////////
////////////////////////////////////
////////////////////////////////////
////////////////////////////////hxKqSfrz////////Um29LLA4AA3v////////////////////////////////

///61mJgAAAAAAAEjK////////rlglAAAAAAAAABua////////UaAkAAAAAAA
AABxf////////+YlgAAAAAAAAAAAAABL2f////////+1TAAAAAAAAAAAAAAArvP////////

////////70QAAAAAAAAAAAAAAOOcf////////3YAAAAAAAAAAAAAAAIf////////
/xQAAAAAAAAAAAAAAAcv////////vQAAAAAAAAAAAAAAAAXf////////EgAAA
AAAAAAAAAAAAAAS////////NAAAAAAAAAAAAAAAAAAAAAAO/b////////OAAAAAAA
AAAAAAAAAAAAAMOr////////AAAAAAAAAAAAAAAAAAAAAJ93////////YZ10ORIAAA
AAAAAAAAAAH9D////////9bBzSS8cDwkKERkfJ8X////////7cy0raqttL/HZO////

////////////////////////////////////
////////////////////////////////////
////////////////////////////////////
////////////////////////////////p0NL////////z6ila08zfIXa////////7d2QBUA AAAACGk////////

////////nk4EAAAAAAAAABAxB////////+tSwAAAAAAAAAAAAABB0P////////+
BIAAAAAAAAAAAAAAWpv////////6MfAAAAAAAAAAAAAAAgv////////18AAAA
AAAAAAAAAAAAAY////////0BcAAAAAAAAAAAAAAAAAAR/L////////iAAAAAAAAAA
AAAAAALNr////////HQAAAAAAAAAAAAAAAAAAAE8P////////AAAAAAAAAAAA
AAAAAAAK3////////+nAAAAAAAAAAAAAAAAAAAAAJn////////VAIAAAAAAAAAAA
AAAAAAAlv////////AAAAAAAAAAAAAAAAAAAAAH3////////KAIAAAAAAAAAAAAA
AAAAAAHT////////cAAAAAAAAAAAAAAAAAAAAAG7////////zThFKg8AAAAAAAAAAAA

AAAGj//////////6L6VcFtMREBBR1BWW3H//////////+vj4/P//////////
//////////
//////////
//////////
//////////zdwqeNcVi4//////////
9yjdVAuEAAAAAB6//////////+pVoZAAAAAAAAAAAAABE1f//////////+XPwAAAAAAAAAA
AAAAUpP//////////7RGAAAAAAAAAAAAAAAAAd//////////+WoAAAAAAAAAAAAAAAAAA
AS/L//////////tilAAAAAAAAAAAAAAAAAKND//////////bgAAAAAAAAAAAAAAAAACLH/
//////////4HgAAAAAAAAAAAAAAAAAAAAAJX//////////+IAAAAAAAAAAAAAAAAAAAAAAHn///
//////////+OAAAAAAAAAAAAAAAAAAAAAGD//////////+zAAAAAAAAAAAAAAAAAAAAAEn///
//////////cAAAAAAAAAAAAAAAAAAAAADf//////////AAAAAAAAAAAAAAAAAAAAACry////////
//////////KgAAAAAAAAAAAAAAAAAAAAACDm//////////YgAAAAAAAAAAAAAAAAAAAAABfb////////
//////////qgAAAAAAAAAAAAAAAAAAAAA3S//////////xcAAAAAAAAAAAAAAAAAAAAAPK////////
//////////4AqLilPAAAAAAAAAAAAAADD//////////+OHR5yTjo6QlZylq6/Y//////////
//////////
//////////
//////////
//////////+48fM//////////uxaOEaUsuEABV6v//////////
//xIBJGwAAAAAAAAAarv//////////+aRwAAAAAAAAAAAAAd//////////503AAAAAAA
AAAAAAAAAROj//////////y0gAAAAAAAAAAAAAAAFrv//////////egAAAAAAAAAAAAAA
AAAAAAJX//////////vKAAAAAAAAAAAAAAAAAAAAAHP//////////+XAAAAAAAAAAAAAA
AAAAAAFT//////////99AAAAAAAAAAAAAAADb9//////////+fAAAAAAAAAAAAAA
AAAAAABri//////////CAAAAAAAAAAAAAAAADJ//////////mAAAAAAAAAAAAAA
AAAAAAACx//////////BwAAAAAAAAAAAAAAACe//////////MgAAAAAAAAAAAAAA
AAAAAAACQ//////////ZAAAAAAAAAAAAAAACG//////////ngAAAAAAAAAAAAAA
AAAAAAAB9//////////5QEAAAAAAAAAAAAAAAB1//////////1IAAAAAAAAAAAAAAA
AAAAAABr//////////8EAAAAAAAAAAAAAAABf//////////9xREU+Ni0nlyMIKTA4PUB
W//////////nv7u/y9vv//////////
//////////
//////////
//////////w1bqghGZLyP//////////8S
Ta0kpCwAAAAAAiP//////////+zZiUAAAAAAAAAAAAATfL//////////5o8AAAAAAAAAA
AAAAAAFrV//////////uToAAAAAAAAAAAAAAAIj//////////VgAAAAAAAAAAAAAA
AAAFn//////////+vAQAAAAAAAAAAAAAAADT2//////////9sAAAAAAAAAAAAAA
AAAABPV//////////+LAAAAAAAAAAAAAAAC2//////////+wAAAAAAAAAAAAAA
AAAAAAACZ//////////TAAAAAAAAAAAAAAAB+//////////2AAAAAAAAAAAAAA
AAAAAAABk//////////GAAAAAAAAAAAAAAABO//////////PwAAAAAAAAAAAAAA
AAAAAA7//////////awAAAAAAAAAAAAAAAt//////////ngAAAAAAAAAAAAAA
AAAAAAAj//////////2AAAAAAAAAAAAAAAc//////////0AAAAAAAAAAAAAA
AAAAAAR9f//////////48AAAAAAAAAAAAAAAE6v//////////oyAAAAAAAAAAAAAA
AAAAAA3f//////////+2AQAAAAAAAAAAAAAAAyv//////////rIWLh4B+foCFipGZoaap
yv//////////

////////////////////////////////////
////////////////////////////////////
////////+nK0f/////////0zq6RdVo+IAAAX/////////tomg4DwAAAAAAAAAAAH8f////////
/////////7NXCgAAAAAAAAAAAAAAAAA7/////////ujwAAAAAAAAAAAAAAAAAAAFj////////
/6RAAAAAAAAAAAAAAAAAAACTh/////////+BAAAAAAAAAAAAAAAAAAAAAAC3////////
//9sAAAAAAAAAAAAAAAAAAAAACT/////////+ZAAAAAAAAAAAAAAAAAAAAABx////////
////AAAAAAAAAAAAAAAAAAAAABS/////////jAAAAAAAAAAAAAAAAAAAAA0////////
////CQAAAAAAAAAAAAAAAAAAAAAY/////////LQAAAAAAAAAAAAAAAAAAAAA5v////////
////UQAAAAAAAAAAAAAAAAAAAAA0P/////////eQAAAAAAAAAAAAAAAAAAAAAvP////////
////////pQAAAAAAAAAAAAAAAAAAAAArf/////////1gAAAAAAAAAAAAAAAAAAAAAov////////
////////zgAAAAAAAAAAAAAAAAAAAAAm/////////3wAAAAAAAAAAAAAAAAAAAAAIP////////
/////////8sQAAAAAAAAAAAAAAAAAAAAiP/////////9uAAAAAAAAAAAAAAAAAAAAAd////////
////zQgAAAAAAAAAAAAAAAAAAAXf/////////7UYAAAIHCw8UGylqMTY4Qf////////
2/f79+/v+////////
////////////////////////////////////
////////////////////////////////////
////////////////////////////////////7dK3nH5d
QuH/////////MKUbk0uEQAAAAAAAJz/////////CORwwAAAAAAAAAAAAAF3////////
/////////0FMAAAAAAAAAAAAAAAAAACTi/////////3RgAAAAAAAAAAAAAAAAAAAAACs////////
/////////9pAAAAAAAAAAAAAAAAAAAAAB7/////////9uAAAAAAAAAAAAAAAAAAAAABU////////
/////////+gAAAAAAAAAAAAAAAAAAAAAAAt/////////OAAAAAAAAAAAAAAAAAAAAAAL7f//
/////////1AAAAAAAAAAAAAAAAAAAAAAzv/////////HwAAAAAAAAAAAAAAAAAAAAAs
v/////////QgAAAAAAAAAAAAAAAAAAAAAAI/////////ZAAAAAAAAAAAAAAAAAAAAAAf
v/////////igAAAAAAAAAAAAAAAAAAAAAAaP/////////swAAAAAAAAAAAAAAAAAAAAA
Vv/////////4AcAAAAAAAAAAAAAAAAAAAAAR/////////zkAAAAAAAAAAAAAAAAAAAAA
PP/////////3IAAAAAAAAAAAAAAAAAAAAAANf/////////7gAAAAAAAAAAAAAAAAAAAAAAL
f/////////9OAAAAAAAAAAAAAAAAAAAAAHv/////////+tDQAAAAAAAAAAAAAAAAAAACf//
/////////gAAAAAAAAAAAAAAAAAAAAAPb/////////44OAAAAAAAAAAAAAAAAAAAAAMb////////
/////////YYIFfaXN9hY2VnqWpqrT////////
////////////////////////////////////
////////////////////////////////////
////////////////////////////////////gucv////////
/////////3bucgGRJLAWAAGz/////////jmmEzCgAAAAAAAAAAAAACvq/////////4kuAAAA
AAAAAAAAAAAAAAACv/////////XQAAAAAAAAAAAAAAAB4/////////9qAAAAAA
AAAAAAAAAAAAAAABD/////////9nAAAAAAAAAAAAAAAU8f/////////+jAAAAAA
AAAAAAAAAAAAAAAYf/////////XAAAAAAAAAAAAAAAPp/////////CgAA
AAAAAAAAAAAAAAAG/////////MgAAAAAAAAAAAAAAAZP/////////WA
AAAAAAAAAAAAAAAR/////////fAAAAAAAAAAAAAALv/////////n
wAAAAAAAAAAAAAAAFv/////////xAAAAAAAAAAAAAAAf////////
7RUAAAAAAAAAAAAAAAPj/////////0IAAAAAAAAAAAAAAAOf////////
//3UAAAAAAAAAAAAAANr/////////7AAAAAAAAAAAAAANH////////
////M5AAAAAAAAAAAAAAMn/////////+MAAAAAAAAAAAAAAALv////////
////////sTQAAAAAAAAAAAAAAKb/////////xDgAAAAAAAAAAAAAIP////////
////9BNAAAAAAAAAAAAAEn/////////ni8AAAAAAQPGiMoK0n////////

////////////////////////////////////
////////////////////////////////////
////////////////////////////////////31rKLYU/4////////////////////////////////8WWbUoqDQAAAAAAAAAC2////////////////////////////////9h5
MQAAAAAAAAAAAAAAAAAB3////////////////////////////////mB8AAAAAAAAAAAAAAAAABAAAAAAAAAAAAA+FAAA
AAAAAAAAAAAAAAAAAAAAAK5P////////////////////////////////9VAAAAAAAAAAAAAAAAAAAAASv////////////////////////////////+d
AAAAAAAAAAAAAAAAAAAAAAAhv////////////////////////////////ZAAAAAAAAAAAAAAAAAAAAAXf////////////////////////////////
//FgAAAAAAAAAAAAAAAAAAAAAOP////////////////////////////////QwAAAAAAAAAAAAAAAAAAAAAFv////////////////////////////////
////awAAAAAAAAAAAAAAAAAAAAAAP3////////////////////////////////kAAAAAAAAAAAAAAAAAAAAAOL//
////////tQAAAAAAAAAAAAAAAAAAAAAMv////////2gMAAAAAAAAAAAAAAAAAAAAAAL
X////////ygAAAAAAAAAAAAAAAAAAAAAKD////////1AAAAAAAAAAAAAAAAAAAAA
AI3////////30AAAAAAAAAAAAAAAAAAAAAH3////////7IAAAAAAAAAAAAAAAAAAAAAA
AHH////////+0zAAAAAAAAAAAAAAAAAAAAAGf////////95AAAAAAAAAAAAAAAAAAAAA
AF3////////JKwAAAAAAAAAAAAAAAAAAAAE7////////jQUAAAAAAAAAAAAAAAAAAAAAD
H////////3kDAAAAAAAAAAAAAAAAAAAAAb////////+SlgAAAAAAAAAAAAAAAAAAAAAD7////
////////4nUXFyApNEFRZHiSuOr////////
////////
////////fT////////
////////9LiKd2dZTUA0JxkKAAAw////////73gbAAAAAAAAAAAAAAAAAAAA3f////////CGwA
AAAAAAAAAAAAAAAAAAAAAp////////84AAAAAAAAAAAAAAAAAAAAAc////////+PA
AAAAAAAAAAAAAAAAAAAAAAARf////////XAAAAAAAAAAAAAAAAAAAAAGf////////
/GgAAAAAAAAAAAAAAAAAAAAAO////////TgAAAAAAAAAAAAAAAAAAAAAMv////////
////fAAAAAAAAAAAAAAAAAAAAAKr////////pQAAAAAAAAAAAAAAAAAAAAAlz////
////////ywAAAAAAAAAAAAAAAAAAAAAHL////////7xgAAAAAAAAAAAAAAAAAAAAAFr//
////////z0AAAAAAAAAAAAAAAAAAAAAEb////////2QAAAAAAAAAAAAAAAAAAAAADT
////////44AAAAAAAAAAAAAAAAAAAAACH////////7sAAAAAAAAAAAAAAAAAAAAA
BL////////+40AAAAAAAAAAAAAAAAAAAAAX////////9yAAAAAAAAAAAAAAAAAAAAA
AAD////////+5GQAAAAAAAAAAAAAAAAAAAAAD////////awAAAAAAAAAAAAAAAAAAAAA
AAD////////zUQAAAAAAAAAAAAAAAAAAAAADp////////71CAAAAAAAAAAAAAAAAAAAAAA
C4////////aZwYAAAAAAAAAAAAAJJ06x////////7yHIJ+rt8bX7f////////
////////
////////
////////9Obb0ce8saWakouFz////////bOJaE02lxI
EAAAAAAAAAAAAAXP////////+QQgoAAAAAAAAAAAAAAAAAAAAALf////////9yAAAAAAA
AAAAAAAAAAAAAAPz////////NAAAAAAAAAAAAAAAAAAAAAND////////FQAA
AAAAAAAAAAAAAAAKX////////UwAAAAAAAAAAAAAAAAAAAAAH7////////h
wAAAAAAAAAAAAAAAFr////////tQAAAAAAAAAAAAAADn////////
//3wkAAAAAAAAAAAAABv////////zAAAAAAAAAAAAAH////////
////1UAAAAAAAAAAAAAD////////3kAAAAAAAAAAAAAD////////
////////6AAAAAAAAAAAAADy////////8sQAAAAAAAAAAAAADi//
////////o/AAAAAAAAAAAAADR////////90AAAAAAAAAAAAAD
D////////+wEAAAAAAAAAAAAAC2////////3WQAAAAAAAAAAAAA
ACm////////qyMAAAAAAAAAAAAAACP////////4QNAAAAAAAAAAAAA

ABr////////////////////////////////+FHAAAAAAAAAAAAAAAAAAr////////////////////////////////rUcMFyltOUdXaoGbven////////////////////////////////
////////////////////////////////+//
//
//
////////////////////////////////007elloI+dGthWE5FOzlrJd////////////////////////////////toWpClwoAAAAAAAAAAAAAAAAAAAAAK3/////////
//////+0CQAAAAAAAAAAAAAAAAAAAAAIP////////////////////////////////AwAAAAAAAAAAAAAAAAAAAAAAAFr//
////////////////////////////////TgAAAAAAAAAAAAAAAAAAAAADH////////////////////////////////jAAAAAAAAAAAAAAAAAAAAAA
Av////////////////////////////////wwAAAAAAAAAAAAAAAAAAAAAAD////////////////////////////////8RsAAAAAAAAAAAAAAAAAAAA
AAAADw////////////////////////////////0UAAAAAAAAAAAAAAAAAAAAAADU////////////////////////////////2wAAAAAAAAAAAAAAAA
AAAAAAAC7////////////////////////////////5IAAAAAAAAAAAAAAAAAAAAAAACI////////////////////////////////7cAAAAAAAAAAAAAAAA
AAAAAAACR////////////////////////////////94jAAAAAAAAAAAAAAAAAAAAAAB////////////////////////////////9OAAAAAAAAAAAAAA
AAAAAAABw////////////////////////////////9+AAAAAAAAAAAAAAAAAAAAAABh////////////////////////////////+0EwAAAAAAAA
AAAAAAABBS////////////////////////////////xUQAAAAAAAAAAAAAAAAAAAAABC////////////////////////////////mA8AAAAAA
AAAAAAAAu////////////////////////////////7WQAAAAAAAAAAAAAAAAAAAAAP////////////////////////////////8pQAAAAAA
AAAAAAAAAAA////////////////////////////////MWwAAAAAAAAAAAAABRkwTXKk////////////////////////////////+Zh9jJiks8LV6//
////////////////////////////////
////////////////////////////////
////////////////////////////////
////////////////////////////////rv5tzUzMW+ubSytsf////////////////////////////////suliHBdTKE1KyEYDgMAAAAAACz//
////////////////////////////////ZTMLAAAAAAAAAAAAAAAAAAAAAT////////////////////////////////OwAAAAAAAAAAAAAAAAAAAAAA
AD////////////////////////////////iwAAAAAAAAAAAAAAAAAAAAADe////////////////////////////////ywAAAAAAAAAAAAAAAAAAAA
AAAAC6////////////////////////////////ycAAAAAAAAAAAAAAAAAAAAACY////////////////////////////////1cAAAAAAAAAAAAAA
AAAAAB5////////////////////////////////4IAAAAAAAAAAAAAAAAAAAAAABe////////////////////////////////6kAAAAAAAAAAAAAA
AAAAABF////////////////////////////////88VAAAAAAAAAAAAAAAAAAAAAv////////////////////////////////U7AAAAAAAAAAAA
AAAAAAAAAc////////////////////////////////9jAAAAAAAAAAAAAAAAAAAAAL////////////////////////////////+NAAAAAAAAAA
AAAAAAAAAAA////////////////////////////////+8HAAAAAAAAAAAAAAAAAAAAA////////////////////////////////zVAAAAAAAA
AAAAAAAAAAA////////////////////////////////kggAAAAAAAAAAAAAAAAAAAAA////////////////////////////////108AAAAAA
AAAAAAAAAAA////////////////////////////////6QuAAAAAAAAAAAAAAAAAAAAA6P////////////////////////////////+UKQAAAAAA
AAAAAAABpF6P////////////////////////////////pj8gKzZDUF9wg5q11v////////////////////////////////u////////////////////////////////
////////////////////////////////
////////////////////////////////
////////////////////////////////
////////////////////////////////6NPBtKidllqDfXZwamdmZ27////////////////////////////////wYxnSTEEUA
AAAAAAACp////////////////////////////////fwUAAAAAAAAAAAAAAAAAAAAACF////////////////////////////////zgAAAAAA
AAAAAAABi////////////////////////////////ywAAAAAAAAAAAAAAAAAAAAA////////////////////////////////2QAAAA
AAAAAAAFf////////////////////////////////5QAAAAAAAAAAAAAAAAAAAAAB////////////////////////////////8AG
AAAAAAAAAAAAAAAAAAAAAAAAAAA////////////////////////////////+guAAAAAAAAAAAAAAAAAAAAA////////////////////////////////
9UAAAAAAAAAAAAAAAAAAAAA////////////////////////////////96AAAAAAAAAAAAAAAAAAAAAv////////////////////////////////
////+iAgAAAAAAAAAAAAAAAAAAAAA7P////////////////////////////////OLQAAAAAAAAAAAAAAAAAAAAA3P/////
////////////////////////////////9XQAAAAAAAAAAAAAAAAAAAAzP////////////////////////////////kwkAAAAAAAAAAAAAAAAAAAAu////
////////////////////////////////0UkAAAAAAAAAAAAAAAAAAAApP////////////////////////////////5EbAAAAAAAAAAAAAAAAAAAAhv///
////////////////////////////////+luBwAAAAAAAAAAAAABbP////////////////////////////////dbQ4AAAAABhUmOU5lg6nZ////////////////////////////////

////////8puNnKi2xdXp////////
////////
////////
////////
////////Pfx7Obh3dnZ3+r6////////+i/pZGCdWtiWINLRT02Lykklic8////////11s5HgkAAAA
AAAAAAAAAAAAAAAAAAAA////////ycAAAAAAAAAAAAAAAAAAAAAAAAAAAA////////2oAAAAAA
AAAAAAAAAAAAAAAAAAAA////////6UAAAAAAAAAAAAAAAAAAAAAAAAAAAA9v////////9UaA
AAAAAAAAAAAAAAAAAAAA2v////////9FAAAAAAAAAAAAAAAAAAAAAAAAAAAAAwf////////
9uAAAAAAAAAAAAAAAAAAAAq////////+VAAAAAAAAAAAAAAAAAAAAAAAAAAAAlv////////
////+7GwAAAAAAAAAAAAAAAAAAAAhP////////jQwAAAAAAAAAAAAAAAAAAAAAc////////
////////bwAAAAAAAAAAAAAAAAAAAAAYf////////nxUAAAAAAAAAAAAAAAAAAAAUP///
////////1UsAAAAAAAAAAAAAAAAAAAAOv////////4oTAAAAAAAAAAAAAAAAAAAAHv///
////////9ReAAAAAAAAAAAAAAAAAAAAAP////////+3SgAAAAAAAAAAAAARYtSGuWzf////////
////////t1JDT1xpelmds8zm////////4////////
////////
////////
////////
////////+7g1MvDvLWvqaOemZeYm
6W////////7eQdmJSRDkwJyAaFA4HAQAAAAAA0f////////3AYAAAAAAAAAAAAAAAAAAAA
AAAAAr////////7IAAAAAAAAAAAAAAAAAAAAAAAkv////////+ooAAAAAAAAAAAAAAAAAAAA
AAAAAAAdf////////9aAAAAAAAAAAAAAAAAAAAAAAWv////////+HAAAAAAAAAAAAA
AAAAAAAAAAAAAQv////////+vDQAAAAAAAAAAAAAAAAAAAAALP////////WNgAAAAAA
AAAAAAAAAAAAAGP////////9XQAAAAAAAAAAAAAAAAAAAAABf////////hgAAAA
AAAAAAAAAAAAAAP////////sScAAAAAAAAAAAAAAAAAAAAAAP////////4lgAA
AAAAAAAAAAAAAAP////////44XAAAAAAAAAAAAAAAP////////8tVA
AAAAAAAAAAAAAAP////////+gNwAAAAAAAAAAAAAJJERsn////////kjALG
CUyQVJle5OvzfL////////7WpucfW5vn////////
////////
////////
////////
////////
////////oy7aonZWOilJ9eHNva2hnanGEnv////////89oUD0ulhgQCAEAAAAAAAAAAAA
AAJf////////wtAAAAAAAAAAAAAAAAAAAAAB////////9pAAAAAAAAAAAAAAAAAAAA
AAAAAP////////+cAAAAAAAAAAAAAAAAAAAAAP////////JJwAAAAAAAAAAAAA
AAAAAAAP////////yUQAAAAAAAAAAAAAAAP////////eQAAAAAAAAAAAA
AAAAAAAP////////oBUAAAAAAAAAAAAAAAP////////yj8AAAAAAAAAAAA
AAAAAAAPP////////9WsAAAAAAAAAAAAAAAN////////5sjAAAAAAA
AAAAAAAMj////////9BbAAAAAAAAAAAAAAKv////////+ZMQAAAAA
AAAAAAADy9Ug+////////nexwAAAAMGyw/VG2JqMz1////////3H9vf46drsHX8f///
////////
////////
////////

9e3n4t3Z1dLPzc3O09/v//////////BopCCd25mYFpVUEtHQ0A/QEZSav//////////+BNSITBwAAA
AAAAAAAAAAAAAAAAAAP//////////+zBwAAAAAAAAAAAAAAAAAAAAAOj//////////kPQAAA
AAAAAAAAAAAAAAAAAAAAANH//////////awAAAAAAAAAAAAAAAAAAAAALr//////////lgo
AAAAAAAAAAAAAAAAAAAAAKb//////////vjIAAAAAAAAAAAAAAAAAAAAAJL//////////5V
oAAAAAAAAAAAAAAAAAAAAAH7//////////4MLAAAAAAAAAAAAAAAAAAAAAGn//////////
/7A4AAAAAAAAAAAAAAAAAAAAAFL//////////+BpAAAAAAAAAAAAAAAAAAAAAEL//////////
///+dNgAAAAAAAAAAAAAACChOd6z//////////fdRgAAAAAAhMmPFVykrbf//////////xW
JGVWR0hZmvyOX//////////9/P4/P//////////
9eDSx7+5tLCsqaajokI
qrTD3v//////////jg29iV09HQTW3Mi0oJCEfHyQvRpj//////////Tw4AAAAAAAAAAAAAAAAAAAA
AAFj//////////gwAAAAAAAAAAAAAAAAAAAAAEH//////////siQAAAAAAAAAAAAAAAAAAAA
AAAACz//////////3FAAAAAAAAAAAAAAAAAAAAAABf//////////3kAAAAAAAAAAAAAAAAAAAA
AAAAAAP//////////6EoAAAAAAAAAAAAAAAAAAAAAAD//////////8pRAAAAAAAAAAAAAAAAA
AAAAAAD//////////V9FAAAAAAAAAAAAAAAAAAAAAATb//////////+tRQAAAAAAAAAAAAA
AADzJZg7f//////////ieh0AAAAAAAcBmKxqi7Da//////////vFwqOUIZan6Vrszs//////////
//////////7ugtcbY7P//////////
fv6+/z//////////
///272vpJ2WkY2Jh4WDgoSHjZeqwO7//////////rllLQDgxLCYiHRkUDwwJBwkTJUj//////////0T
wAAAAAAAAAAAAAAAAAAAAAAD//////////W0AAAAAAAAAAAAAAAAAAAAAAD//////////
///5keAAAAAAAAAAAAAAAAAAAAAAD//////////8FHAAAAAAAAAAAAAAAAAAAAAAD//////////
////+lwBgAAAAAAAAAAAAAAAAAAAAAAD//////////+ZLwAAAAAAAAAAAAAAAAAAAAAJ1n//////////
//////EWgAAAAAAAAAAAAAACl0dxntX//////////viSwAAAAAAAgdNVFxILrk//////////wWl
ZKzpLXXKKpMPI//////////6uBl6m80Of//////////
/67+jj39zZ2NjY2t7i6vb//////////6
+Vi4N9eXVycG5ubm9ze4aYsNn//////////5M8MiojHRkUDwsGAQAAAAAAAEzXg//////////7k8
AAAAAAAAAAAAAAAAAAAAAC0//////////+JpAAAAAAAAAAAAAAAAAAAAACf//////////+
SJgAAAAAAAAAAAAAAAAAAAAAyI//////////+5TwAAAAAAAAAAAAAAAAAAAAAnVYnb//////////i

[illegible]


```
<%@ Control Language="C#" AutoEventWireup="true"  
CodeBehind="av_default_footer_template.ascx.cs"  
Inherits="Web_App_Master.Account.Templates.av_default_footer_template" %>  
    </div>  
  
    </div>  
    </div>  
    </div>  
</div>
```

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace Web_App_Master.Account.Templates
{
    public partial class av_default_footer_template : System.Web.UI.UserControl
    {
        protected void Page_Load(object sender, EventArgs e)
        {

        }
    }
}
```

```
<%@ Control Language="C#" AutoEventWireup="true"
CodeBehind="av_default_header_template.ascx.cs"
Inherits="Web_App_Master.Account.Templates.av_default_header_template" %>
<div class="row">
    <div class="col-md-12">
        <div class="awp_box rounded bg-starrag shadow">
            <div class="awp_box_title bg-sg-title">
                <span class="fg-white shadow-metro-black" style="text-align:center
!important;"><h2><strong>Assets</strong> </h2></span>
            </div>
            <div class="awp_box_content bg-sg-box" style="text-align:left !important;">
                <div class="listview padding10 default" data-role="listview" id="lv1">
```

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace Web_App_Master.Account.Templates
{
    public partial class av_default_header_template : System.Web.UI.UserControl
    {
        protected void Page_Load(object sender, EventArgs e)
        {

        }
    }
}
```

```
<%@ Control Language="C#" AutoEventWireup="true"  
CodeBehind="av_default_template.ascx.cs"  
Inherits="Web_App_Master.Account.Templates.av_default_template" %>
```

```
<div class="list shadow" style="overflow:hidden;" onclick="BarcodeScanned('<##  
Eval("AssetNumber")%>', '<## Eval("IsHistoryItem")%>', '<## Eval("DateShippedTicks")%>');">  
  <i title="Add To Task List" style="font-size:1.5em;" class="glyphicon glyphicon-plus c-lime av-  
add-cursor" onclick="<##  
prepareAjax(Eval("AssetNumber").ToString(),Eval("IsOut").ToString())%>"></i>  
  <img src='<## Eval("FirstImage") %>' class="list-icon">  
  <span id="InOutIndicator" runat="server" class="list-title fg-white shadow-metro-black"  
style="margin-left:5px;padding-left:5px;"><strong><## Eval("AssetNumber") %> - <##  
Eval("AssetName") %></strong></span>  
</div>
```



```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace Web_App_Master.Account.Templates
{
    public partial class av_default_template : System.Web.UI.UserControl
    {
        protected void Page_Load(object sender, EventArgs e)
        {

        }

        protected string prepareAjax(string num, string isout)
        {
            if (isout.ToLower() == "true")
            {
                string str = "AjaxAddCheckin(";
                str = str + num;
                str = str + ")";
                return str;
            }
            else
            {
                string str = "AjaxAddCheckout(";
                str = str + num;
                str = str + ")";
                return str;
            }
        }
    }
}

```

```
<%@ Control Language="C#" AutoEventWireup="true"  
CodeBehind="av_footer_template.ascx.cs"  
Inherits="Web_App_Master.Account.Templates.av_footer_template" %>  
</div>
```

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace Web_App_Master.Account.Templates
{
    public partial class av_footer_template : System.Web.UI.UserControl
    {
        protected void Page_Load(object sender, EventArgs e)
        {

        }
    }
}
```

```
<%@ Control Language="C#" AutoEventWireup="true"  
CodeBehind="av_header_template.ascx.cs"  
Inherits="Web_App_Master.Account.Templates.av_header_template" %>  
<div class="tile-container bg-metro">
```

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace Web_App_Master.Account.Templates
{
    public partial class av_header_template : System.Web.UI.UserControl
    {
        protected void Page_Load(object sender, EventArgs e)
        {

        }
    }
}
```

```
<%@ Control Language="C#" AutoEventWireup="true"  
CodeBehind="av_history_footer_template.ascx.cs"  
Inherits="Web_App_Master.Account.Templates.av_history_footer_template" %>  
    </div>  
    </div>  
    </div>  
    </div>  
</div>
```

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace Web_App_Master.Account.Templates
{
    public partial class av_history_footer_template : System.Web.UI.UserControl
    {
        protected void Page_Load(object sender, EventArgs e)
        {

        }
    }
}
```

```
<%@ Control Language="C#" AutoEventWireup="true"  
CodeBehind="av_history_header_template.ascx.cs"  
Inherits="Web_App_Master.Account.Templates.av_history_header_template" %>  
    <div class="col-md-12" style="margin-top:-21px!important; margin-bottom:0px  
!important;">
```

```
    <div class="awp_box rounded bg-metro-dark shadow">  
        <div class="awp_box_title bg-metro-dark">  
            <span class="fg-white shadow-metro-black">History</span>  
        </div>  
        <div class="awp_box_content bg-metro-light" style="text-align:left !important;">  
            <div id="AssetHistoryDiv" style="min-height:300px; text-align:center;">  
  
                <div class="listview padding10 default" data-role="listview" id="lv1">
```



```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace Web_App_Master.Account.Templates
{
    public partial class av_history_header_template : System.Web.UI.UserControl
    {
        protected void Page_Load(object sender, EventArgs e)
        {

        }
    }
}
```

```
<%@ Control Language="C#" AutoEventWireup="true"  
CodeBehind="av_history_template.ascx.cs"  
Inherits="Web_App_Master.Account.Templates.av_history_template" %>
```

```
<div class="list shadow" style="overflow:hidden; text-align:left" onclick="BarcodeScanned('<%#  
Eval("AssetNumber")%>', '<%# Eval("IsHistoryItem")%>', '<%# Eval("DateShippedTicks")%>');">  
    <span id="InOutIndicator" runat="server" class="list-title fg-white shadow-metro-black"  
style="margin-left:5px;padding-left:5px;"><strong><%# Eval("AssetNumber") %> - <%#  
Eval("AssetName") %> - <%# Eval("DateShippedString") %> - <%# Eval("DateRecievedString")  
%></strong></span>  
</div>
```

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace Web_App_Master.Account.Templates
{
    public partial class av_history_template : System.Web.UI.UserControl
    {
        protected void Page_Load(object sender, EventArgs e)
        {

        }
    }
}
```

```
<%@ Control Language="C#" AutoEventWireup="true" CodeBehind="av_list_template.ascx.cs"
Inherits="Web_App_Master.Account.Templates.av_list_template" %>
```

```
    <div class="col-md-12" >
        <span class="av-lv-linkbadge"><i title="Add To Task List" style="font-size:1.5em;"
class="glyphicon glyphicon-plus c-lime av-add-cursor" onclick="< %#
prepareAjax(Eval("AssetNumber").ToString(),Eval("IsOut").ToString())%>"></i></span>
        <div class=" av-lv rounded bg-sg-box shadow" onclick="BarcodeScanned('< %#
Eval("AssetNumber")%>', '< %# Eval("IsHistoryItem")%>', '< %# Eval("DateShippedTicks")%>');">
            <div class="av-lv-row" >
                <div class=" av-lv-thumb" >"
width="80" height="70"></img></div>
                <div class="av-lv-seperator"></div>
                <div class="av-lv-content fg-white shadow-metro-black" >
                    <strong>< %# Eval("AssetName")%> - < %# Eval("Description")%></strong>
                </div>
                </div>
                <span id="InOutIndicator" runat="server" class="edge-badge fg-white
shadow"><strong>< %# Eval("AssetNumber")%></strong></span>
            </div>
        </div>
```

```

using Helpers;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace Web_App_Master.Account.Templates
{
    public partial class av_list_template : System.Web.UI.UserControl
    {

        protected void Page_Load(object sender, EventArgs e)
        {

        }

        protected string prepareAjax(string num, string isout)
        {
            if (isout=="true")
            {
                string str = "AjaxAddCheckin(";
                str = str + num;
                str = str + ")";
                return str;
            }
            else
            {
                string str = "AjaxAddCheckout(";
                str = str + num;
                str = str + ")";
                return str;
            }
        }
    }
}

```

```

<%@ Control Language="C#" AutoEventWireup="true"
CodeBehind="av_sm_tile_template.ascx.cs"
Inherits="Web_App_Master.Account.Templates.av_sm_tile_template" %>
<div id="TESTTEST"></div>
<div id="TESTSERVER" runat="server">

<div id="DataItemTile" class="tile tile-small-y bg-metro-light" data-role="tile"
onclick="BarcodeScanned('<%# Eval("AssetNumber")%>', '<%# Eval("IsHistoryItem")%>', '<%#
Eval("DateShippedTicks")%>');">
    <div id="IsOutContainer" runat="server" class="tile-content zooming bg-metro-dark ">
        <div class="image-container zooming bg-metro-light" style="margin-top:25px">
            <div class="frame">
                ">
            </div>

        </div>
        <span class="av-smtile-linkbadge shadow-metro-black"><i title="Add To Task List"
style="font-size:1.5em;" class="glyphicon glyphicon-plus av-add-cursor c-lime" onclick="<%#
prepareAjax(Eval("AssetNumber").ToString(),Eval("IsOut").ToString())%>"></i></span>
        <span id="InOutIndicator" runat="server" class="tile-badge fg-white bg-metro-dark
shadow-metro-black"><strong><%# Eval("AssetNumber")%></strong></span>
        <div class="tile-label fg-white shadow-metro-black bg-metro-dark" style="width:100%;
text-align:center"><span style="z-index:1004"><strong><%#
Eval("AssetName")%></strong></span></div>
    </div>
</div>
</div>
</div>

```

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace Web_App_Master.Account.Templates
{
    public partial class av_sm_tile_template : System.Web.UI.UserControl
    {
        protected void Page_Load(object sender, EventArgs e)
        {

        }

        protected string prepareAjax(string num, string isout)
        {
            if (isout == "true")
            {
                string str = "AjaxAddCheckin(";
                str = str + num;
                str = str + ")";
                return str;
            }
            else
            {
                string str = "AjaxAddCheckout(";
                str = str + num;
                str = str + ")";
                return str;
            }
        }
    }
}

```

```
<%@ Control Language="C#" AutoEventWireup="true" CodeBehind="emptytemplate.ascx.cs"  
Inherits="Web_App_Master.Account.Templates.emptytemplate" %>
```



```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace Web_App_Master.Account.Templates
{
    public partial class emptytemplate : System.Web.UI.UserControl
    {
        protected void Page_Load(object sender, EventArgs e)
        {

        }
    }
}
```

```
<%@ Control Language="C#" AutoEventWireup="true" CodeBehind="icon_view.ascx.cs"
Inherits="Web_App_Master.Account.Templates.icon_view" %>
<!--Icon Template-->
```

```
<div class="file " title=" draggable="true" onclick="BarcodeScanned('<%#
Eval("AssetNumber")%>', '<%# Eval("IsHistoryItem")%>', '<%# Eval("DateShippedTicks")%>');">
    <div id =" file-color " class="file-color <%# Eval("Color") %>" style="width:100%;
height:100%;"></div>
```

```
    <div class="i-icon">
        <img style="box-shadow: rgba(0, 0, 0, 0.70) 0px 0px 10px;" src='<%#
Eval("FirstImage")+ ".r?w=133&h=100" %>' />
    </div>
```

```
    <div runat="server" class="i-title"><strong class="fg-white shadow-metro-black"><%#
Eval("AssetName") %></strong> </div>
```

```
    <div class="i-check">
<span id="InOutIndicator" runat="server" class="edge-badge fg-white shadow-metro-black icon-
badge " style="z-index:3 !important;"><strong><%#
Eval("AssetNumber")%></strong></span></div>
```

```
    <div runat="server" id="DownloadLink" class="i-download" style="margin-top:-4px;padding-
left:2px">        <i title="Add To Task List" id="dl_link" style="font-size:1.5em;" class="glyphicon
glyphicon-cloud-download c-lime av-add-cursor" onclick="<%#
prepareAjax(Eval("AssetNumber").ToString(),Eval("IsOut").ToString())%>"></i>
</div>
```

```
</div>
```

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace Web_App_Master.Account.Templates
{
    public partial class icon_view : System.Web.UI.UserControl
    {
        protected void Page_Load(object sender, EventArgs e)
        {

        }

        protected void prepareImg()
        {
        }

        protected string prepareAjax(string num, string isout)
        {
            if (isout.ToLower() == "true")
            {
                string str = "AjaxAddCheckin(";
                str = str + num;
                str = str + ")";
                return str;
            }
            else
            {
                string str = "AjaxAddCheckout(";
                str = str + num;
                str = str + ")";
                return str;
            }
        }

        protected void RenameInputBox_TextChanged(object sender, EventArgs e)
        {

        }
    }
}

```

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
```

```
namespace Web_App_Master.Account.Templates
```

```
{
    public class StarragPackingSlipTemplate:Pdf.PdfTemplate
    {
        public StarragPackingSlipTemplate()
        {
            this.TemplateBase64 =
"JVBERi0xLjUNCiW1tbW1DQoxIDAgb2JqDQo8PC9UeXBIL0NhdGFsb2cvUGFnZXMGMiAwIFlvT
GFuZyhlbi1VUykgL1N0cnVjdFRyZWVSb290IDEyIDAgUi9NYXJrSW5mbzw8L01hcmtlZCB0cnVIPj
4+Pg0KZW5kb2JqDQoyIDAgb2JqDQo8PC9UeXBIL1BhZ2VzL0NvdW50IDEvS2lk1sgMyAwIFJdl
D4+DQplbmRvYmoNCjMgMCBvYmoNCjw8L1R5cGUvUGFnZS9QYXJlbnQgMiAwIFlvUmVzb3Vy
Y2VzPDwvRm9udDw8L0YxIDUgMCBSL0YyIDkgMCBSPj4vRXh0R1N0YXRIPDwvR1M3IDcgMC
BSL0dTOCA4IDAgUj4+L1hPYmplY3Q8PC9JbWFnZTEuIDAgUj4+L1Byb2NTZXRbL1BERi9
UZXh0L0ltYWdlQI9JbWFnZUMvSW1hZ2VJXSA+Pi9NZWRpYUJveFsgMCAwIDU5NS4zMjA4ND
EuOTJdIC9Db250ZW50cyA0IDAgUi9Hcm91cDw8L1R5cGUvR3JvdXAvUy9UcmFuc3BhcmVvY3k
vQ1MvRGV2aWNIUkdCPj4vVGficy9TL1N0cnVjdFBhcmVudHMgMD4+DQplbmRvYmoNCjQgMC
BvYmoNCjw8L0ZpbHRlci9GbGF0ZURlY29kZS9MZW5ndGggMzMxOT4+DQpzdHJlYW0NCicrZ
xdbx23EYbvBeg/LJCbcwJ4ze8liYBAYyep26RNEAO9KHohul5qFFESx7nlvy+5Sy6HX0OKMlo4kM
6r4exL8uEsD3d/vb3hahV62ZRcuVokE6vWiyAr0cv7t7c3//x0ebi9WZ5/t3z22fNvX7x6uZDn39w93C
+X/7x99ulv188/X754+WL5dSzOF69vb55/RRezGrW8/vH2hi7E/o8ugq5KLJtRK2fL65+s6usftuX+t9
sbstzvP2n/09e3N/+6fHd9Rs3lz1fGLy/+9urvX1+fqcvyjf2luLz64Sovr6//XI7/9fbmS9ve97c3T82OUrY
yBtPbs/LJLGlj9v/LI9++WKBn1NtE1k0Y96c2CU3kKoT9lc1r/8cl8uOnvc/cT/5jfxmJQCq6pgEIEY+W
2I+9Vfuvo1Hg74HA/X2q2R04bZdSr3phiq1ULYyo9uBi6OBycTZZBJJHIPOI0SXluqmi+17bYSMvP1
+fscufGsNmtl25razdcGPc2FaZMSsT0DaO28Ybtu2BeJl/f0T+nliVba38v3KWvb/yw7ufSvOe2jpnqz
SPdA+6lrwR+2CNU6UyofzHvoMTATN6VRwGIITmErNSCoMwWZ1ySZByynmj9l8Dm0AWQOGy
yES+IUQkclE2MyIfOQ1TU8jG+JL4tARBZBqkMyWlsM6Zom93sH/Y56X9l18e0Mk50brtByOazfcnJ
miSt11TnVnZjtKbkY3Ev7uKy/oP59gnzck40yjbVtJsdGAibnEiusHJaZylbhinc9ErwFyEGjARfJgwHV
PVPiNBKFaGAtPIDCUqSQXPjqVQFqnET93fAUFoAAoEFBRG6bNO0NQtrZpo/+7++CPMDaqPB
cjCiGSTm6oVEBZjITj2/WzklipJbWkq+a2+FIFO2NKchq08uvVWEyvxjbN141WB/bd+6vHwflSooFe
3v52fbZd3rhfv/vlgysB3/38kA7GMI36kdJ1VGW0le9lSch8Jn5i+jwot/3NGt3q8dOSlv1KeZh5nhtHz1J
GVsV6PSsKYuENt/khiVvlx/rWdi3J8VXmJj9abptehXpKP6owd2zLst05sB93aZoyx/txOxuxxRDsRtn
rRR2d6mTYW7E3uaqmU99fn4nL726dvnuwHWmn5Af3wx+9jjQfKz1J1b40T3ckl5WbNGnv/GzfYM
vVoegsVzEMtlwdKnflplyInD1E9UJ5E0Eoa2Jr5YOhqiuf74ujlk2Gsr8wlNgL0kTjM0Y1PpVEI/BVTR
mHsriqqda9JEWrvVhhH5nGOYcZadavQ1F39zCg2Kfu7wK7jITikFdN9NeVIYvPyC82s8+qk/yStaH
BKujHGm66FRqft0tmdO07FeAKIBx1SiVsBUbJrk9bSa8pn3zT8zbpCruEsTlsGLu8AmcXCIOwy6tw
dnVCnQvAGarNriRUnV1HTwR2qeK64uceS6rIF5GEPKBEVEZk0k1mpL4Xm7vNyrIUASEQoiDkJ
CnvhduW2kB5jw1xXhbM9fzGuCnNKuXMCOD5uXxkMYLMurJiU47Mw6cxZPIKMrGGu8icdypHZt+
```

pgEygRJHJM2RGo7rI5BVkTvkUmp62qYpMFXfg7D+bMfI0oVdEZCaaEy3JNl2pOpB5qClyE9GJI
DTUueqcoQ5kJqITVcm+YHl9oSfaylyfN5GJSEleUFJDZkQN46fZlnBsNaw1lg1e64FAsgjUgxbfrFV
TG9CkVuWduYThfuZSQ1dDWl6/oBm77Oxwc2qIXYKV7EJb7sJr3jGewWvArEAvIOW4WSLFV/Sqi
y8hS3zNWRXannZK1fgllLv7xUq+Q9Ep+WlYrOQ7VJ2SDw91LgFnKKTkg6HqJd/RFQe/krESLiwK
djolmpAxpmpQI2ojcukq7aRso9Q0DEYFIEQLfuETss+O7QdA4fKPmEKhNbZGyMoFe4LhIIRLklW
9h1ZjJR9dWXlu60cnYdPY+iUFXRiDXfJOe9UTs6+UwGcQImWfTLjZjSqy01Z4eaUT6HpaZuq2GS
886WUV3SwyfiAl1Je1cEmHgosPbyLTcY732/5nmiXffHzZtmHSEleUIKWfXzbZ3AknDGtEbmhZd8
eCH63Y7QYpxY39g/mBpmu1H0+mTjeQzIVeHkHgLbpgEnpxbWxiwz8hoch3w4oUuCrmeYYwOZ
NUzQF2JBfHmHQL476pVjCMGCX7LvFC4jNuuVbnzdL1DBmZxhDv6zWig7GYhgMY4eqgzE8FFg
JWO/LiiRUFWO+LwLG4HDxVwYUnlJQ5HPGRT6bRCSqozPpLjQAXKpVgVvUpvn6XalBvfFzYNh
SiU1IFfbDsShGIBtBU8bGY7hVBk3VWcGu06LQJ9GvOOqGBVAejhVSCtO5SA9rBorA7cKRxtJDm
J02qwtX+iAWYGiWcyOmrVIF1edSvBrQLROatC29NO1RjqzgNs6JcmXoEzFIRBGOpVOEM7oc6F
6AzVZmgSqs7QoyvCHTQptjaBwN8dk2LPEtWEVKBG4LTjlisHWzcoG35OcQ+UliOPulflQk2u1Zu
qVYNHLqBwaN7J+usvpJXr3zKGCZEVg8i81CXCGjkOlmeMJMhbMCrgDDgFUe90iRFWLSqd6R
G05Jgc06FpmeN0qxGMLmfTMMldig6BlthMldqg7B8FBgFWC9m9kkVJ1gR1ecVWBxNwsEob4rb
mdRTUgFakRtWCZdxUcqQDvwBitApxyqALVIK0DGxitALQuENjlcQygTk0u1VlkFeKQxVAEeTg1U
gDqn52HVWAWoK/hsJDmlz3mzcnwOmBXwCczCK0CT4TN61a0ATYWfc1aFtmedMIV+ct6rAA9F
h58xDmBPQ9XhJx7qXIP4QAUIQ9X5eXQFUgFGQbsCxQhFajBK0CmzbqxoYPsrSehQCCOH/Z
u44vM7tYYUakAfS4jB7799Q8c+DYyZZgLMFoBGIUgrJXjIMLmDdtShl145REGvclPVRudlAxY1as
ATXlqedlpT7BpoyipnVp2UwuUgJVjLF6BH2MBYZBjLF6FH2PphALLQFIC1o6xJKGqx1h8X4QSSD
y1DATNU8uoxqeSaKqnlto+ogM1IFNq5bw8qlKhotofRx44HUMJS4pAtm07BoeKQEp4gdFGjkMYZ
dr6O/eoBRFpGejzGDIK480aOEpDSc7Qw62xOpCSCkUbaY5R9Cl+5Rgd8CtgFPIFn6ahJONotKtb
CIJSlemcW77xebNojaRuhoe7yTpJvQlnKQiDkNSrcJJ2QoHIKG5INEiahKqT9OiL9oEalGgeqEE1l
RWoqR6oifAR2XZga4eHUvwBkD0Q3A58xG4gc9uXk+Os9gSlz2XgXlu//P65Fpo//Mek2CfWGMcqj
/81khzE2BMcy040D5gVKCay7cC2WemRZuBVH2K0PNM8Z1Voe96p2qFmpuSql8P2CZYx7FAkN
7TFI6cgTLwLFQXDDIVyQ1t8xdoJdS4EZ6jzhrb4ljYJVTboFCLZECz2A8HnrdMtmCTklfLdQGQzkN
KRB0EYg90N3JVDu4GUpc+CMP6l7UDKiodBWjmOQZTPbtXQlj0O4vMY2RD0Zg1sCNL8ITrv1iB
BK0/RtdlcROgT/MoROuBXYCgb3ROk2aN0wK4+RCvP0k26FRqfN6tKUZFSc9YqQZFtC1YrQZHv
5VUrQZFtC1YrQTzUuRJI24LVSiDk24JfJcg624JA0NwWRDUhFTa8LUjD7dX5xfDWehULZfgDIns
oNntSkNnBbftpZqzx2iMiPpuho4LeBKBtm8Czo86MmHXTo4cFKS8POzczHSTaE4zLzjsPeeARbj3j
Pc/SQ8/Asv6BQcrLY8+zjvnmn2BY7eSzm234TqFXdKjGRnYKvapDNTwUWBq6O4VJqCrVfGcgR
waBon1kEBf5bBJR/chgxJG9/Yhks6Wlbo6+cPwl9B7pBFseqfN6LqqNA0M+2n64isu7+w3lo+7D7
+7/7i3wL1tvaBrMgVDV0qaOQy8PC88rsLt6mfXNyKaFure6/PWjYhmrOaUIZK7bQyt95eNTc3Z81C
iGz/aTlzsNs4r8Ns4EAa5jfMq/DauE+oE3hmqfRuXhKrexvIRtF9+HEPwwqJgfw+XKOrNOCKJGZ1e
An/d6R5penpZfWVoHy+kdFPq7o2daf9793C/vxnpk+YEm0vCSNuJzSxGhmd86tT1D+8Nz0PRGZ
4xTG1MuYal6QwEYkYHQoSEHQcSY4TovZn0RIS9tjxSZxwYR7gSEA6z/3UvUXr3y/Ly7oN7wVK
TsZPte8Y2EhgaAhwMAIIFBkhFEGxsr+yb7/avaF4rfFDxmIW+ZKPf+y9ZEx0Kh6QgksyzSJqQIP1
qmhEVbFc1IZiu9R27/Xcaw2sfyvNJO9Y2xdUWyRtZL3BtFgDmXumX+8tbmq11TQn1st96WPLG/c
+51c/3d2/tSvLy5+XMJD+D3ddh8sNCmVuZHN0cmVhbQ0KZW5kb2JqDQo1IDAgb2JqDQo8PC9
UeXBIL0ZvbnQvU3VidHlwZS9UcnVlVHlwZS9OYW1lL0YxL0Jhc2VGb250L0FyaWFsLUJvbGRNV
C9FbmNvZGluZy9XaW5BbnNpRW5jb2RpbmVudERlc2NyaXB0b3IgNiAwIFlvRmlyc3RDdGFy
yIDMyL0xhc3RDdGFyIDEyMS9XaWR0aHMgMjAxIDAgUj4+DQplbmRvYmoNCjYgMCEvYmoNCj

[illegible]

UUUUAAAAFABRRRQAUUUUAAAAFABRRRQAUUUUAAAAFABRRRQAUUUUAAAAFABRRRQA
AUhGaWigDx34qeDfLLa9YRHB/4+kUdP8Aa/xryivrSeGO4heKVQ8brggjgivnXx14Tl8M6w5jQmxn
JaFh2/2fwrzsVRt76Pt+Hc154/Vqr1W3+RyIFFFFcZ9aFFFFFABRRRQAUUUUAAAAFABRRRQA
UUAAAAFABRRRQAUUUUAAAAFABRRRQAUUUUAFHajtQtxT+Fn1LoP/IA0/wD690/IRRoP/IA0/
wD690/IRXtx2R+RVf4kvU1aKKKoxCiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACii
igAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKA
CiiigAooooAKKKKACiiigAooooAKKKKACiiigBKxfE/h638SaLNYT/KxGY5AOUbsa26Q0mk1ZI06kq
clOL1R8p6rplzo+pz2F2m2aFtp9/Q/jVKvfPiP4NXX9NN9aIP7QtIJGP417rXgrKUcqwLZTgg9q8mtSd
OR+mZTmMcbRUvtLcbRRRWJ6oUUUUAAAAFABRRRQAUUUUAAAAFABRRRQAUUUUAAAAFABRRRQA
ABRRRQAUUUUAAAAFABR2oo7ULcU/hZ9S6D/yANP/AOvdP5UUaD/yANP/AOvdP5UV7cdkfkV
X+JL1NWiiiQMQooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAo
oooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiig
AooooAKKKKACiiigAooooAKKKKA EYZGK8S+KPg3+z7htbsY/9Hmb9+qj7jev0Ne21XvbOC/tJbW4
jDwyKVZSOoNZ1aaqRszuy7HTwVdVI7dfQ+TqK6Hxj4Zm8Ma5LbFWNq5LW8h/iX0+orn+1ePKLi7
M/UMPXhXpqpB6MSiiiKbRRRRQAUUUUAAAAFABRRRQAUUUUAAAAFABRRRQAUUUUAAAAFABRRRQA
ABRRRQAUDqKO1C3FP4WfUug/wDIA0//AK90/IRRoP8AyANP/wCvdP5UV7cdkfkVX+JL1NWiiiQ
MQooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKA
CiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKK
ACiiigAooooAKKKKACiiigDnvF3hq38TaLLayALMvzQyf3W7V8331jcabfTWd0myaFtrLX1hXm3xQ
8G/2nZnWLGln3AP3iqOZF/wARXJiaPMuZbn0mQZp9XqewqP3X+DPD6KXpxjHrSV5p+gJ3Ciiig
AooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAo7UudqFuKfws+pdB/5AGn/APXun8
qKNB/5AGn/APXun8qK9uOyPyKr/El6mpRmuN8e+MLjwbWclvbJOZ3ZSHbGMCuH/4XRqX/AEC
4P+/p/wAKznXhB2Z24bKMxiaaq0o3T8z2rNGa8V/4XRqP/QLg/wC/h/wo/wCF0aj/ANaUD/v4f8Kn6
zT7nR/q9j/5fxR7VmJNeK/8Lo1H/oFwf9/T/hR/wujUf+gXB/39P+FH1mn3D/V7H/y/ij2rNGa8V/4XRqP
/AEC4P+/p/wAKP+F0aj/OC4P+/h/wo+s0+4f6vY/+X8Ue1ZozXiv/AAujUf8AoFwf9/D/AIUf8Lo1L/oFQ
f8Afw/4UfWafCP9Xsf/AC/ij2rNGa8V/wCF0al/OCOP+/h/wo/4XRqP/QLg/wC/h/wo+s0+4f6vY/8AI/FHt
WaM14r/AMLo1H/oFwf9/D/hR/wujUv+gXB/38P+FH1mn3D/AFex/wDL+KPas0ZrxX/hdGo/9AuD/v4
f8KP+F0aj/wBAuD/v4f8ACj6zT7h/q9j/AOX8Ue1ZozXiv/C6NR/6BcH/AH8P+FH/AAujUf8AoFwf9/D/
AIUfWafCP9Xsf/L+KPas0ZrxX/hdGo/9AuD/AL+H/Cj/AIXRqP8A0C4P+/h/wo+s0+4f6vY/+X8Ue1Zo
zXiv/C6NR/6BcH/f0/4Uf8Lo1L/oFQf9/D/hR9Zp9w/1ex/8v4o9qzRXiv8AwujUf+gVB/38P+FH/C6NR/
6BcH/fw/4UfWafCP8AV/H/AMv4o9qzRmvFf+F0al/OCOP+/h/wo/4XRqX/AECOP+/h/wAKPrNPuH+r2
P8A5fxR7VmJNeK/8Lo1H/oFwf8Afw/4Uf8AC6NR/wCgXB/38P8AhR9Zp9w/1ex/8v4o9qzRmvFf+F
0aj/OC4P8Av4f8KP8AhdGo/wDQLg/7+H/Cj6zT7h/q9j/5fxR7VmJNeK/8Lo1H/oFwf9/T/hR/wujUf+gX
B/38P+FH1mn3D/V7H/y/ij2rNGa8V/4XRqP/AEC4P+/h/wAKP+F0aj/OC4P+/h/wo+s0+4f6vY/+X8U
e1ZozXiv/AAujUf8AoFwf9/D/AIUf8Lo1H/oFwf8Afw/4UfWafCP9Xsf/AC/ij2rNGa8V/wCF0aj/ANaUD/
v6f8KP+F0aj/OC4P8Av6f8KPrNPuH+r2P/AJfxR7VmJNeK/wDC6NR/6BcH/fw/4Uf8Lo1L/oFQf9/D/h
R9Zp9w/wBXsf8Ay/ij2rNGa8V/4XRqP/QLg/7+H/Cj/hdGo/8AQLg/7+H/AAo+s0+4f6vY/wDI/FHtWa
M14r/wujUf+gXB/wB/D/hR/wALo1H/AKBcH/fw/wCFH1mn3D/V7H/y/ij2rNGa8V/4XRqP/QLg/wC/h/
wo/wCF0aj/ANaUD/v4f8KPrNPuH+r2P/I/FHtWaM14r/wujUf+gXB/38P+FH/C6NR/6BcH/fw/4UfWaf
CP9Xsf/L+KPas0ZrxX/hdGo/8AQLg/7+H/AAo/4XRqP/QLg/7+H/Cj6zT7h/qj/5PxR7VmJNeK/8AC6
NR/wCgXB/38P8AhR/wujUv+gVB/wB/D/hR9Zp9w/1ex/8AL+KPas0ZrxX/AIXRqP8A0C4P+/p/wo/4
XRqP/QLg/wC/h/wo+s0+4f6vY/8AI/FHtWaM14r/AMLo1L/oFQf9/D/hR/wujUf+gXB/38P+FH1mn3D/

AFex/wDL+KPas0ZrxX/hdGo/9AuD/v4f8KP+F0aj/wBAuD/v6f8ACj6zT7h/q9j/AOX8Ue1ZozXiv/C6
NR/6BcH/AH8P+FH/AAujUf8AoFwf9/D/AlUfWafcP9Xsf/L+KPas0ZrxX/hdGo/9AuD/AL+H/Cj/AlXR
qP8A0C4P+/h/wo+s0+4f6vY/+X8Ue1ZozXiv/C6NR/6BcH/fw/4Uf8Lo1H/oFwf9/D/hR9Zp9w/1ex/8v
4o9qzRXiv8AwujUf+gVB/39P+FH/C6NR/6BcH/fw/4UfWafcP8AV/H/AMv4o9qzRmvFf+F0al/0CoP+
/h/wo/4XRqP/AEC4P+/h/wAKPrNPuH+r2P8A5fxR7VmkYBIII4PUV4t/wujUv+gVB/38P+FH/C6NRz/
yC4P+/p/wo+s0+4f6v49a8v4oo/EvwYdEvm1WyT/Qbh/nUf8ALNz/AENef19I6V9p8V+FW/tywjgW6
U4hByQvYn0NeD+KPD8/hvXJrGYMY87onl++vauTEUre/HZn0uR5i6ieFrP34/iYtFFFcp9IFFFFAB
RRRQAUUUUAAAAFABRRRQAUUUUAAAAFABRRRQAUUUUAAFHaijtQtxT+Fn1LoP/IA0/8A690/I
RRoP/IA0/8A690/IRXtx2R+RVf4kvU89+Nf/HjpX/XV/wCVeOdq9j+Nf/HjpX/XVv5V452rzMV/EZ+hc
Of7hH1f5hRRRXOe4FFFFABRRRQAUUUUAAAAFABRRRQAUUUUAAAAFABRRRQAUUUUAA
FFFFABRRRQAUUUUAAAAFABRRRQAUUUUAAAAFABRRRQAUUUUAA
FFFFABRRRQAUUUUAAAAFABRRRQAUUUUAAAAFABRRRQAUUUUAA
FFFFABRRRQAUUUUAAAAFABRRRQAUUUUAAAAFABRRRQAUV3/w18
GnW9RXUryPNjbtIVYf6xv8BXMeGvD9x4k1iKxgBCk5lkx9xe5r6U0rTbfSNNgsrVAsUS7R7+9dWG
o8z5nsfNcQZr7Cn7Cm/ef4ltqgVQqjAA4Fcx448KReJ9EeJVUXkQLQSEdD6fQ11NGK9GUVJWZ8
LRrTo1FUg9UfJdxby2tzJbzoUljYqysOQRUVey/FLwX9phbXdpj/AH0Y/wBIQD7y/wB76ivGq8irTdO
Vj9Py3HwxtBVFv1XmFFFFZnoBRRRQAUUUUAAAAFABRRRQAUUUUAAAAFABRRRQAUUUU
AAFHaijtQtxT+Fn1LoP/ACANP/690/IRRoP/ACANP/690/IRXtx2R+RVf4kvU89+Nf8Ax46V/wBdX/IXjn
avY/jX/wAeOlf9dX/IXjnavMxX8Rn6Fw5/uEfn+YUUUVznuBRRRQAUUUUAAAAFABRRRQAAtJSgF
mCqCSTgADrXTaR4A8RawVaKyMER/5aTnaPy6/pTjCUtjCviaVBc1WSRzFFafiDRpPD+tz6ZLKJ
Xh25dRgHIB6fjWbSaadmaUqkasFOGzEooooLCiiigAooooAKKKKACiiigAooooAKKKKACiiigAoooo
AKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAoo
ooAKKKKAFp8EEtzPHBcheWRgqKOPJqOvYfhX4N8qMa9fxfvHH+jlw6D+9+NaUqbqSsjz8yx8MF
QdSW/T1Ot8DeE4fDGjKrKGvZgGnKx39PoK6uiivXjFRVkfMfetOtUdSbu2LRRRVGQx41kRkdQVY
YIPevn/4i+D38Pas13blf7PuWJXA4Ru4r6CrN1zRrXXdKnsLtcxyrjPdT2lrGtSVSNj08rzCWCrf2Xuf
LFFaevaJc+HtYm066HzxnKvjh17EVmV5Mk07M/TaVWNWCnB3TCiiikaBRRRQAUUUUAAAAFA
BRRRQAUUUUAAAAFABRRRQAUDqKO1C3FP4WfUug/8AIA0//r3T+VFGg/8AIA0//r3T+VFe3HZH
5FV/iS9Tz341/wDHjpX/AF1f+VeOdq9j+Nf/AB46V/11f+VeOdq8zFfxGfoXDn+4R+f5hRRRXOe4FF
FFAXsFFdDpXgnxBrg1rbTpFjb/AJaS/lv+Nbi/CHxKwzvsR7GVv/ia0VGB2RwVc0wlJ2nUVzgqK7K
8+F/ieziLi2inx2hkyf1xXK3dldWE5hu7eSCQfwyLipTIHdGtDG4ev8Aw5plelpK09L8P6rrSyNptk9wlz
htpAwfxNSk27I3qVIU4803ZGt8PEV/HOnK6hl3NwR/smvo9QFACgAe1eGeBfCGv6d4wsbu702W
GCMtudiub8p9DXunavTwsWoan5/xHXhVxKdOV1Y+c/iZ/wAj/qX1j/8AQFrk69G8e+Edf1Lxnf3dnps
s1vJs2OrLg4QDufWuZk8D+Joo2kk0iZUQbmJZeB+dcdWnJzbsfV5djCpHC04uavZdTnqKWkrA9i4
UVf03RdS1iXy9Ps5ZznkqOB9SeK6+0+EXiKfa0z2sCkc5clh+GP61caU5bl46+YYahpUmkcF2pK9
HI+DetLGTFe2rt2U7gD+NYGq/D/xHpALy2JmjAyXgO8D8Ov6VT01FujKlm2Dqu0aiOXopSCpKsC
CDggikrI9FNPVBRS9q3lfB3iGeFJotKnaN xuVgRyPzpqLexIVr06X8SSXqYVFawpaBquixpJqFIJbo
52qXxyfzqnbWtxeSiK1gkmkP8Ma7jRytOwRr05Q51JWlaK6OPwF4plXcmjzY92Uf1rl1LSr3R7s2uo
W7QTgBtjEE4P0NNwklDoini6FWXJCSb9SnRWHPmi6nrBkGn2b3HI437McZ6davN4L8SlpZtJnCG
ZJ4/wAaFCTV0gni6EJcsppP1MGilxzjHNBnJ4S1/UU32uIXDqehIC5/PFJRb2RdSvSpK85JGLS1t6j
4Q1/SLJ7y/wBNkgt0IDozKcZOB0NYdDi46MKVenWXNTkmvIKKciNI6pGpZmOAqjJP0rq9M+HHI
XU4xltmtuh6G4bbn8BmnGepfCiK+Lo0FerJI5KivSU+DWsIFL31qrEcgbjiqF78J/ElqrtELA4Ucgl5D
H8CMfrWn1ep2OOOdYGTsqiOFoq3f6XfaXN5N9ay27+jrwfoe9VKxaaep6UKkZrmi7oKKKKCgpaS
tXTvDetasu6x06eVf72MA/QmmouWxnUrU6SvN2RIUV2Q+GHigqCbOMcZwZORWNqPhXXdJQy

XumTxoM5YDcAPfGcVTpTS1RzwcDC1Hyxmm/UxqKlt7eW6ul4IELzSMFRR1Jrb/wCEJ8S/9Aif9P
8AGkoSeyNamJo0nackjApKsXtlc6ddyWt3C0U8eNyN1GRn+VQVO2jNoyUlzR2EooooGFFFFABR
RRQAUUUUAAAAFABRRRQAUUUVp6Fol14g1aHT7Rfmc/O+MhF7k01Ft2RFWpGIBzm7JHQfD3w
g/iPVhPcRn+z7cgyNjhm/u19BxRrDEsaKFVRgADoKoaHo1toOkw2FqgVl15OOWPcmtKvWo0ITjY/
Mc1zGWNrOX2VsLRRRWx5gUUUUAAAAFAHGfEDwhH4k0hpYUAv4AWibu3+zXz3JG8MrRyKV
kQkMpHQ19bGvH/in4NKM2vWEXB/4+UUf+Pf41x4qjzLnR9Vw9mvspfvqr0e3keTUUUUV5x90FFF
FABRRRQAUUUUAAAAFABRRRQAUUUUAAAAFABR2oo7ULcU/hZ9S6D/wAgDT/+vdP5UUaD/
wAgDT/+vdP5UV7cdkfkVX+JL1PPfjX/AMeOl8AXV/5V452r2P41/8AHjpX/XV/5V452rzmV/EZ+hcO
f7hH5/mFFFFc57gtezfD34e28NnFq2rQiS4kAeKJhwg7E+9ea+DdNXVvFun2jgFGk3OCOoHJH6V
9NKOvQAMADpXbhKSI7zPkuJcxnSth6btfViKqxqAAFAqJ761jba88at6Fq8k+KPjO+h1RtE0+4eBII
BneNirEnkDP0x+deVli7FmYlieST1rWpilB8qR5uA4dqYqkqs5WvsfWquki5VgwPoayNe8Nab4hs2g
vbdWOPIkAwyn1Br5/8L+KtR8OalDJBPI1tu/eQFjsYHrx2NfSsMqzQpKpyrqGB+taU6kayODMMvrZ
ZVi1LfZo+YvEugz+Gtbm06c7tvzRv/fQ9D+lekfBYj7Lqecf6xeV0o+NVmn2XTb0ffEjRH3BGf6V5Ra6
he2QYWI5cW4Y/N5MrJn8q4najWufWQU82yxK9pP9GfV2RnqKdXz98P8AWNUnufG2nzw6leSxM
W3JJOzKfIPYmvoHtXfSqgoro+NzHL5YGqqcnfS43K+oqlrBH9jXvl/1D/wAjXh/xF1jU7bxzqMNvqV5
FEuzakc7Ko+Qdga5Vtd1d1Ktq1+VlwQbh8H9awnioxbjY9fCcOVatOFZTVnZIA9T9a6Hwb4Zk8Ua4
trkrbxjfm47L6fjXPV7d8HLFlvDlxeYG+acqT7CuShBTqWZ9NnGLihMG5R32R3mmaTZaRaJbWVu
kUajACim6hrOnaSga/vlbcHgeY4GaTXNSXSNFvL9hkQRM+PXArsi1HUrvVr+W8vJWklkbJJPT2Ht
XfWrKkrl+NyrKp5jOU5ysl1PpG18YeH7ydYYNVtXkY4CiQZNbfyuvYg18kV698I/E1zcSzaJdzNKET
zINxyQBwR/Ks6WK55crR2Znw99VpOtSldLc1PiF4BtdSsZdT06FYr6JS7KgwJR6fWvDq+tnUMjKR
kEYr5f8UWq2XinUrdBhUnbA9M8/1rPF00rSR38MY6pUUqE3e2qMivqPwyd3hrTmlHNun8q+XO1f
UXhf/kWNN/690/kKMFuw4r/h0/VnM/ErRJ9fh0ewtl+eW8wWx91djEn9K6Hw94X03w7ZLBAqrux80j
D5mPvW0VGQxAyOhrhfFXxM0/QLhrO2jN3dKfnCnCp9TXXJQg+eR81RlisTTjhaN2l/Wp3fFeAfFn/
keJP+uEf9auSfGPXTIxS1s1XPAKsf61yfibxBL41b+0ZolikMSoyqeMjP+NcmIrwnC0T6PJMoxWEx
XtKq0s+p6B8E/8AXax9lv8A2avWL5R9hn4/5Zn+VeT/AAT/ANbq/wBlv/Zq9fIDAg8g9a6cOv3SPEz
yVswm/T8jy/4f/D63it01jVoRJPL88MTjhAehI9a9OVFRcKoAHYCSnX/EWn+GrA3N9IFHRIx95z6AV
5dqHxl1B5cafYQxx+sxLE/lilz06KsCw2OzSTqJXX4HbfFT/kQb0/7cX/oYr57UMxCqCSTgAd67PXPi
PqXiHQrjTL22gCylSHjyCMNnnP0rO8C2K6h4z02FI3lsnmNn/ZGR+uK460IVqLIPqsqoVcuwVR1lqr
v8D1nwF4EtdE06K9vYVvK1GVQzbhny8/wAlruXdlloyzsFUdSaeOAB6V4n8WvEdzPrH9iwyMltCoaR
VP32Pr9K7ZONGGh8jQpV81xdpPV6+iPTZPG3huKUxPrFoGBwf3g4rWstQs9RgE1ncxzxt0aNgR
Xyh2rW8P6/e+HdUivLSRgFb95Hu+Vx6GueOMd9Ue7iOFIGm3Snqu59I6volhrdm9te26ylwxkjkfQ1
86+LPDkvhnXJLJjuil3xP6rX0rZ3C3dnDcJysiBh+lrzH4z2Ktp2n3oADJIUJx1BH/ANatcTTUocx5+Q
Y2pQxSoN+69LeZ45SUV2Pw48OjXvEyNMm61tcSSAajnsK86EXKXKj7rFYiOGoyqy2R1PgH4bR
zQRatrcRO75orZvT1b/CvWoolreMRxRqiKMAKMCnqAqgAcDpXk3xF+IVzaXj6Po8vlsngxPMvUH0F
ep7ICB+dc2KzfE2v/AJJHrBljBxvXP1prxxTxIHVXQjBBGRXyhPcz3Mvm3E8ksh/jkcsfzNb+geN9a0
C5VoruWaAH5oZnLKR7Z6fhWKxkW7NHqVOFq8lc0Jpv7j0/Vvh5Db+KNM1nSltpdo1xCvTG7lhX
ogAxWR4b8Q2viXSI762OM8Oh6o3cVs11QjFK8ep89iqtaUICtvHQ+c/iX/yUDVPrH/6LWuSrrfiZ/yU
DVPrH/6LWuSryav8Rn6Zlf8AudL/AAoKKKKzO4KKKKACiigAooooAKKKKKACiigByl0jrGilmY4AHc
mvoP4e+EF8OaQs065vrgBpD/dHZfwrj/hb4L8+Vde1CLMaHNqjDqf73+H5+leyYr0MLRsudnw3EO
a+1l9WpPRbi0UUUV2nygUUUUAAAAFABRRRQAIRXEEdxC8MqBo3BVIPQipaKATad0fOXjzwnJ4
Z1ljGpNjcEtC3p6r+FcpX1D4j0K38Q6NNYXCj5x8jY5RuxFfNer6Vc6Lqk1hdtlibGezD1FeXiKPI7rY/
RMizRYql7Ko/fj+KKNFFFcx9AAAAFABRRRQAUUUUAAAAFABRRRQAUUUUAFHajjtQxtT+Fn1L

oP/ACANP/690/IRRoP/ACANP/690/IRXtx2R+RVf4kvU89+Nf8Ax46V/wBdX/IXjnavY/jX/wAeOlf9dX
/IXjnavMxX8Rn6Fw5/uEfn+YUUUVznuHXfDN1Tx7p5bod4/wDHTX0XXyjpL/Lpmp219D9+CQOPwr
6a0HW7TXtLhvbWQMmqNy55U9wa9DBzXLynw3FGGmq0a1tLWPJPin4Uvo9dk1q2hea1uFBkKjJ
RgMcyj0wBXmx619bMquuGAI9CKwNT8EeHtW3tdaZB5j9ZEXYx/Ec0VcLzPmixZbxH9XpKIWjdLqj
5ors7b4o+JLW2it43tflJqLuYiTgDHrXY6r8GrSQM+mXrwt2ST5IH9a8/13wNrugBpLi1MkAP+ti+Yfj6
VzunVpao92OOy3MbRnZvsyPxB4z1bxNbQ2+omExxPvXy0wc4I9feueoorCUnJ3Z7FChTow5KSsj
qvvhx/yPem/7zf+gmvo8V84fDj/AJHvTf8Aeb/0E19HivRwfwHwvFH+9r0PnP4l/wDI/wCpfVP/AEBa5L
vXW/Ev/kf9S+qf+gLXJVw1vjZ9jln+50vRBXR6T4413RLFbKxuEjgUkgFAeTXOVZsbC61O8jtLOFp
ZpDhVUVMXJP3TfE0qNSH75Jpdze1Hx94h1Sxls7q6RoZRtcBMZFcxXsnh74QWscTa3K00hGT
DGSqj8Rya7e08G+HbLBh0m0DAY3GME/nXUsNUnrNnzcs+wOEvDDwv6al+ZcH0Ndx8KCV8cRj
pmBwf0r3P+ydPAx9jg/74FPi06yhlWWK2iSRRwyoARWkMJyyUrnDi+JFiKEqXs7XVty0elfMvjf/AJH
XVv8Aruf5Cvpo9K+ZfG//ACOurf8AXc/yFPGfAjPhb/eZen6mBX1F4X/5FjTf+vdP5Cv12vqPwv8A8ix
pv/Xun8hWeD3Z38V/w6fqyHxhq50PwvfXyECREwmf7x4H6mvmWSR5pWlkYu7nczE8k17z8XHZf
BbqOjzID+ea8DqcZL3kjbhajFYeVTq2FFFFch9Qes/BP/W6v9lv/Zq9gJwDXj/wT/1ur/SL/wBmr16XiJ
j7GvWw/wDCR+Z57/yMJ/L8kfO3xH1yTWfFtygcm3tGMMa9gR94/n/KuRq3qc/2rVbu4GcSTM3Pua
qV5lSTIJs/QsFRjRw8IR6lWruk6td6Lf2LhJ1UqGlz1qjRUJtO6Oic1zi4yV0dj/AMLO8U/8/if9+XxM6l
qVzq1/Je3kgeeQ5ZgMV1vhL4cX/iJFurpmtLLsxHzOPb2r1LTvht4Z09MHT0uG7tP8+fwPSuqNGrV
XvPQ+br5nl2XzapRvLy/zPnbBowfQ19T6FpVugSHT7ZFAwFWMAVldJ0/obOD/vgVp9Sfc5HxXH
/n3+JQ8HsW8H6ST1+yR9f90Vynxk/5Fi3/AOvkfyNeiRRRwxrHGqoijCqowAK86+Mn/ls2/wD18D+R
rpqq1Jo8DLJKeYQl3Z4dXuXwe04QeG57wqQ9xMevoK8Nr6O+HEYTWlpuP4kJP51xYRXnc+s4n
qOOEUv1Zu6xerpuj3d4xwsMTOT9BXyzcXD3V1LcSnMkrl2J9TX0Z8Q3MfgbVCO8W386+bqvGN
3SObhSkITnU63sFFFFcR9aeifCHVntPEsmns37q7jJC/7S8/yzXutfN3w8cx+OdNI7uR+YNfSPavTw
jvA/POJaUYYzmXVHzn8TP+Sgap9Y/wD0WtclXW/Ez/koGqfWP/0WtclXBW+Nn2uV/wC50v8ACgo
oorM7gooooAKKKKACiiigAooooAWup8DeE5PFGsBZARZQENM3r/s/jWFpemXOsaldY2iFpZWw
Mdh3NfSnhrQLbw7o8NjbqMqMu+OWbua6cNR53d7Hz+e5osLS9nB+/L8DSt7eK1t44IUCRou1VH
QCphRiivUPzttt3YtFFFABRRRQAUUUUAFFFFABRRRQAld8SfBw17TTf2if6fbKSAo/wBYvcfX0r
vqCMipnBTVmb4bETw1VVYPVHyOQVJVhhgcEHtRXpvxR8GCwuG1uwiAt5T+/RR91v734968xrx
6lNwZn6hgcbTxdFVYhRRRUWOzmXcKKKKLBzLuFFFFFg5l3Ciiiwcy7hRRRRYOZdwooooooSHMu
4UdqKkaWpM5Llep9S6D/wAgDT/+vdP5UuAf/wAgDT/+vdP5Civajs8lq/HL1PPfjX/AMeOlf8AXV/5
V452r2P41/8AHjpX/XV/5V452rzMV/EZ+g8Of7hH5/mFFFFc57gVq6J4i1Pw9defp9wUz96NuVb6io
9J0PUdbkkj0+2aZoxlgvatX/hX/ib/AKBktXCM94nFia+Ed6VeS9Gd7o/xltnVU1axeFuhkhO5fqR1H0
5rt9M8Z6BqwX7LqUJZuiudrfkea+arm3ltLqW2mXbLE5R19CDg1GCVOVOCO4roji5x0keNX4awtV
c1JtX+aPrZZEcAqwYH0NI8aSoVdQyYIlr5n0TxfrWgzo9reyGJTkWucqwr6M0XUo9Y0a11CMfLP
GHx6etddKvGrslL5lINXANOTun1PFPib4Pi0C/TUbFNtldNgp2jfrx7HmuAr6J+Jlmt34HvifvRASKfTB
zXztXDiYKM9D7Hh7FzxGEtPVxdjqvvhx/yPem/7zf+gmvo8V84fDj/AJHvTf8Aeb/0E19H9q6sH8B87x
R/va9D5z+Jf/l/6l9U/wDQFrkq634l/wDI/wCpf9s//QFrku9cNb42fY5Z/udL0QvXgda98+GvhOLRdFjv
50BvbpQ5JH3FPQV4noVqt9r9hauMrLOin6Zr6ljUJGqqAAAABXTg4JtyZ8/xRi5QjGhF76sq6jqVpp
NIJd3kyxQoMljXlusfGVhM8ekWAZB92adsZ/4CP8aq/GXVZZNWs9KUsIY4vPYdmYkgflg/nXI9OviJ
RlyxM8lyOjVoqvX1v0O6f4s+Jnzzarn+7GeP1rpvAPjzW/EHiVLG+aAw+UznahByMe/vXj9eh/B+0eb
xXNcj7kMBDfViMfyNZUatSU0mz0c0y7B0cHocYJNi92PSvmbxt/yOurf9dz/IV9MnpXzN42/5HXVv+
u5/kK6MZ8CPD4W/3mXp+pz9fUXhf/kWNN/690/kK+Xa+ovC/wDyLGm/9e6fyFZ4PdnfxX/Dp+rOX
+L3/lnH/run868Fr3r4vf8AlnH/AK7p/OvBajF/xDs4Y/3N+rCiiiuU+iPWfgn/AK3V/pF/7NXr03+pf/dNe

Q/BP/W6v9lv/Zq9em/1L/7pr1sP/CR+aZ5/yMJ/L8kfJc3+vk/3jTKfN/r5P940yvKe5+j0v4a9EFdh8PP
C6+JNf3XCk2drh5B/ePZa4+vdvhBZLB4Ue4KYeeZiSR1A4FbYeHPPU8rPcXLDYRuG700/jjSGJY
41CoowABwBXE+KfiZpnh+4e0t0a8vF+8inCqfc/wCGa6HxXqD6X4Y1C8i/1kcLFfrivmJ5HlkaSRi7s
ckk8kmuzEVnT0R8tkeUwxrlUrfCvxO/uvi/r80hMEFpCnZSpY/nkVVHxX8TBid9sfbyz/jXD0oBJAHO
TXF7eo3ufXf2RgYR/ho+pfD97LqXh7T72cL5s9ukj7RgZIBNcX8ZP+RYt/8Ar4H8jXcaNbi00Wzt1UK
l4VUAdsCuH+Mn/lsW/wD18D+Rr0av8JnwmW2/tGFtuY8Nr6Q+HTBvA2mYPSMj9TXzf2r3n4R332
nwibcuC9vMykegPlrjwb99n1PFEG8LGS6M2PilhfwNqYHaLP5V83V9V6zZLqWj3dmwys0TIQfcV8
s3EElrcy28qlZImKMCO4qsYtUzDhSqVZzp9b3lqKKK4j606r4cx+Z4704Y6Mx/JTX0fXh3wf0lrnxBN
qTJ+7to9qsR/E3/ANbP517hXp4RNU9T874lqqeNsuiPnT4mf8lA1T6x/wDota5Kut+Jn/JQNU+sf/ota
5KuCt8bPtsr/wBzpf4UFFFFZncFFFFABRRRQAUUUUAFKAWIVQSxOAAOtHavSfhf4N/tK8XWr6
M/ZoG/cqw++3r+FXtg5ysjx2MhhKLqzOw+Gvg06Dp32+9Qfb7kZ2kf6pew+vr/wDWrv6AABgdqK9
iEFCNkfl+KxM8TVdWe7FoooqjnCiigAooooAKKKKACiigAooooAKKKKAlbi3huoHgnjWSNxlYZB
FZX/CJaD/0C7b/v2K2qKTinuaQq1IK0W0Yv/CJaD/0C7b/vgUf8lloP/QLtv++BW1RS5l9ivrNb+d/eY
v8AwiWg/wDQLtv+/Yo/4RLQf+gXbf8AfsVtUUCKewfWa387+8xf+ES0H/oF23/fsUf8lloP/QLtv+/Yrao
o5l9g+s1v5395i/8ACJaD/wBAu2/79ij/AIRLQf8AoF23/fsVtUUCKewfWa387+8xf+ES0H/oF23/AH7F
H/CJaD/0C7b/AL9itqijkj2D6zW/nf3mL/wiWg/9Au2/79ij/hEtB/6Bdt/37FbVFHJHsH1mt/O/vMX/AIRL
Qf8AoF23/fsUf8lloP8A0C7b/v2K26KOSPYPnNb+d/eRxxJDEscY2ogwoA6Cin0VRIdnlPxr/wCPHSv
+ur/yrxztXsfxr/48dK/66v8AyrxztXIYr+lz9H4c/wBwj8/zCiiuc9w9D+D92sHiqu3d8CeAhV9WBB/lmvd
cCvlfRNVm0TWbXUYPvwwux6juPyr6Z0bV7XWtMhvbSQNHlueD0Poa9HCTTjynwXE2EnDEKutpf
meG/E3QJ9L8Uz3giP2W8Pml4HAb+IH3zz+NcRX1dqGm2eqWrW17Ak0LdVcZrj5/hN4bmmZ1inj
B/hSQ4FRVwrcrxOrLul6dKiqVdO66o8GggluriOCBGeWRtqqo5Jr6e8Naadl8OWNg3DQxAMM9+9
UdD8D6FoEiy2doDMM4lkO5h+NdC7rGhZ2AUDJJ7VtQo+z1Z5mc5sse4wpq0UcX8U9bPwVcR7
gJLhljUHvzz+ma+fa7f4keLE8R6uttZvusbQkKwPDt3P09DXEVxYmanPQ+ryDCSw2EXPvLU6n4d
MF8d6aSeCzD/x019ldq+W/DWoDS/Emn3jNhlpLH2zzX1BFIssSylQVYAgg11YNrlaPnuKacliLz6N
Hg/xZ02W18XveeURDdRqQ/YsBgj8gK4Ovq2+02z1KHvb23jmmj/uuuawLzwd4fsNMvJbbSreOQQu
QwXnoamrhXKTkmbZfxHGjQjRnFtrQ8H8LyrB4q0uVyAq3KEk/WvqEcgeelfJSsY5Q6nDK2Qa+kP
BPiWDxHoMEokH2mJQk6dww7/jRg5JXiXxRh5S5MQttmcR8YvD88ktrcEZdETyp8D7ozlT+p/Ovl
6+tZol7iFopkV0YYKsMg1xl78LPDV5OZRbyQZ6rE5A/KnWwznLmiY5Tn8MNRVGSnps0fP6l0jqkal
nY4AA5Ne/fDTwtL4e0Vprtdt3dEO6n+Adh/n1rS0bwHoGhyLLbWYaZekkp3MKvXOv2sGv2ejq6tcz
hmZR/CoHX86qjQVN80jLnc5lj4+xoRfLu/kbJ6V8zeNv+R11b/ruf5CvpntXzN43/5HXVv+u5/kKWM+
BD4W/wB6l6fqc/X1F4X/AORY03/r3T+Qr5dr6i8L/wDIsab/ANe6fyFZ4PdnfxX/AA6fqzl/i9/yJx/67p/O
vBa96+L3/Inf9t0/nXgtRi/4h2cMf7m/VhRRRXKfRHRpWT/1ur/SL/2avXpv9S/+6a8i+CX+t1j6Rf8As1e
uzf6l/wDdNeth/wCEj80zz/kYT+X5l+S5v9fJ/vGmU+b/AF8n+8aZXIS3P0el/DXoFe//AAnuRP4LiQD
DRSOHgfevAK9G+FPiePS9Tk0q6kCQXRzGzHgP6fjW+Gmo1NTxulcNKvg3ybx1PYPEOm/2voF7
YA4aeJkB9DivmG9s7jT7yW0uozHNE21InfWAO4ZB4NYOueD9F8QHdfWamXGPMXhvzrsr0Paa
o+VybN1gG4zV4s+Za7T4d+FLjXdbhu5YyLC2fe7kfeYdAPxr0u2+E/hq3nWVoppdv8LyEg/hXTytpv
hzSXk2x21pAuSBwAKxp4Vp3merj+l41qfssMnd6Gko2qAB04rzn4yf8ixb/wDXwP5Gu30TUhq+kW
1+qIFnTeF9AelcP8ZP+RYt/wDr4X+RrprO9NngZXFxx9NPe54dXoHwo19dM8QNp8z7Yb0YXPQO
On515/T45HhIWSNirodysDyDXl058krn6NjcLHFUJUn1PrXtXIXxC+HVzqN4+raOivK4/fQZA3H1Hv
8AWtfwJ8QrbW7WKx1CVldRQbfmOBL7j39q70EHkeGvUahWifnEJ4nK8RfZr7mfK0uj6nBI0cun3S
upwR5RrW0PwPruu3CrFZvBDuw80w2gD2HU19ltDExyY1J9wKbJJDbxs7skaKMknAArBYOKd2z
2KnFFecOWEE mzO8OaBaeG9lisbUcKMu+OXbuTWvXn6ePltX8dWGj6Y4e0BczSjo5CnAhtXoHa

uqDi1aPQ+fxVKtCalW3lqfOfxM/5KBqn1j/9FrXJV1nxLOfiBqh94/8A0WtcnXk1v4jP0zK/9zpf4V+QU
UUUVmdwUUUUAAAAFAC0UVPY2c+o3sNnbRmSaZgqqKEr6lmc1CLILZGz4Q8MT+KNZW2TK28
ZDTyf3V9Pqa+j7Gyh0+yitLZAKUSHVUdhWR4R8M2/hjRo7WMBp2G6aTHLNXQV61CI7OOu5+a5
zmUsbW934Vt/mLRRRW544UUUUAAAAFABRRRQAUUUUAAAAFABRRRQAUUUUAAAAFABR
RRQAUUUUAAAAFABRRRQAUUUUAAAAFABRRRQAUUUUAAeUfGv8A48dK/wCur/yrxztXsfxr/w
CPHSv+ur/yrxztXIYr+lz9l4c/3CPz/MKKKK5z3ArZ0DxPqnhu582wnlUn54m5RvvrGopqTi7ozq0YV
oOFRXR7Lp3xms3jxqFhLG47xHcDWl/wuHw7/wA87z/v0P8AGvCKK6Fi6iR4c+GsFJ3V18z227+M
ukpExtbK5lfsHAUfnzXn/iT4haz4iRoDJ9mtGzmGLjcPc965KjtUTxE5Kx04XI8Hh5c0Y3fmFFFFYnrh
Xb+GviXqugW6Wkyi7tUGFVz8yjsAa4iiqhOUHeJz4nCUcTDkrRuj2eH40WLR5m02ZH9FYefnVHU
/jFFdWs9tBpT7ZEKB2l9R6YryaitvrVRo8uPD2BjLm5fxYvX8a0NG1u/0G+W70+cxSDAYdnHoRWd
RWCK07o9idKFSHJJXR7NpXxlsnhVdUs5Y5BjLQ/MD+FbbfFbwyqFvtEpwM7RGcmvn+kroWlmjw
qnDWDnK6uj1nXfjEZYWh0a0ZGYcTTdvwrgtJ8TXum+JY9blc3NwC2/efvZBFYdLWcq05O7Z34fK
cLQpunCO+56j/wALovcf8g2L/vs153rGotq+sXWoOgRp33lQelUaKU6s5q0jTC5bhsLJyoxs2Fekad8
W7vT9Ot7NdPjZYYwgYsecV5vRShUID4S8VgaGKSVZXsdt4p+llx4o0n7BLZpEu8PuDZ6VxNFFK
c3N3kXhsLSw0PZ0IZBRRRUnQdP4P8ZTeEmujDbLN9oC53HGMZ/xrqn+M166lTpsXlx9815dRW
sa04qyZ5tfKcJXqOpUjdsczbnZiOpzTaKXvWR6SVIZCUoJByDg9eKSigGrnofhn4q6hpMKWupRm8
gUACTOHA+veu5tvi14cnTMjTwsOzx14HRXRDE1lqx4eJ4fwdeXPaz8j3Tufi/olujCzinuZMccbV/Ov
LfEj/LVPFE3+kyeXbqfkgQ4UfX1Nc7RUzrznozXB5LhcLLmirvuz0PRvird6PpFrYJYRusEYQMWPO
Kz/FnxAuPFWnR2ktmkISQPuVs1xLSdabXLc1hIOehV9tGPvbiVJFBNOSsMUKhA5CKTio69D+EE
kX/AAk1zBlqsZIMqCPQ/wD16mnDnkom+OxLw2HIWSvY4Yadfg5Fnc8H/nk3+FbFn408TaUpgi1O
4XacbZhul/765r6R+ywY/wBSn/flr55+JGmtp3jS7OwrHPiVPQ5HOPYroqUXRjzRZ4WAZwImdX2N
amtvUsS/FLxPJfSfW6jRv76xjP61gap4l1nWhtv8AUJpkPVM4X8hxWTRXO6s3uz3KWX4Wk7wgl8jt
fhXbtN45tnHSGN2b8sf1r6ErX/4NaM/m3msSKQmPjiPr3b+levscKSewr0cLG1PU+D4hrKpjmo9Ekf
OHxFbd491Qj+8n/oC1y1a3ie7N94o1O53Bg9w+0juoOB+grJrzqjvNs++wEXDDU4vokFFFFQdQU
UUUAAAAFAC+1e3fC/wd/ZlKNXvosXk4/dqw5jT/ABNcb8NvBz67qS6jdp/oFs2cEf61x2+gr3tVCqFA
AAGABXdhA22fGcRZrf/AGWk/X/IUUtFFd58cFFFFABRRRQAUUUUAAAAFABRRRQAUUUUAF
FFFABRRRQAUUUUAAAAFABRRRQAUUUUAAAAFABRRRQAUUUUAAAAFABRRRQB5R8a/w
Djx0r/AK6v/KvHO1ex/Gv/Al8dK/66v/KvHO1eViv4jP0jhz/cl/P8woornPcCiiigAooooAKKKKACiiigA
ooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiig
Be9bfhHWf7B8T2d87YiVtkv+6eD/Q1h0U4txd0ZV6Ua1N05bM+tlJ47iBJYmDo6gqwOQRWF4p8l2
HiqzWK6yk0ZzHMn3l9a8d8JfEbUPDcaWk6fa7AdELYZPof6V6bp3xS8N3qJ5tw9tI38EqHj8RkfrXq
RrU6kbM/O6+VY3A1uemm7bNHEXHwa1VJCIL6CRM8FIIOku6T8GpftCvql8piHjHGCfbNegjxr4cl
z/a9rg/9NBUNz4+8M2sZd9Vhb2Q7j+ISqNFO5vLNM2nHks/uNywsbfTbKK0tYljhiXaqqK5vx94pi8P
aFKqSD7ZOpSFQeRnv+Fc1rxisoo3i0i1knlxxLJ8qD+teT6rq99rd815fzGWVunoo9AKmtilxVomuW
ZFXrVVVxKtHfXdlJJJJ5J5pKKK84+9StogooooAKKKKACtrwx4dufEusR2UAljzmWTHCL/jWVb28
t1cx28CF5ZGCl07k19F+B/CcXhfR1Q4a7mG6aT39PoK3w9H2krvY8XOsZWD02j8b2/wAzc0vTbfS
dPhsrSMJFEu0AD9au0UV6qSSsfm0pOT5pbsWiiimlKKKKACiiigAooooAKKKKACiiigAooooAKKK
KACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigDyj41/8eOlf9dX/IXjnavY/jX/AM
eOlf8AXV/5V452rysV/EZ+kCOf7hH5/mFFFFc57gUUUUAAAAFABRRRQAUUUUAAAAFABRRRQ
AUUUUAAAAFABRRRQAUUUUAAAAFABRRRQAUUUUAAAAFABRRRQAUUUUAAAAFABRRR
QAUUUUAAAAFABRRRQAUUUUCsgooooGFFFFABRRRQAAtJRXafDzwe3iPVRc3KH+z7dgXyPv
t/dqoQc5WRz4vFQw1J1Z7l674V+DfJhXXr+liVx/oyMPur/AHvqf5V6uKbHGsaKiKAqjAA7U6vYpwU

I2R+XY3FzxdZ1Zi0UUVZyBRRRQAUUUUAAAAFABRRRQAUUUUAAAAFABRRRQAUUUUAAAAF
FABRRRQAUUUUAAAAFABRRRQAUUUUAAAAFABRRRQAUUUUAAAAFAHmXxf0691Cx00WV
pPcFJGLCKMtjj2ryf/hGdd/6A1/8A+A7f4V9SYHpSYHoPyrmqYaM5czZ72Bz6rg6KoximkfLn/CM67/
0Br/8A8B2/wo/4RnXf+gNf/wDgO3+FfUWB/dH5UYH90flUfU4dzt/1rr/yl+Xf+EZ13/oDX/8A4Dt/hR/w
jOu/9Aa//wDAdv8ACvqLA/uj8qMD+6Pyo+pw7h/rXX/kR8u/8lZrv/QGv/8AwHb/AAo/4RnXf+gNf/8Ag
O3+FfUWB/dH5UYH90flR9Th3H/rXX/kR8u/8lZrv/QGv/8AwHb/AAo/4RnXf+gNf/8AgO3+FfUWB/d
H5UYH90flR9Th3D/Wuv8Ayl+Xf+EZ13/oDX//AIDt/hR/wjOu/wDQGv8A/wAB2/wr6jwPQUuB6Cj6n
DuH+tVf+RfifLf/AAjOu/8AQGv/APwHb/Cj/hGdd/6A1/8A+A7f4V9R4HoPyowPQfIR9Th3D/WrEfyI+
XP+EZ13/oDX/wD4Dt/hR/wjOu/9Aa//APAdv8K+osD+6PyowP7o/Kj6nDuL/Wuv/lj5d/4RnXf+gNf/A
PgO3+FH/CM67/0Br/8A8B2/wr6iwP7o/KjA/uj8qPqcO4/9a6/8iPl3/hGdd/6A1/8A+A7f4Uf8lZrv/QGv/
wDwHb/CvqLA/uj8qMD+6Pyo+pw7h/rXX/kR8u/8lZrv/QGv/wDwHb/Cj/hGdd/6A1//AOA7f4V9RYH9
0flRgf3R+VH1OHcP9a6/8iPl3/hGdd/6A1//AOA7f4Uf8lZrv/QGv/8AwHb/AAr6iwP7o/KjA/uj8qPqcO4
f611/5EfLv/CM67/0Br//AMB2/wAKP+EZ13/oDX//AIDt/hX1Fgeg/KjA9B+VH1OHcX+tdf8AkR8u/wD
CM67/ANAa/wD/AAHb/Cj/AIRnXf8AoDX/AP4Dt/hX1Fgeg/KjA/uj8qPqcO4f611/5EfLv/CM67/0Br//A
MB2/wAKP+EZ13/oDX//AIDt/hX1Fgf3R+VGB/dH5UfU4dx/611/5EfLv/CM67/0Br//AMB2/wAKP+EZ
13/oDX//AIDt/hX1Fgf3R+VGB/dH5UfU4dw/1rr/AMiPl3/hGdd/6A1//wCA7f4Uf8lZrv8A0Br/AP8AAAdv
8K+o8D0FGB6Cl9Th3D/WrEfyI+XP+EZ13/oDX/wD4Dt/hR/wjOu/9Aa//APAdv8K+osD+6PyowP7o/
Kn9Th3F/rXX/kR8u/8ACM67/wBAa/8A/Adv8KP+EZ13/oDX/wD4Dt/hX1Fgf3R+VGB/dH5UfU4dx/6
11/5EfLv/AAjOu/8AQGv/APwHb/Cj/hGdd/6A1/8A+A7f4V9RYH90flRgf3R+VH1OHcP9a6/8iPl3/hG
dd/6A1/8A+A7f4Uf8lZrv/QGv/wDwHb/CvqPA9BRgego+pw7h/rVX/kR8uf8ACM67/wBAa/8A/Adv8K
P+EZ13/oDX/wD4Dt/hX1Fgf3R+VGB/dH5UfU4dx/611/5EfLv/AAjOu/8AQGv/APwHb/Cj/hGdd/6A1/
8A+A7f4V9RYH90flRgf3R+VH1OHcf+tdf+RHy7/wAlZrv/AEBr/wD8B2/wo/4RrXf+gNff+A7f4V9R4H
oPyowPQfIR9Th3F/rXX/kR8uf8lZrv/QGv/wDwHb/Cj/hGdd/6A1//AOA7f4V9R4HoPyowPQUfU49w/
wBa6/8Alj5c/wCEZ13/AKA1/wD+A7f4Uf8ACM67/wBAa/8A/Adv8K+o8D0H5UYHoPyfVldw/1rr/yl
+XP+EZ13/oDX/wD4Dt/hR/wjOu/9Aa//APAdv8K+osD0H5UYH90flT+pw7h/rXX/AJEfNGm+DNd1D
UYLU6ZdQrl4DSSQsqQO5JNfQ+haNbaFpUNhaqAka8nux7k1o4Gegpe1bUqEaex5eZZtWx1IPRL
ohaKKK2PKCiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigA
ooooAKKKKACiiigAooooAKKKKACiiigD//2Q0KZW5kc3RyZWFTdDQplbmRvYmoNCjlxIDAgb2JqDQ
o8PC9UeXBIL09ialN0bS9OIDE4Ny9GaXJzdCAxNTkzL0ZpbHRlci9GbGF0ZURlY29kZS9MZW5n
dGggMjc0Mz4+DQpzdHJlYW0NCnclcnVrbjhs3En0PkH/gH4jFO4EgwCJOsAsjhjEzwD4EeZBtxR5k
PDImMpD8/Z5qFqVui+zm6mGGLamreOrw8FYkZaUVJeVJGaNIJ0VRkc/KeHxhlXHKRP5T1uKV
oGwyyiTILF6xymWt8L0PpKxWgQxeUyFkZY2KBr85FWNUJQsElxtVyvgqtqezxW0B9OioHADqScqSI
4BOeiRI+W0XGeeUcyoz3gM96fPaKHMENI2wHzN5I5QA8aICBv+CnkCiSBkaUwSsPfwlIPPwl+Pf
wlwECLo3WeC+gxJcwNYRomAQYKB+ZiYjoEDH8BxBg4SxosAPCAuhx0ShUYTz4CmDNxwwaw
BeC8FxmR8CKiRF28J9AL6CblJgn8MNEwV9OWYFuq+E/elCr8R4oQWAqgnzD/tEWBh9QhbUIJjH
NGp9BsANluLQe/hPsPMhHqDagPRL8hRxUgp8l0hLqSdqdBH8JTvGqzeCVmzRnbio0DcjLBk0D/
xnNTSAng3lQhDZECSO4dBa8ZocSTrmpHPuHSrzmXgW3Hugl5LkAZqEBFwCHULuL3E4atgn/S
KOyFNglXrNhicBdRqxMhdfc6JrbFr4yl9yqxI0JBgmt7Q3ilcDzIAMPDg+ZFYWGdGhqIm5x1jqE7n3
kb+A4WFYfHlBMQKKGSCZeB52RZiKwLMEeGhcFS5JbQeJFAddCBv0Hbsg8yLBaOD5oPxvl3AQ9
5EjL0EbgupDigl7RMcOCQgCCwjgmRhMA9EcGGwC9zp4rcMVmGMbEVPCfL38BzQtMQug1aF8
yB4KjZM1BGzZ4hwkjcLdDF0CvhGX0oGo4dHEV/uPElvi5apQ8tFx30EUKKfXoY2PNTMLmJgNhw/
8Mss0whhE/pNTCCAQHfKvZNnhXlfh6+YOXZoJ2kOBz0rMevkWZ3QN8s7GR4IPOuXX0YvQnw

[illegible]

A3NzggMCAyNzggMCA3MjlgNjExIDAgNzlyIDc3OCA2NjcgNzc4IDAgNjY3IDYxMSAwIDA gMCAwIDA gMCAwIDA gMCA1NTYgMCA1NTYgMCA1NTYgMCA2MTEgNjExIDI3OCAwIDU1N iAwIDg4OSA2MTEgNjExIDYxMSAwIDM4OSA1NTYgMzMzIDYxMSAwIDA gMCA1NTZdIA0KZW5 kb2JqDQoyMDIgMCBvYmoNClsGmjc4XSANCmVuZG9iaG0KMjAzIDA g2JqDQo8PC9UeXBIL1hS ZWYvU2I6ZSAyMDMvV1sgMSA0IDJdIC9Sb290IDEgMCBSL0luZm8gMjAwIDA gUi9JRFs8RDUz Qzk0RkE0RjAzN0I0MTk4Nzc1RTc1RDkzQ T I C O D E + P E Q 1 M 0 M 5 N E Z B N E Y w M z d C N D E 5 O D c 3 N U U 3 N U Q 5 M 0 E 5 Q j g x P I 0 g L 0 Z p b H R I c i 9 G b G F 0 Z U R I Y 2 9 k Z S 9 M Z W 5 n d G g g M z k 3 P j 4 N C n N 0 c m V h b Q 0 K e J w 1 1 E V T I m E U B u D n w 0 7 4 U O w A G 7 t b U e x u s Q O w u x O 7 u 7 s D u 0 X U G V e u n e F f u H H h P 8 D X 9 9 K z O N e c x X 2 W d w j R I J c n o p 0 M 4 S 9 F K I I J / I x J z U Z Z T F o p f s U k s / A 9 J r 0 4 h J Q o n h F K 8 R I f 8 A k f U A E J / M t 9 j b 7 k l f 2 / U I A N I V A R V V A Z V Z G K G q i O W q i J 2 m i E d K Q h i Q z U Q V 0 0 Q D 3 U R 0 O 0 Q h M 0 R n M 0 R T O 0 Q C a y 0 B J d 0 B a t 0 Q b t 0 Q 7 Z 6 I Q O 6 I j O 6 I d u 6 I q e 6 I 4 e 6 I N e 6 I 2 + G I 6 B 6 I 8 B y M E g D E Y u h m A o h m E i R m I E x m A U R m M 8 x m l c J m A 2 p m A S J m M 6 p m l a 8 j A D M z E L B Z i L O V i A e Z i P x V i I R c j H W i x F I Z Z g B Z Z h O V Z j J V Z h D b Z j P d Z h E z Z g I 7 Z i M 7 Z g G w 5 i F 3 Z g J / Z g N 4 q w H 3 u x D w d w G o d x C M d w B E d x E s d x A q d w D e d w B m d x E e d x A V d w C Z d x F Q 9 w A 9 d x G z d x C / d w B 3 d x H 0 / x C A 9 R j M d 4 g l d 4 j m d 4 g Z d 4 j z d 4 j b d 4 h 4 8 o i U o u / 3 f c m A X f Y g p / h P A H H m x i V w 0 K Z W 5 k c 3 R y Z W F t D Q p l b m R v Y m o N C n h y Z W Y N C j A g M j A 0 D Q o w M D A w M D A w M D E y I D Y 1 N T M 1 I G Y N C j A w M D A w M D A w M T c g M D A w M D A g b g 0 K M D A w M D A w M D E y N S A w M D A w M C B u D Q o w M D A w M D A w M T g x I D A w M D A w I G 4 N C j A w M D A w M D A 0 O D c g M D A w M D A g b g 0 K M D A w M D A w M z g 4 M S A w M D A w M C B u D Q o w M D A w M D A 0 M D Q 4 I D A w M D A w I G 4 N C j A w M D A w M D Q y N z k g M D A w M D A g b g 0 K M D A w M D A w N D M z M i A w M D A w M C B u D Q o w M D A w M D A 0 M z g 1 I D A w M D A w I G 4 N C j A w M D A w M D Q 1 N D c g M D A w M D A g b g 0 K M D A w M D A w N D c 3 N C A w M D A w M C B u D Q o w M D A w M D A w M D E z I D Y 1 N T M 1 I G Y N C j A w M D A w M D A w M T Q g N j U 1 M z U g Z g 0 K M D A w M D A w M D A x N S A 2 N T U z N S B m D Q o w M D A w M D A w M D E 2 I D Y 1 N T M 1 I G Y N C j A w M D A w M D A w M T c g N j U 1 M z U g Z g 0 K M D A w M D A w M D A x O C A 2 N T U z N S B m D Q o w M D A w M D A w M D E 5 I D Y 1 N T M 1 I G Y N C j A w M D A w M D A w M j A g N j U 1 M z U g Z g 0 K M D A w M D A w M D A y M S A 2 N T U z N S B m D Q o w M D A w M D A w M D I y I D Y 1 N T M 1 I G Y N C j A w M D A w M D A w M j M g N j U 1 M z U g Z g 0 K M D A w M D A w M D A y N C A 2 N T U z N S B m D Q o w M D A w M D A w M D I 1 I D Y 1 N T M 1 I G Y N C j A w M D A w M D A w M j Y g N j U 1 M z U g Z g 0 K M D A w M D A w M D A y N y A 2 N T U z N S B m D Q o w M D A w M D A w M D I 4 I D Y 1 N T M 1 I G Y N C j A w M D A w M D A w M j k g N j U 1 M z U g Z g 0 K M D A w M D A w M D A z M C A 2 N T U z N S B m D Q o w M D A w M D A w M D M x I D Y 1 N T M 1 I G Y N C j A w M D A w M D A w M z I g N j U 1 M z U g Z g 0 K M D A w M D A w M D A z M y A 2 N T U z N S B m D Q o w M D A w M D A w M D M 0 I D Y 1 N T M 1 I G Y N C j A w M D A w M D A w M z U g N j U 1 M z U g Z g 0 K M D A w M D A w M D A z N i A 2 N T U z N S B m D Q o w M D A w M D A w M D M 3 I D Y 1 N T M 1 I G Y N C j A w M D A w M D A w M z g g N j U 1 M z U g Z g 0 K M D A w M D A w M D A z O S A 2 N T U z N S B m D Q o w M D A w M D A w M D Q w I D Y 1 N T M 1 I G Y N C j A w M D A w M D A w N D E g N j U 1 M z U g Z g 0 K M D A w M D A w M D A 0 M i A 2 N T U z N S B m D Q o w M D A w M D A w M D Q z I D Y 1 N T M 1 I G Y N C j A w M D A w M D A w N D Q g N j U 1 M z U g Z g 0 K M D A w M D A w M D A 0 N S A 2 N T U z N S B m D Q o w M D A w M D A w M D Q 2 I D Y 1 N T M 1 I G Y N C j A w M D A w M D A w N D c g N j U 1 M z U g Z g 0 K M D A w M D A w M D A 0 O C A 2 N T U z N S B m D Q o w M D A w M D A w M D Q 5 I D Y 1 N T M 1 I G Y N C j A w M D A w M D A w N T A g N j U 1 M z U g Z g 0 K M D A w M D A w M D A 1 M S A 2 N T U z N S B m D Q o w M D A w M D A w M D U y I D Y 1 N T M 1 I G Y N C j A w M D A w M D A w N T M g N j U 1 M z U g Z g 0 K M D A w M D A w M D A 1 N C A 2 N T U z N S B m D Q o w M D A w M D A w M D U 1 I D Y 1 N T M 1 I G Y N C j A w M D A w M D A w N T Y g N j U 1 M z U g Z g 0 K M D A w M D A w M D A 1 N y A 2 N T U z N S B m D Q o w M D A w M D A w M D U 4 I D Y 1 N T M 1 I G Y N C j A w M D A w M D A w N T k g N j U 1 M z U g Z g 0 K M D A w M D A w M D A 2 M C A 2 N T U z N S B m D Q o w M D A w M D A w M D Y x I D Y 1 N T M 1 I G Y N C j A w M D A w M D A w N j I g N j U 1 M z U g Z g 0 K M D A w M D A w M D A 2 M y A 2 N T U z N S B m D Q o w M D A w M D A w M D Y 0 I D Y 1 N T M 1 I G Y N C j A w M D A w M D A w N j U g N j U 1 M z U g Z g 0 K M D A w M D A w M D A 2 N i A 2 N T U z N S B m D Q o

wMDAwMDAwMDY3IDY1NTM1IGYNCjAwMDAwMDAwNjggNjU1MzUgZg0KMDAwMDAwMDA2
OSA2NTUzNSBmDQowMDAwMDAwMDcwIDY1NTM1IGYNCjAwMDAwMDAwNzEgNjU1MzUgZ
g0KMDAwMDAwMDA3MiA2NTUzNSBmDQowMDAwMDAwMDczIDY1NTM1IGYNCjAwMDAwM
DAwNzQgNjU1MzUgZg0KMDAwMDAwMDA3NSA2NTUzNSBmDQowMDAwMDAwMDc2IDY1NT
M1IGYNCjAwMDAwMDAwNzcgNjU1MzUgZg0KMDAwMDAwMDA3OCA2NTUzNSBmDQowMDA
wMDAwMDc5IDY1NTM1IGYNCjAwMDAwMDAwODAgNjU1MzUgZg0KMDAwMDAwMDA4MSA2
NTUzNSBmDQowMDAwMDAwMDgyIDY1NTM1IGYNCjAwMDAwMDAwODMgNjU1MzUgZg0KM
DAwMDAwMDA4NCA2NTUzNSBmDQowMDAwMDAwMDg1IDY1NTM1IGYNCjAwMDAwMDAw
ODYgNjU1MzUgZg0KMDAwMDAwMDA4NyA2NTUzNSBmDQowMDAwMDAwMDg4IDY1NTM1I
GYNCjAwMDAwMDAwODkgNjU1MzUgZg0KMDAwMDAwMDA5MCA2NTUzNSBmDQowMDAw
MDAwMDkxIDY1NTM1IGYNCjAwMDAwMDAwOTIgNjU1MzUgZg0KMDAwMDAwMDA5MyA2NT
UzNSBmDQowMDAwMDAwMDk0IDY1NTM1IGYNCjAwMDAwMDAwOTUgNjU1MzUgZg0KMDA
wMDAwMDA5NiA2NTUzNSBmDQowMDAwMDAwMDk3IDY1NTM1IGYNCjAwMDAwMDAwOTgg
NjU1MzUgZg0KMDAwMDAwMDA5OSA2NTUzNSBmDQowMDAwMDAwMTAwIDY1NTM1IGYN
CjAwMDAwMDAxMDEgNjU1MzUgZg0KMDAwMDAwMDEwMiA2NTUzNSBmDQowMDAwMDAw
MTAzIDY1NTM1IGYNCjAwMDAwMDAxMDQgNjU1MzUgZg0KMDAwMDAwMDEwNSA2NTUzNS
BmDQowMDAwMDAwMTA2IDY1NTM1IGYNCjAwMDAwMDAxMDcgNjU1MzUgZg0KMDAwMDA
wMDEwOCA2NTUzNSBmDQowMDAwMDAwMTA5IDY1NTM1IGYNCjAwMDAwMDAxMTAgNjU1
MzUgZg0KMDAwMDAwMDExMSA2NTUzNSBmDQowMDAwMDAwMTEyIDY1NTM1IGYNCjAw
MDAwMDAxMTMgNjU1MzUgZg0KMDAwMDAwMDExNCA2NTUzNSBmDQowMDAwMDAwMTE
1IDY1NTM1IGYNCjAwMDAwMDAxMTYgNjU1MzUgZg0KMDAwMDAwMDExNyA2NTUzNSBmD
QowMDAwMDAwMTE4IDY1NTM1IGYNCjAwMDAwMDAxMTkgNjU1MzUgZg0KMDAwMDAwMD
EyMCA2NTUzNSBmDQowMDAwMDAwMTIxIDY1NTM1IGYNCjAwMDAwMDAxMjIgNjU1MzUgZg
0KMDAwMDAwMDEyMyA2NTUzNSBmDQowMDAwMDAwMTI0IDY1NTM1IGYNCjAwMDAwMD
AxMjUgNjU1MzUgZg0KMDAwMDAwMDEyNiA2NTUzNSBmDQowMDAwMDAwMTI3IDY1NTM1I
GYNCjAwMDAwMDAxMjggNjU1MzUgZg0KMDAwMDAwMDEyOSA2NTUzNSBmDQowMDAwMD
AwMTMwIDY1NTM1IGYNCjAwMDAwMDAxMzEgNjU1MzUgZg0KMDAwMDAwMDEzMjA2NTUz
NSBmDQowMDAwMDAwMTMzIDY1NTM1IGYNCjAwMDAwMDAxMzQgNjU1MzUgZg0KMDAwM
DAwMDEzNSA2NTUzNSBmDQowMDAwMDAwMTM2IDY1NTM1IGYNCjAwMDAwMDAxMzcgNj
U1MzUgZg0KMDAwMDAwMDEzOCA2NTUzNSBmDQowMDAwMDAwMTM5IDY1NTM1IGYNCjA
wMDAwMDAxNDAgNjU1MzUgZg0KMDAwMDAwMDE0MSA2NTUzNSBmDQowMDAwMDAwMT
QyIDY1NTM1IGYNCjAwMDAwMDAxNDMgNjU1MzUgZg0KMDAwMDAwMDE0NCA2NTUzNSBm
DQowMDAwMDAwMTQ1IDY1NTM1IGYNCjAwMDAwMDAxNDYgNjU1MzUgZg0KMDAwMDAwM
DE0NyA2NTUzNSBmDQowMDAwMDAwMTQ4IDY1NTM1IGYNCjAwMDAwMDAxNDkgNjU1MzU
gZg0KMDAwMDAwMDE1MCA2NTUzNSBmDQowMDAwMDAwMTUxIDY1NTM1IGYNCjAwMDA
wMDAxNTIgNjU1MzUgZg0KMDAwMDAwMDE1MyA2NTUzNSBmDQowMDAwMDAwMTU0IDY1
NTM1IGYNCjAwMDAwMDAxNTUgNjU1MzUgZg0KMDAwMDAwMDE1NiA2NTUzNSBmDQowM
DAwMDAwMTU3IDY1NTM1IGYNCjAwMDAwMDAxNTggNjU1MzUgZg0KMDAwMDAwMDE1OS
A2NTUzNSBmDQowMDAwMDAwMTYwIDY1NTM1IGYNCjAwMDAwMDAxNjEgNjU1MzUgZg0K
MDAwMDAwMDE2MiA2NTUzNSBmDQowMDAwMDAwMTYzIDY1NTM1IGYNCjAwMDAwMDAx
NjQgNjU1MzUgZg0KMDAwMDAwMDE2NSA2NTUzNSBmDQowMDAwMDAwMTY2IDY1NTM1IG
YNCjAwMDAwMDAxNjcgNjU1MzUgZg0KMDAwMDAwMDE2OCA2NTUzNSBmDQowMDAwMDA

wMTY5IDY1NTM1IGYNCjAwMDAwMDAxNzAgNjU1MzUgZg0KMDAwMDAwMDE3MSA2NTUzN
SBmDQowMDAwMDAwMTcyIDY1NTM1IGYNCjAwMDAwMDAxNzMgNjU1MzUgZg0KMDAwMD
AwMDE3NCA2NTUzNSBmDQowMDAwMDAwMTc1IDY1NTM1IGYNCjAwMDAwMDAxNzYgNjU1
MzUgZg0KMDAwMDAwMDE3NyA2NTUzNSBmDQowMDAwMDAwMTc4IDY1NTM1IGYNCjAwM
DAwMDAxNzkgNjU1MzUgZg0KMDAwMDAwMDE4MCA2NTUzNSBmDQowMDAwMDAwMTgxID
Y1NTM1IGYNCjAwMDAwMDAxODIgNjU1MzUgZg0KMDAwMDAwMDE4MyA2NTUzNSBmDQow
MDAwMDAwMTg0IDY1NTM1IGYNCjAwMDAwMDAxODUgNjU1MzUgZg0KMDAwMDAwMDE4Ni
A2NTUzNSBmDQowMDAwMDAwMTg3IDY1NTM1IGYNCjAwMDAwMDAxODggNjU1MzUgZg0K
MDAwMDAwMDE4OSA2NTUzNSBmDQowMDAwMDAwMTkwIDY1NTM1IGYNCjAwMDAwMDAx
OTEgNjU1MzUgZg0KMDAwMDAwMDE5MiA2NTUzNSBmDQowMDAwMDAwMTkzIDY1NTM1IG
YNCjAwMDAwMDAxOTQgNjU1MzUgZg0KMDAwMDAwMDE5NSA2NTUzNSBmDQowMDAwMD
AwMTk2IDY1NTM1IGYNCjAwMDAwMDAxOTcgNjU1MzUgZg0KMDAwMDAwMDE5OCA2NTUz
NSBmDQowMDAwMDAwMTk5IDY1NTM1IGYNCjAwMDAwMDAwMDAgNjU1MzUgZg0KMDAwM
DAyNTMyOCAwMDAwMCBuDQowMDAwMDI1NTMzIDAwMDAwIG4NCjAwMDAwMjU4MDUgM
DAwMDAgbg0KMDAwMDAyNTgzMyAwMDAwMCBuDQp0cmFpbGVyDQo8PC9TaXplIDIwNC9S
b290IDEgMCBSL0luZm8gMjAwIDAguIDAgUi9JRFs8RDUzQzk0RkE0RjAzN0I0MTk4Nzc1RTc1RDkzQT
ICODE+PEQ1M0M5NEZBNEYwMzdCNDE5ODc3NUU3NUQ5M0E5QjgxPI0gPj4NCnN0YXJ0eHJ
IZg0KMjY0MzQNCiUIRU9GDQp4cmVmDQowIDANCnRyYWlsZXINCjw8L1NpemUgMjA0L1Jvb3
QgMSAwIFlvSW5mbyAyMDAgMCBSL0IEWzxENTNDOTRGQTRGM3QjQxOTg3NzVFNzVEO
TNBOUI4MT48RDUzQzk0RkE0RjAzN0I0MTk4Nzc1RTc1RDkzQTICODE+XSAvUHJldiAyNjQzNC
9YUmVmU3RtIDI1ODMzPj4NCnN0YXJ0eHJIZg0KMzA2NzQNCiUIRU9G";

}

}

}

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
```

```
namespace Web_App_Master.Account.Templates
```

```
{
    public class StarragReceivingReportTemplate : Pdf.PdfTemplate
    {
        public StarragReceivingReportTemplate()
```

```
        {
            this.TemplateBase64 =
"JVBERi0xLjUNCiW1tbW1DQoxIDAgb2JqDQo8PC9UeXBIL0NhdGFsb2cvUGFnZXMGMiAwIFlvT
GFuZyhlbi1VUykgL1N0cnVjdFRyZWVSb290IDE3IDAgUi9NYXJrSW5mbzw8L01hcmtlZCB0cnVIPj
4+Pg0KZW5kb2JqDQoyIDAgb2JqDQo8PC9UeXBIL1BhZ2VzL0NvdW50IDEvS2lkc1sgMyAwIFJdl
D4+DQplbmRvYmoNCjMgMCBvYmoNCjw8L1R5cGUvUGFnZS9QYXJlbnQgMiAwIFlvUmVzb3Vy
Y2VzPDwvRm9udDw8L0YxIDUgMCBSL0YyIDkgMCBSL0YzIDE0IDAgUj4+L0V4dEdTdGF0ZTw8
L0dTNyA3IDAgUi9HUzggOCAwIFI+Pi9YT2JqZWN0PDwvSW1hZ2UxNiAxNiAwIFI+Pi9Qcm9jU2V
0Wy9QREYvVGV4dC9JbWFnZUIvSW1hZ2VdL0ltYWdlSV0gPj4vTWVkaWFCb3hbIDAgMCA1OT
UuMzlgODQxLjkyXSAvQ29udGVudHMgNCAwIFlvR3JvdXA8PC9UeXBIL0dyb3VwL1MvVHJhbnN
wYXJlbnN5L0NTL0RldmliZVJHqj4+L1RhYnMvUy9TdHJ1Y3RQYXJlbnRzIDA+Pg0KZW5kb2JqD
Qo0IDAgb2JqDQo8PC9GaWx0ZXIvRmxhdGVEZWNVZGUvTGvUz3RoIDM4MjE+Pg0Kc3RyZWFi
tDQp4nK1dTdY/dthXdDzD/QUA3bwKMzG+RRZCF7SRN0QBxYqCLoAvDnrhGG9txnKL99yUIUr8
uuRwBgkm8bzzLq8OLw8PKUr+7fqKq1XoZVny5WqRTKxaL4KsRC+f7q6v/v7F8v76annyw/LlI0++f
/bd84U8+dur92+Xy5u722d/ufnqq+Xp82fLb2NxnR68vnryDV3MatTy8pfrK7oQ+w9dBF2VWDajVs6
WI79a1Lc/bcvb36+vyPJ2/5P2f/r2+urny493r+9uxOXdzT+WI3+9vvrahn1xffXQJJReqU6y2Bv3bf7nc
Rvb+KpZs7F379/e3MrL8uPdxw83t1RdPt3wy+fHzYAym4JqZbCkjd/I6+/f7bASqC+88m6CeO+apP
QRK5C2F/ZvPYfLpFfvuh95v7kP/aXkQCkomsagBBxb4j92FO1/zoSBb4PAO77KWZn4KRdSr6KhS
m2UrUwSldLQn3IMHTIuDjcHIFkGag3ZqRcN1V03zc3t2wvmg/uf379c6N0ZttW9je62fhA7fCzdrSk7
qvMmJXxZSNsIXqh3H7AFpsWt72Q879jxYHdIJY2OkD41mywIw3LxvRKmB0AZFX2P3SVLGDflf
G/url2CGN22a1bd1+T5QD7tubW3H01v7jTcphkZaaS4tV0rKdSHgjrS8Jeaq+wIPZHo8hua2bqaey3
NqxSDe9vHz98+X3pUOPfryclFpJs9deuF77w3XYq3+/s1PSZ6vW8vK/Tnbm0bITxLKiwtmNiDjxuQ
gbQoqxAbhjeTIAOTaAKT1bsYoBB6DsJt/Kllm9HHtk2a60LI0OQPvDZrLYXI2evuqNR8ofLU3J9jno
AX0qKjPzpuK8Wpl9/cd+JkgAzOhVcRiAEJpDzEppgLgrNtnk6zghCg0SaltBLG21ST4JUpnkD973X
2cFHC4llwWGSikioJCKhAkOhOW4mbVFBQEa5lsKhuWwcvFHom1lrW1zPa12haltIWV0/ZTW/f8
8t/lxR/OeyJ69vA8jEWqVh6NCo+6A3nkOI+qwaOvtcn8vZA08r+XkDw4F68W9+My4SjM376cadQI
V/eVoUOT8ZuCDh3YQUAGIAYMr3YgIKEUSgHEgCFIEzGAIEfNQWzQA1YoGwD4kc4K4UIxPpM
ElzJMybtueGBvyv1Va2IJBepKieWyNRZM6oKIOezvkatmpNCgZppdG6xWI23z3HmAvGB/sKPk1e
ub2+3yr30RzC/LC/fz3h6L0UfLWBBqlzutjAeGGGOpyRrqYS92EMvxHmY8cVmagg7sui4ICG6fZO
SxwdbDXBaTj5am180HdKqquCwnAby2xwGkhgvURHkiYqKwIGAY8UEaJioJUPOOT+v+66xA/aUA
hMsiA/IUCZBPJQGGJ1njwOxaUu2WG33ax16Bag2HDd11AHJnH6ey5SMHdMnXctwSxBm1yNHeN
2z80Uft4ve7fK/skLROdgpRsNdEqbMbKkwqPiCuKBHmw0RnGI3KG2oeER8UhkYDOUkNDnXJx
```

hjoGRgl6CxKEKht0iKMH9ss/+YfXFT93ySSQ0AiEiBRS7peRaBao25/URPufn94GxoWxgWXczYN
DOh1FEKkAsixFfq7cj9K3U6HgVoepm43cfiGSM8sHXSvB7k61Xo1t2tYtrRuGTzfS2QJrFp5bc0Avd
787B/Ha/frdR/uBurz78L4zt3D+SOKyolZGW/mObl6GydgPd5+Hr9l6v3qJaUErHSszO3j0bLSDWM+
Wu494w8iOnx2Vo31ru5Zc/tTrxnI7cja3Ta9CPaQfdWr7RvrRuz4l5Xg/mtT0xW6UnV4UpDBTk0wJv
a2ywZRVKSKthrtt2xefe65d0MfKSbqyFg/oPcEq/k5S5jqkuklOxmJHgl20yioahEGW0R6Fr6M7ocK0
F0O1V9JJqOpS2vfFfvlp/foLAwD31RTjM0YxPpUEI/C5TBmnX3Eua7kqwVFHCOPiPM5RZrZlCIW
m7upnqixoPdwb8KnEkldNva8jKxefyb11NPuoGlr9i4reYw032QqNz9O1ZZLaZyooKkBylCmdCCog
qruKFqZUrymefNPTNEIS0S5hiLv/iGiXR+DaBclg2uVRuHZ1Qp0TwBmqrV1JqLp2HT0RtKuw7OB
zL0uqyBeBhDwgRFQqMukmOuLqxbZSU+hSRQgBEBVCyRJPL+S2C9/p6dHb4aVLruc3ppvSrFJ
OVXjukY8sRiSzjqzQIEvmwdPgS YGKZGINdyVznqlcMvtMBckESFQyZSaZkaiuZMqKZE7xFJqepUI
VJVNpt1eC7HJ4BL7LAclguxwehe9ydEKds84Zqr3LkYSq7nL4nmhLZvy8KZkIJOQBITXJjFLD+Em
2VTi2GtaoSEVxrcwCySJQT7T4ZqmaqjFWM3tHLqHcz1xq0tWAVq6fZ9pIR4cbU0PapUSpXWjLXf
GaZ0xm4jVAVIAvAOU4WSqVr8hVv77UVsrXHFWh7WmmdE2/hHKrX8zyHYiO5YthMMt3oDqWD
w91TgFnKMTywVB1y3d0xaFfSa2EC4uAXZ0STMgYw4RUIEbU6jLpKjNi+wgFHYOJlgCitm8jqe2z
pe00cMj2bbSQ0Hp+YwpKhTtTN1HIG8ts35HFiO2rlys05dJ58DQmnVtFOrGGu8o5z1SunH2mgnA
CJGr7tkw3l1Fd3dwqujnFU2h6mqaqbLJ4d6chmwzePmzJJktuArVk80B1ZBMPBaYe3pVNltx3qsn
m0RNt2xc/b9o+BBLygBDU9vFtH8FR4UzrpP1mUNu3B4l3dNwh9mHV4sZ+YarlNKn4Pp9MrPeQ
TEW8PAMA22JA01S9uD7OysXbOgy5JaBZIV/NNMcE7AGk8VTAhvjjEgb54jhfltEwQJfss1WeRpl
y7c+T1btMlobYQy9WeERHRmLYTAZO1AdGcNDgZmA9W5WJKGqMub7lsgYLBd/ZQDhVQqCf
M44yGeTgES1OpPu2gYclJddqVXCR2nwUaUdqsC5uPr2ndelBudp2QRzygNoUetrIcExOIXFDdaLY
DUINoE8jrrgqRAUhPZgqoCVTJhfSg6oxG2gqOtplclBG58nKZXSArKCigCyOk5WpaOSq6wRNRU
TnqAptTzNV01B3CGBDb5p4BK6hlAyioR6Fa2gn1DkRnaHaGppEqmvo0RVhBU2KrU0A8KtjUuxZ
opiQCsQIXO24yOxga4Fi8DOAe6DEDt5jDcuFmpyrja65wSMXYBYaK1l//QW0cv3ZkW8uRGYg2+
OSkvLldyvJQQ2bZowSmonYAFtBxABbHGWLEpaqWGSrd5SGkvLxr0myQtvzXNWe/nLDi6ErWo/
oyFgMg8nYgerlGB4KTAWst6JNQTVI7OiL0woWS1oACcavWNOimJAKxIhaZaZ9JUd8oK29QR/ok
EM+kBKVGkELHjaCIgyFIDZyHJNSJianbEp0ZgWPPIas4EHwgBWkJNfRg60xL0hpRUgBaQ4K6T
xfNBfSAb6CkAK+cDdlaSakka7+476VB2kn2QqNz5NVVVLOe4bwQHSUNibBIPRAAdZQUd3VOR3
zAEMJQdSU9+glxhBHQNoQYJqQCMbghZNqsGxs5zU6bD9aCSBw/8d2WMTK7e0OpqlhCn8zls
W/PwMCxbxof9zzExEUy9oS0fK9CK8tBKXsAZyaVshG6vJRBuvDT1ZSRRMoAW11PyMqzzJNke
SWb54rVzjK7EQY8YeVwi0fgh1tAGORwi0fhh1s6ocB0wHqP8CSHqodbfF8ET1ieZQaA5llmFONTS
TDVs8xpX/EBT8iUWjkd7BUxFHt73caODNDz4dADz1j27aL4Zgnhl9m4jkOiSnT7nUvc8WuUk/o8
xg5YOPJGjhgQ1muowdbg56QVYS0keaYkD6Er1xIB/gKQgr4ws/YUJ4JaaSr7wl5RUnn2PKNz5PF
a0rqRnhYXtaV1CNwJQVhECX1KFxJO6HAdBS3KBpKmoSqK+nRF+1jNgDQPGaDYklqEFM9Zh
PFR2SbhM1dH44/FrIHgpuE99gjZG5Tc7LOas+F+FwGTrv4y++fdqH5c4BMin1gjelY5UnARpKDMv
YAxrJzzgNkBRUT2RZhm6z0oDPgakDEypPOc1SftqeZErWjzkzJVUCN2wdYpmEHllnXFrdUQZi4
GBWfhh2oZF1b3HjthDongjMUePdKIRwgVNmgQ4hkh7DYIASft868YJCQh8i3B7HdQTHyeAhjo7
uDO3Jsd1CkT4gwfp/dQVE8ltLKcUxE+fQGjsgeEvF5jOwOerJGdgfzR+s8W4MKWnm2rpXmolQ+g
K9cQgf4ChrKhncHswfsAF19Ea08YTfJVMh8mqzqM3ZMJLuDNsCost3BqhMU+ZZe1QmKbHew6g
TxUOdMIO0OVp2gyHcHCyflOruDANDcHUQxIRU2vDtlw/LqvF28NV+RJfHHRvZQbPb8oHu9ru2n
qVqrPTjisxk6QOhJAFiEhOwANCNm3fToEUlqyyPQzUwHFe0BxGWnolc485lGOeM9ztKj0lCy/jFC
KsvD0LOM+eYfQFjtPLQbbfhOoUd0VI2N7BR6VEfV8FBgaujuFCahqqrmOwM5SAgQ7YOEOMhn
k4DqBwmjHNNlR1Q2ay1lc/dF4gej90insOWROq/Doto4Ycir7acbcbl7/8a97+3u5tbsbyNvvYd8sv1Nr

[illegible]

EiExQQYTUWEHInEUMoGRoQgjQrHBFVLR8CQzYnKCCQoWFxgZGiUmJygpKjQ1Njc4OTpDRC
 EVGR0hJSINUVVZlZXWFlaY2RlZmdoaWpzdHV2d3h5eoOEhYaHilmKkpOUlZaXmJmaoqOkpaanq
 KmqsR00tba3uLm6wsPExcbHyMnK0tPU1dbX2Nna4eLj5OXm5+jp6vHy8/T19vf4+fr/xAAfAQADA
 QEBAAQEBAQEBAQAAAAAAAAQIDBAUGBwgJCgv/xAC1EQACAQIEBAMEBwUEBAABAncAAQID
 EQQFITEGEkFRB2FxEylygQgUQpGhscEJlzNS8BVicEKFiQ04SXxXfXgZGiYnKCkqNTY3ODk6Q
 0RFRkdISUpTVFVWV1hZWmNkZWZnaGlqc3R1dnd4eXqCg4SFhoeliYqSk5SVlpeYmZqio6Slpqe
 oqaqys7S1tre4ubrCw8TFxsflycrS09TV1tfY2dri4+Tl5ufo6ery8/T19vf4+fr/2gAMAwEAAhEDEQA/A
 Pf6KKKACiigAooooAKKKKACiigAooooAKKKKACiigAooooAKKKKACiigAooooAKKKKACiigAoo
 ooAKKKKACiigAooooAKKKKACiigAooooAKKKKACiigAooooAKKKKACiigAooooAKKKKACiigA
 oooooAKKKKAERj1/Xrfw7p/267hmeAHDNE m7b9a1qhubaK7tPlJ0V4pFKsrDlIpO9tC6bipJzV0cL/wt
 /w56XX/AH6pf+Fv+HPS6/79V5p478Gy+F9SMkCs2nTH90/90/3T/SuQrz54mrB2Z9thchy/E0IVpt2f
 me9f8Lf8Oel1/wB+qP8Ahb/hz0uv+/VeC0VP1uZ0f6sYPu/vPev+Fv8Ahz0uv+/VH/C3/Dnpdf8AfvB
 aKPrcw/1Ywfn9571/wALf8Oel1/36o/4W/4c9Lr/AL9V4LRR9bmH+rGD8/vPev8Ahb/hz0uv+/VH/C3/
 AA56XX/fqvBaKPrcw/1Ywfn9571/wt/w56XX/fqj/hb/Alc9Lr/v1XgtFH1uYf6sYPz+896/4W/4c9Lr/v1R
 /wALf8Oel1/36rwWij63MP8AVjB+f3nvX/C3/Dnpdf8AfvB/AlW/4c9Lr/v1XgtFH1uYf6sYPz+896/4W/4
 c9Lr/AL9Uf8Lf8Oel1/36rwWij63MP9WMH5/ee9f8Lf8ADnpdf9+qP+Fv+HPS6/79V4LRR9bmH+rGD
 8/vPev+Fv8Ahz0uv+/VH/C3/Dnpdf8AfvB/KPrcw/1Ywfn9571/wALf8Oel1/36o/4W/4c9Lr/AL9V4LR
 R9bmH+rGD8/vPev8Ahb/hz0uv+/VH/C3/AA56XX/fqvBaKPrcw/1Ywfn9571/wt/w56XX/fqj/hb/Alc9L
 r/v1XgtFH1uYf6sYPz+896/4W/4c9Lr/v1Sf8Lf8Oel1/36rwaij63MP9WMH3f3n0n4c8caV4ou5baxE2
 +JN7b0wMUV5v8ABn/kP3//AF7j+dFdtGcpwuz5HNcJDC4l0qeyse30UUUVseYFFFFABRRRQAUU
 UUAFFFFABRRRQAUUUUUAFFFFABRRRQAUUUUUAFFFFABRRRQAUUUUUAFFFFABRRRQAUUUUUAFFFFABRRRQA
 UUUUAFFFFABRRRQAUUUUUAFFFFABRRRQAUUUUUAFFFFABRRRQAUUUUUAFFFFABRRRQA
 UUUUAFFFFABRRRQAUUUUUAFFFFAGfq+k2mtabNZXkYeKRcH1HuK+cfFHhq68Mas9ncAtGTm
 KXHDr/jX07WB4s8MWvifSHtZgFmUboZcco3+Fc9eiqi8z2snzWWCq2l8D3/zPmWirmp6bdaRqM1j
 eRmOaJsEEdfce1VK8ppp2Z+kU6kakVKL0YIFFFBQUUUUAFFFFABRRRQAUUUUUAFFFFABRR
 RQAUUUUUAFFFFABRRRQAUUUUUAFFFFABRRRQB6X8Gf+Q9ff9e4/nRR8Gf+Q9ff9e4/nRXq4b+
 Gj844g/3+XyPb6KKK6DwgoooooAKKKKACiigAooooAKKKKACiigAooooAKKKKACiigAooooAKKK
 KACiigAooooAKKKKACiigAooooAKKKKACiigAooooAKKKKACiigAooooAKKKKACiigAooooAKK
 KKACiigAooooAKKKKACiigAooooAKKKKAOK8feC4/E2mma3VV1CBSYm/vf7J9q+fpoJbad4Jka
 OWM7WVhyDX1qa8v8AiZ4FOoRtrWmx/wCkxrmaNR/rB6j3rxNDmXNHc+oyHN/YS+r1n7r28jxail5
 BwaSvOPvE7q4UUUUUAFFFFABRRRQAUUUUUAFFFFABRRRQAUUUUUAFFFFABRRRQAUUUU
 AFFFFABRRRQB6X8Gf+Q9ff8AXuP50UfBn/kPX3/XuP50V6uG/ho/OOIP9/l8j2+iiiug8IKKKKACiig
 AooooAKKKKACiigAooooAKKKKACiigAooooAKKKKACiigAooooAKKKKACiigAooooAKKKKACii
 gAooooAKKKKACiigAooooAKKKKACiigAooooAKKKKACiigAooooAKKKKACiigAooooAKKKKA
 CiiigApCAQQRkGlooA8T+JngX+z5G1vTlv9Hc5ul1H3D/AHh7eteY19azwx3ELwyoHjcYZSMgivn3
 x94Mk8M6k09upOmztmNv7h/umvOxNC3vxPt+H849olhqz16P9DjKKKK4z60KKKKACiigAooooAK
 KKKACiigAooooAKKKKACiigAooooAKKKKACiigD0v4M/8h6+/69x/Oij4M/8Alevv+vcfzor1cN/DR+
 ccQf7/AC+R7fRTSwHUgUnmL/eH510Hh2Y7mjmm+Yv94fnR5i/3h+dFwsx9FM8xf7w/OjzF/vD86V
 wsx3NHNN8xf7w/OjzE/vD86dwsx9FM8xf7w/OjzF/vD86Asx9FM8xf7w/OjzF/vD86Asx9FM8xf7w/Oj
 zF/vD86Vwsx9FM8xf7w/OjzF/vD86dwsx3NHNN8xf7w/OjzF/vD86Asx9FM8xf7w/OjzF/vD86Vwsx9
 FM8xf7w/OjzF/vD86dwsx3NHNN8xf7w/OjzE/vD86Asx9FM8xf7w/OjzF/vD86Asx9FM8xf7w/OjzF/v
 D86LhZi6KZ5i/3h+dHmL/eH50XCzH0UzzF/vD86PMX+8PzouFmO5o5pvmL/AHh+dHmL/eH50XC

zH0UzzF/vD86PMX+8PzpXCzHc0c03zF/vD86PMX+8PzouFmPopnmJ/eH50eYv94fnTuFmO5o5p
vmL/eH50eYv94fnQFmPopnmL/eH50eYv94fnRcLMdzRzTfMX+8Pzo8xf7w/Oi4WY+imeYv94fnR5i
/3h+dFwsx3NHNN8xf7w/OjzF/vD86Vwsx9FM8xf7w/OjzF/vD86dwsx9FM8xf7w/OjzF/vD86LhZj6K
Z5i/wB4fnR5i/3h+dAWY+imeYv94fnR5i/3h+dFwsx3NHNN8xf7w/OjzF/vD86LhZj6KZ5i/wB4fnR5i/3
h+dFwsx9FM8xf7w/OjzF/vD86LhZjuaOab5i/3h+dHmL/AHh+dK4WY+imeYv94fnR5i/3h+dFwsx1U
dV0u11jTprG7jDxSqQcjk496ueYv8AeH50eYn94fnQ7Naji5RfNHdHzN4r8M3PhfV2tZgWhfLQSD
mX/GsGvpzxP4esvE+kyWdxjF8AeikHVG7EV846vpN3ompTWF7GUljOM9mHYj2ry8RR5HdbH6Jk
ubLGU/Z1PjX4+ZRoornPeCiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigD0v4
M/wDlevv+vcfzoo+DP/levv8Ar3H86K9XDfw0fnHEH+/y+Rv/ABjvLm0sdMNtcSwlpWBMblc8e1eR/
wBs6p/0E7z/AL/tjXq3xr/AOPDSv8Arq38q8brjXTaqOx9Nw9SpywMXKN9/wAy9/bOqf8AQTP+/7f
40f2zqn/AEE7z/v+3+NUaK5+Z9z3PYUv5V9xe/tnVP8AoJ3n/f8Ab/Gj+2dU/wCgnef9/wBv8ao0Ucz7
h7Cl/KvuL39s6p/0E7z/AL/tjR/bOqf9BO8/wC/7f41Roo5n3D2FL+VfcXv7Z1T/oJ3n/f9v8aP7Z1T/oJ
3n/f9v8ao0Ucz7h7Cl/KvuL39s6p/0E7z/v8AtjR/bOqf9BO8/7/ALf41Roo5n3D2FL+VfcXv7Z1T/oJ3n/
f9v8AGj+2dU/6Cd5/3/b/ABqjRRzPuHsKX8q+4vf2zqn/AEE7z/v+3+NH9s6p/wBBO8/7/tjVGijmfcPY
Uv5V9xe/tnVP+gnef8Af9v8aP7Z1T/oJ3n/AH/b/GqNFHM+4ewpfyr7i9/bOqf9BO8/7/tjR/bOqf9BO8/
7/tjVGijmfcPYUv5V9xe/tnVP8AoJ3n/f8Ab/Gj+2dU/wCgnef9/wBv8ao0Ucz7h7Cl/KvuL39s6p/0E7z/
AL/tjR/bOqf9BO8/wC/7f41Roo5n3D2FL+VfcXv7Z1T/oJ3n/f9v8aP7Z1T/oJ3n/f9v8ao0Ucz7h7Cl/K
vuL39s6p/0E7z/v8AtjR/bOqf9BO8/7/ALf41Roo5n3D2FL+VfcXv7Z1T/oJ3n/f9v8AGj+2dU/6Cd5/3/
b/ABqjRRzPuHsKX8q+4vf2zqn/AEE7z/v+3+NH9s6p/wBBO8/7/tjVGijmfcPYUv5V9xe/tnVP+gnef8
Af9v8aP7Z1T/oJ3n/AH/b/GqNFHM+4ewpfyr7i9/bOqf9BO8/7/tjR/bOqf9BO8/7/tjVGijmfcPYUv5V9
xe/tnVP8AoJ3n/f8Ab/Gj+2dU/wCgnef9/wBv8ao0Ucz7h7Cl/KvuL39s6p/0E7z/AL/tjR/bOqf9BO8/w
C/7f41Roo5n3D2FL+VfcXv7Z1T/oJ3n/f9v8aP7Z1T/oJ3n/f9v8ao0Ucz7h7Cl/KvuL39s6p/0E7z/v8At
/jR/bOqf9BO8/7/ALf41Roo5n3D2FL+VfcXv7Z1T/oJ3n/f9v8AGj+2dU/6Cd5/3/b/ABqjRRzPuHsKX8
q+4vf2zqn/AEE7z/v+3+NH9s6p/wBBO8/7/tjVGijmfcPYUv5V9xe/tnVP+gnef8Af9v8aP7Z1T/oJ3n/A
H/b/GqNFHM+4ewpfyr7i9/bOqf9BO8/7/tjR/bOqf9BO8/7/tjVGijmfcPYUv5V9xe/tnVP8AoJ3n/f8Ab/
Gj+2dU/wCgnef9/wBv8ao0Ucz7h7Cl/KvuL39s6p/0E7z/AL/tjR/bOqf9BO8/wC/7f41Roo5n3D2FL+
VfcXv7Z1T/oJ3n/f9v8aP7Z1T/oJ3n/f9v8ao0Ucz7h7Cl/KvuL39s6p/0E7z/v8AtjR/bOqf9BO8/7/ALf4
1Roo5n3D2FL+VfcXv7Z1T/oJ3n/f9v8AGj+2dU/6Cd5/3/b/ABqjRRzPuHsKX8q+4vf2zqn/AEE7z/v+
3+NH9s6p/wBBO8/7/tjVGijmfcPYUv5V9xe/tnVP+gnef8Af9v8aP7Z1T/oJ3n/AH/b/GqNFHM+4ewpf
yr7i9/bOqf9BO8/7/tjR/bOqf9BO8/7/tjVGijmfcPYUv5V9xe/tnVP8AoJ3n/f8Ab/Gj+2dU/wCglef9/wBv
8ao16P8ADXwOdWuV1fUlv9DibMSMP9YfX6CrpxnOVkzxtXDYSi6s4o6j4beGtSjiTWdWvLtncluJ
JWIA9SM9a2fHng2PxPpZaEKt/AMxP/AHv9k+1dgqhFCqMKBgCINeoqS5ORn51PMKrxP1mOj6H
yXcW8trcSW86FJY2Ksp6gioq9u+JXgT+1YW1jTY/9NiH72Mf8tV/xFeJdDg8Y4NeZWpOnKx+h5Z
mMMdR51ut0JRRRWR6QUUUUAFFFFABRRRQAUUUUUAFFFFABRRRQAUUUUUAFFFFABRRR
QB6X8Gf+Q9ff8AXuP50UfBn/kPX3/XuP50V6uG/ho/OOIP9/l8jZ+Nf/HhpX/XV/5V432r2T41/wDHH
pX/AF1f+VeN9q4sV/EZ9Xw5/uEfV/mFFFFc57gUUUUUAFFFFABRRRQAUUUUUAFFFFaOnaFqurOq
2FhPPk43Knyj6noKaTexFSrCmrzdkZ9JWjrOi3ug3ws79FScxiQqrZwD/8AqrOpNNOzCnUjUipQd0
wooooLCiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooo
oAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKAfOpK2fDPh268TavHZW4K
p1lkxwi+v1pxi5OyM61aFGDqTdkjS8DeD5/FOqKXUrp8LZmk9f8AZHua+iLW1hs7aO3gRUijXaqqM
ACqmiaNaaFpcVhZxhlox+JPcn3rRr1qNJU4+Z+aZrmU8dWv9lbIWiiitjyhpGRg9DXjXxN8DG1kk1z
TYv3THNzGo+6f7wH8/wA69ngKWKOEJo5FDlwwQRwRWdWmqkbM7cBiqmDrKpD5rufJVFdt8Qv

BT+G9QN3aqW02dvlOP9W390+3pXFV5E4OErM/TsJiqeKpKrTejEoooQpTpCiiigAooooAKKKKACiii
gAooooAKKKKACiiigAooooA9L+DP/levv+vcfzoo+DP/ACHr7/r3H86K9XDfw0fnHEH+/wAvkbPxr/4
8NK/66v8AyrxvtXsnxr/48NK/66v/ACrxvtXFiv4jPq+HP9wj6v8AMKKKK5z3Aoop8cUksixxRtJlxwqq
Mkn6UCckldje1JXbaV8LvEWohJJY07SJuplbLY+g/wAa3l+Cl2RltZiU+n2cn/2atlh6j6HmVc6wNN2l
UX5nldFelXnwa1aKPda39vO391IKfrk1xuteF9Z0BsahZPGmeJV+ZD+l6fjUyojzujWhmeExDtTmmz
HpaStvw74V1LxPJOMnClCAX8xtvXOOx9KiMXJ2R1Va0KMOeo7IPCEEVx4t02GaNZl3mwyMMg
8GvpqGGKGMJFGqKBwFGK8Z8NfDPxBpniOxvrkW3kwybn2yknH5V7UOIelhYOMXzl+C4ixdPEV
oulK6seA/Fv/AJHh/wDr2j/rXCV7H4+8A634i8TNf2ltzCYVT95IQcjPt71yzfCbxOqklbTAGf8AWn/Cu
WrRm5tpH0WWZnhKeEpwnNJpHC0tPmiaGaSF8bo2KnHqKjrmPfTTV0FFbWieFNZ8Qt/xL7NmjB
5lc7UH49/wzXb2fwXvpFVrvVloj/EqRFv1z/StlUzy2RwYjNcJh3apNXPLqK9db4KLtOzWGDY4zFx/
OsbUvg/rdqu6zulLvurjYfw6/wBKp4aouhhTz7ATduf7zzuird/pt7pdy1vfW0KEoP3XXGfoe9VKxaa0Z
6sJxmuaLugoorq9J+Huva1psV/ZxwGCUZUtJg/ypxhKT0Mq+JpYePNVIZHKUV0+teAtc0DTnv76O
EQIQDskyeTj096xdO0q/wBWn8mwTJJ377BwPqegpuEk7NCp4yhUp+0jJcvcpUV3EHwn8UTIGMd
rFkfdeXkfkKxPEnhTUvCslumomHdOGKeU5bpjPUe9N0pxV2jKlmoFqz9nTmmzCorT0PQR3xDqH
2KwCGbYXw7YGBj/ABrpP+FUeKB/yytv+/v/ANaiNOcldlqtj8NQnyVJpM4ilqxeWM9jfy2UygzxPsZV
55rodM+HniXVY1ljsfJjYZVp225/DrSUJN2SKq4yhSipzkkmtSV2958LPEVjZT3cxs/LhQyNiU5wBn
OriKJU5Q+JBh8ZQxN3RlewtJT4opJpVjijeSRjhVRSSfwFdtPHwr8Qakiy3Cx2UbDI8w5b8h/jRGnK
WyDE42hhlerJl4aivW4fgodn77WDu/2lCd+ZqO4+Cs45ttXU8dHh/rmtfq1TseculMA3bn/AAZ5RRX
Va18PvEOil0stp58C9ZIDuwPcdf51y3OeRWUoSjo0enQxNLER5qUk0JRRRUM4UUVv6P4N17XQ
Hs7F/KPSWQ7F/XmnGLlokZVq9KjHmqSsvMwKK9Fg+DuuPHulurWNv7oJb/Cqt78JvElqGaFbe5
UDgl+GP4Hj9a09hUS2OGOC4GUrKojhKKsXtjdadctb3lvJBMvVHXB/D1HvXU2fwy8R31nDdQR25i
mQOpMmDgjl7VMacpOyR01cbh6UVKc0k9jj6St3xD4T1TwwLc6kkS+fu2bH3dMZ7e4rCqZRCXZm
1GtTrQ56bugooopGgUUUUAAAAFABRRRQAUUUUAAAAFABS96SpIYZLiZiYkLySMFRR1JoWo
pNRV2T6dp1zq1/FZWcRkmlOAo/nX0b4R8L23hjSI7aMBp2G6aXHLN/hWT8PvBSeGtPFzdKG1G
dcu2PuD+6K7evTw9DkV3ufnueZu8XP2VN+4vxFooorqPngooooAKKKKAKWqaba6vp81leRLJBK
uGU/wA/rXzl4t8LXXhbVnt5QWtnJMEuPvL/AlivpmsfxH4etPEeky2V0o5GUfHKN6isK9FVI+Z7GUZ
pLBVdfhe/+Z8vUVo65ot3oOqS2F4hV0Pyt2dexFZ9eU007M/SadSNSKnB3TEooopFhRRRQAUU
UUAAAAFABRRRQAUUUUAAAAFABRRRQB6X8Gf+Q9ff9e4/nRR8Gf8AkPX3/XuP50V6uG/ho/O
OIP8Af5fl2fjX/wAeGl9dX/IXjfavZPjX/x4aV/11f8AIXjfaULffGfV8Of7hH1f5hRRRXOe4OVWd1RFL
MxwFA5Jr6B8CeB7bw7p8dzcxrJqUq7pHlzs/2RXkHgG2ju/HOIRSruXzS+D6qpYfqBX0oOld2Eppr
mZ8bxPjakZRw8XZNXZVv9Rs9LtzCxtxHBCvVnYAVykvxU8LRSFBeu+DglYXx/KvOPixqF1c+L5L
SVmFvbovlJ2ORkn684rhO9OripRlyxJy3h2lXoRrVZPXsfT0jeMtB1+Ty7C/R5f+ebAq35Gtm4tobuB
obiJJl2GGVhkEV8owXETrcR3EEjRyxsGR1OCDX0jpni/R7jS7WebU7WOSSJWZGIUEEjkVrQr+0V
pHnZtk7wMoyottP70eO/ETwknhrV1ltFlsbnJjH9w9xWI8KNa03RrrUm1G8ithlqBDI2M4z/Al1vfFvD
K1Lw5Atpe288yTggRyBiBg5rxuuSo1Sq3ifSYGE8xy32Ve6e3nofTtr4v8AD97cx21tq1rJNlCkiYAkmt
ztXzL4K/5HPSv+u39DX00OgrtoVXUjdnyecZdDAVVCdvdXMjUPE+i6Vc/Zr7Ure3mADbJHAOKpS
eOPDjiYDWrMkg/8tBXkvxb/AOR4b/r3j/rXCVhUxUoycbHs4HhyjiKEKzm02rli+dZNQunQgq0zEEd
xmuh8B+Fx4o14RTZ+xwDfOQevov4/0rlq9Y+FWs6Pouj3bX97BBNNPwGPO0Af1zXNRSIU9497N
atTD4N+x1eyPWrw0gsbaO3t4kjjXCqowAK5nxF8Q9E8OzNbyyvcXS9YYRkj6npT9R8daDHP1w9t
qkDTCNigz1OOK+dJ5pLmeSeZy8kjFmY9ya7K9fkVonymUZM8ZOU8RdJfie0Q/GfSXICy2F2iE/ew
px+tdxoviDTPEFr5+nXKzIOGGCCv1B5r5brsPhnqM1j42s4o2xHc7o5FzwRgkfqKypYqTlaR6eY8O
4eFCVSjdNK57d4k8NWHITTLJLW7iG7BMcgHzl3qDXzZqVhNpepXFjcDEsLID7+/4jmvq7tXzz8Uo1

j8c3O0Y3Rox9zirxkFy8xycL4qarOg3o1c4yvvor4ZHPgWwz2Df+hGvnWvor4Zf8iLYf8C/9CNZYP4z0
uKf91j6l3xpo02v+HZNnHIV5pYwWP8ACocEn8gavaJoVhoFglpZQqiKPmbHLH1J9a02IAJPAArxjx
p8T7xr6bT9EkEUMTFGuAMsxHXHt7121JQp++z5TBYfE41fVqXwrV9j2beo6kfnXjvvrOb3R8YP7u
X+a153Nr2r3Ehkl1O7Zj1/fMP5VXur+7vUjW6upZhECE8xi23PXk/SuOriVOPKkfUZbw/UwmljWINO
x2nwj/5HU/8AXs/81r3w/drwp4Rf8jr/ANuz/wA1r3ztXRhP4Z4vEn+/fJHF6D4It7bXr/W76JZLqadmh
DciNfX6muyG0DHAFcn458Zx+FLBBGglvZuloyeAPU14lqPjLxBqkjPcanOoY52RNsA/KidaFL3RYX
K8Xma9rJ2jsr/ofQviRlPhjVMEH/RJe/8Asmvlyr661qih1/tG6KupVg0pIIPUYNQ6ekUmo2yTuEhaRQ
7HsM8muOtWVVo+oyrLJ5bCfNLmv+h7h8NvBsGkaVHqV1EGv7hd2WH+rU9AK7a+v7XTLOS6vJ
khgjGWdjwKxovGnhqOJUJGrW4CjAGa8z+K3ie11eSxs9PvFmt0DSSBDwW6DP6/nXa5wpU/dPko
YXE5jjP3qau97bl6e8+MmiwzMlta3Vwg6SBQoP5kGr2jFTQNUmEMzS2UjHC+eOG/EZH514BRXI
sXO59PPhnBuFo3T73Prf5ZE4wykfnXjfxU8HQWKjXLCIRo7bbhFHAJ6NXY/C7UpdR8GQiZ2d7d2
h3MeSBYP0IFaXju3juPBeqLIoIEJcfUcj9RXZNKpTufK4OpUwGPUE9nZ+Z809qKK6XwJoQ1/wAV
W1tKMwR/vZeOoHQfia8uMXJ2R+j4ivGhSIVlskdp8PPhzHPDHrGtRBIb5oLZhxj+83r9K9cCxxwR4A
VEUfQCnlxofUYUDAFepfFbxfP9r/sKylaONBm4ZTgsT0WvU9yhA/OI9YzjF8rf+SR3t/8AEDwxxpkp
iuNUj8wHBWNWcg/8AAQak07xx4c1aRY7XU4mkb7qOCjH8Gwa+aKXnqK5frkr7H0L4VocllN3Ppv
xH4Y07xPpzW91Gu7GY5lHlzfUGtDSbl6dpFnZMwZolUjLDocADNeS/DTx1cR38Wi6nM0sMp2wS
OclW/uk+leziuyIOM1zl+WzDDV8JP6vVd0tUeR/G77ui/Wb/ANkryKvXfjd93RfrN/7JXkVefiv4rPt+Hf8
Acl/P8woornPcCiiigAooooAKKKKACiiigAooooAWvZfhj4G+yRjrmpRf6Q4zbxSPuD+99TWJ8NfA
p1KZNZ1KP8A0VdmGNh/rD6/SvbVAUAAyA6Cu/DUPtyPjOIM4vfDUX6v9BR0paKK7j44KKKKAC
iiigAooooAKQ0tFAHleO/B8XijSj5eFvORuhf1/wBk+xr55uLea0uZLa4jaOaNirow5Br60rzr4j+BV1m3
bVNPjAv4l+cAf61R6+9cmJocy5o7n0uQ5v8AV5ewqv3Xt5HhdFKQVYqwlYHBBHskrzT79O+qCiiig
AooooAKKKKACiiigAooooAKKKKACiiigD0v4M/8h6+/69x/Oij4M/8Alevv+vcfz0r1cN/DR+ccQf7/AC
+Rs/Gv/jw0r/rq/wDKvG+1eyfGv/jw0r/rq/8AKvG+1cWK/iM+r4c/3CPq/wAwoornPcNvwjQeKeK9O
vpiBHHLh2PYEFSf1r6bjdZl1dDIWQQRXyV2r0Lwf8TrnRII7DU0e5tE4RwfnQf1FdeGrKHuyPI+IMq
qYlqtR1a6Hpfj3wJp3isLLMzQXaLtSdOuPQjuK8y1P4Q6/aF2spLe8jA4AbY5/A8frXrGleM9B1hB9m
1CIOf4HO1gfoa3ldHGVEeoNdcqVOpqfNYfMsdgP3a0S6NHy9qPhrWtJyb7TLiFR1cplfzHFZWCD
gg5r62KKw5UH8K53W/A+ha5G3n2Uccp/5axDa2fXjrXPLB/ys9rD8U3aVeHzX+R81UV0vjDwhdeF
NQEbkY2suTDLjr7H3rmq4pRcXZn1uHr069NVKTumb/gn/kc9K/67f0NftQ6CvmXwT/yOelf9dv6Gv
podBXoYP4GfE8Vf7zD0PAfi5/yPD/8AXvH/AFrhK7v4uf8AI8P/ANe8f9a4SuKv/EZ9XIH+40/RC96SI
96774feAf8AhIm/tHUAy6ejYVBwZSP6VMIObsjGYulhaTq1XocjpehaprUmzTrGa4PcqPIH1J4FddZ
/CLxHOFaZ7S3U9Q0hLD8AMfrXuVpZ21jbrBbQpFEgwFQYArH1fxpoOiuYru/jEo6xp8zfpXesLTirz
Z8dU4hxlefLho2+V2ecD4LXpGTq0IPceUf8a0tA+FN5o2v2WonUopVgk3MnlkEjGOOa03+MHh4H
CR3bD18vFWtH+J+j6zqlvp8MNys07bV3Jx0z/SmoUE9DOtis5ITI7RPItrp0O4r58+Kv/I9T/8AXJP5V
9B18+fFX/kerj/rkn8qeL/hkcM/778mcTX0T8Mv+RFsf+Bf+hGvnavon4Zf8iLY/wDAv/QjXPg/jZ7fFP8
AusfU1vF16+n+E9Tuom2yR277T6HHFfMPNfSXxB/5EbVf+uP9a+bKeNfvJGfCkV7GcutwoorjPqz
vfhF/wAjr/27P/Na987V4H8lv+R1/wC3Z/5rXvh6V6mE/hn53xJ/vz9EfPXxSvHuvG06MxKQlqKPTjJrj
K6Dx0SfG+rc/wDLc/yFc9Xn1XebPtstgoYSml2QvaikqeztZr67itbdDJNKwVFHc1mk3sdspKMxKWx
EiM7hUUsxOAAM5NdRpnw68TamqvHp5gjYZD3DBP06/pXrng3wFY+G7ZJ5kWfUGHyzsM7fZfSu
qu7210+AzXU8cMa9WdgBXdTwritebPjcbxLLn5MLH5/8AAPGrb4M6vlubnULWJvRAz/4VY/4Uref9
BeL/AL8n/Guwwfip4ZtH2JcSXB9YoyRWefjFoOQPlusf7g/xq/Z4dHOsdnc1zKL+43vA/hibwrpEtjNcJ
OWmMgZVxwQB/SrXjT/kTtV/69n/AJGpfDniOz8Tac19ZLI1cxkOMHlx/jUXjT/AJE/Vf8Ar2f+VdFlye
7selpVJYxOr8XMr/efMtet/BWzB/tS8l5ykY9sZP8AWvl+lez/AAV/5BGpf9dx/wCgivOwutVH3XEms

vlbyPUJG2xs3oM18s67dPfa/f3LnJed+fYHAr6kuP+PaX/AHDXyje8X9wCP+WrfzrfGvRHi8KRXtKk
uuhBRRRXAfbE1rM9tdwzxtteNwyt6EGvq21k861ikH8aA/pXycn31HuK+rNMUx6XaqeqxKP0ruwT3
PjOK4q9N+p5d8bvU6L9Zv8A2Svlq9d+N33dF+s3/sleRVhif4rPY4d/3CPz/MKKKKwPbCiiigAooooA
KKKKACiiigArsPAfgyXxRqQlnVl06BsyN/fP90f1rK8LeHLNxnRnEdnApEY+aaXHCL/jX0fpGk2ui6bDY
2kYSKNcDA6+5rqw1DnfM9j5vPc3WGh7Gk/ff4Fq2t4rWBIIUVl0UKqqMACpaKK9M/P223di0UUU
AFFFFABRRRQAUUUUAAFFFFABSEailooA8c+J3gYxPJrumQ5Q83Ma9v9of1ryivraSNZY2jdQyM
MEHuK8w1H4OWl1qE89ftbwyNuWLYDt9hXDXwzk+aB9fk+fwpu/ZYl7bP9DxiivXv+FKL/0Fn/79ij/
hSi/9Bd+/Yrn+rVOx7f+sOA/n/A8hor17/hSi/8AQXf/AL9ij/hSi/8AQXf/AL9ij6tU7C/1hwH8/wCB5DRX
r3/CIE/6Czf9+6P+FKL/ANBd/wDv2KPq1TsH+sOA/n/A8hor17/hSi/9Bdv+/Yo/4Uon/QWb/v3R9Wqd
g/1hwH8/4HkNFevf8KUX/oLP/wB+xR/wpRf+gu//AH7FH1ap2H/rDgP5vwZ5DRXr3/CIF/6Czf8AfsUf
8KUX/oLP/wB+xR9Wqdg/1hwH8/4HkNFevf8ACIF/6C7/APfsUf8ACIF/6C7f9+xR9Wqdhf6w4D+b8
GZnwZ/5D1//ANe4/nRXb+DfAA8J3890t6Z/NjCbSmMc5or0KEXGCTPjM3xNPEYUvSm9HYwfjX/x
4aV/11f+VeN9q9k+Nf8Ax4aV/wBdX/IXjfaUDfXGfZcOf7hH1f5hRRRXOe4TQ209yxW3hklIGSEQn
FT/wBk6j/z4XX/AH5b/Cuz+EV0kPiyW3kKgTWEAHuWlP8ALNe7IFx9wflXXRwyqR5rnzOZ59UwWl
dJQv8AM+S2SSGQqyskinBBGCDWIY+I9Z00r9k1K5jCnlUOSPyroPiZoU+l+Kri7ERFrDnzEcDjd3H
1zzXFVhJSpysexh50cbQjUaTuei6N8XNYs5ETU40vlc/MwG1wPw4r2jTNRg1XTYL61bdDMgdTXy
rHG8sixxqXdjhVUZJNfS/gvS5NG8JafZTAiVl9zg9ixLEfrXZhak5XTPIOl8DhcOozpK0m9jO+JOMJq
Pgy8YqDJAPNQtjr+ma+dq+j/iJfrY+CtQYkbpE8tQT1LHH8q+cKyxiXMj0eFnL6tJPa5v+Cf8Akc9K/
wCu39DX00OlFMngshfGWlknA88fyr6bHQVtg/hZ5nFX+8w9DwH4t/8Al8P/ANe6f1rhe1ejfGHT5Yv
EcF9sbypoQm7HAKn/AOvXnFcddP2jPp8mnGWBp2eyJbeFri5igU4aRwgPuTivqXRtPi0vSLWzhU
KkUYUY+lfMGluser2TtgKtwhOf94V9VxEPEjDoVGK6sElqzwOK5yvTj01PP/ip4nuNF0qGxspWiubs
nLr1VB1x+Y/OvC2Zncs7FmJySTkmvaPjBoU97p1rqdujSG1JWRVGcKe/6CvFaxxTlz67Ho8Nxo/V
E4/F1Cuq+HMDz+O9N2LklzOx9Btl/mRXLdTXsnwn8Jz2Ik1u9iMbypshRhyF7n8eKihBymrHZnOK
hQwk7vVqy+Z6p2r58+Kv/l9XH/XJP5V9B18+fFX/AJHq4/65J/Ku3F/wz5Phn/ffkzia+ifhl/yItj/wL/0l18
7V9E/DL/kRbH/gX/oRrnwfxs9vin/dY+pa+IX/ACl2q/8AXH+or5s7V9J/EL/kRtV/64/1FfNnanjfiRPCn+
7z9f0CiiiuM+pO9+EX/I6/9uz/AM1r3w9K8D+EX/I6/wDBs/8ANa98PSvTwn8M/O+JP9++SPmfxz/yO
+r/APXc/wAhXPV0Pjn/AJHfV/8Aruf5Cuerz6nxs+5wH+60/RfkFek/B3SYrvXLU/kAY2iAICOHbPP5A/
nXm1ev/BS4XyNUtv4ldH+oll/pWmGSdRXOHPpyhgZuJ6w7BEZj0AzXzV4v8TXfiTWppZZX+zl5W
GLPCrn+dfSkqeZE6H+IEV8v+l9GutC1u5s7IGGHJRiOHXPBFdWLcuVWPnOGl0XXk5/FbQyaKK0
dG0a917Uo7KyILu5+ZsclPU156Tbsj7irVhSg5zdkj234S23keCo5Of38zvz+X9K3fGn/ACJ2q/8AXs/
8qvaLpkWjaPa6fCPkgjC5x1Pc/iao+NP+RP1X/r2f+Veuo8tO3kfl8qyrY72i6y/U+ZO9esfBa9CzanZM
Rk7JF9+oP8hXlHet7wbrn/CPeJ7W9b/VE+XL/ut/9fB/CvMoy5aiZ+hZrh3iMHOC3t+R9MsNyEeor5h
8W6a+leKdQtXBA84uue4PNfTcUqTRLljBkYZBHcVwvxD8Ct4khW+sNov4lXg8eYvp9a78TTc46Hx
WRY6ODxLVTSMtDwSir93oup2E7Q3NhcRuvUGM0tnomqX86w21hcSOFSM4/OvN5JXtY/QPrNHI
5uZW9Q0PTpNW1yzsYwSZpVU47Dufyr6miXZEjqoABXBfD7wCPDqf2hqAVtQkXAUHliHoPf3r0C
vSw1Jwjr1PgM/zCGLrpU/hieR/G77ui/Wb/wBkryKvXfjd93RfrN/7JXkVcWJ/is+r4d/3CPz/ADCiisD2
woooooAKKKKACiiigAq7pOlXetajFYWUZeaQ/go7k+1V4lJbm4jghQvLIwVVXqTX0F4B8GxeGdNEs
6htQnGZXx93/ZHtW1Ci6kvl8jN8zjgaWnxPZGp4U8M2vhjSI7SABpiMzS45dq3qKK9ZJJWR+bVas
6s3Obu2LRRRTMwoooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACii
gAooooAKKKKACiiigDyn41/8eGl9dX/IXfavZPjX/AMeGl8AXV/5V432rysV/EZ+kcOf7hH1f5hRRR
XOe4XNL1KfSNUt9QtibNA+5c9D6ivpHw14msfEumpdWso8zGJYifmRvQ18xVbsNSvNLulubG5kg
mX+JGxn6+orehXdN2ex4mb5PHHRUou0kfUI9p9pqds1veW8c8LDBR1yK5O4+FfheeUuLSSPP8

KSsB+VcDp3xg1u2TbeW0F1gcEflfx61pf8LsuP+gln/gQf/ia7HWoy3PmlZTmuHdqV/kzvtG8DaBoc
qzWlivnL0kkJZh+JrfmnitoWlmkWONBlmY4AFeM3Xxn1KSli202CF+zPIXA/DArjda8W63r5xf3rtFnli
T5U/ldfxpPE04K0TSnkGPxM+bEO3q7s3/iP4zTxHepZ2LE2Fuxlf/no3r9K4Silrz5zc5czPtcHhaeFpK
IT2Ra0y8Onara3gz+5lVyB3APNfUenX9vqVhDd20iyRSqGVINfKXatrQ/Fms+HeNPuykROTE43lfw
7fhW2Hrqnozyc7yiWOUZU3aSPpW8sLXUITDd28c0Z/hdcisZfBfhy23SxaRaq4BIOyvLYPjFrqJia1t
JGzwVBX+pqG5+L3iGYssUVnGhGMFCT+ea6niKT1Z87TyPM4PljovU4m/ATUrpVGAszgY7cmvd
Ph14zt9c0qOwuJQuo26bWVj/rAP4hXgssrTTSSvjc7Fmx6mlhnltpkmggleKVTIXRtpH4iuOIWdOV1sf
V5hlkcbh1Tk7SWzPrJ0SVCjqGVhggjg1yWofDTwzqExlax8pycnyXKg/gK830n4ua5YxiO8iivVAwCf
kb8SM/yreHxqh286Rjux0EgxXd7elNanyKyfM8NP91+DOw0z4eeG9KIEsVgJJFOQ0xL4/OrGr+Kb
LStTsdKRle7upVQRr/AvcmvKtW+Lmt30ZjsoYrJWBBIO9vwPT9K4iPUrtNTj1Ezu90kgkEjsSdwOea
yliYR0gjuo5Bi8ReeLl6K9z6ur58+Kv8AyPU//XJP5U3/AlWn4p/5+Yf+/QrmtY1i813UGvr51edlCkquB
xUV8RGpCyOrJsmxGDxPtlrWKFFRPwy/wCRGsP+Bf8AoRr52rp9J8fa/ounx2NIPesEedoame1lh
6ipyuz0s7wFXG0VClun1PbPiD/yIuq/9cf6182V1WpfELxDq2nzWN3PE0Ey7XAJAOK5WjEVY1JJo
WR5fVwVKUKtrt30CiiisD2zvfhH/wAjr/27P/Na987V8saLrd94fv8A7bp7qk2wpllwzf8A9VdF/wALT8U
/8/UP/foV20MRCEOvnyWcZLicXifa07WsjM8c/wDI76v/ANdz/IVz1WtQvp9T1Ca9uWDTzNucgYBN
Va5JtOTaPpsLTdKhCEt0kgro/BfiVvDGvx3bZnu42TqP7vr+Fc5RSjJxd0PEUIV6bpT2Z9YWN9bala
R3VpKssMg3KynINvdW0DTNchEeoWcc4HQsOR9DXzpoPivWPDjk6fdFYycmKQbkP4dq7uz+NM
6qFvNKViBy0cuMn6EV6McVTKrSPHa/D2MoVOahqumtmdUPhP4X8zd9mmxnO3zmxXR2Wl6P4b
snNtBBaQqNztgDj1Jrza6+NLMpFFrpOHPeSXP+lcLr/AlY1rxGdt9c4gzkQxfKn4jv+NJ16UPhRrTyjM8
U1HESaj5s978N+JYPEq3k9qD9nhm8qNj/HgDLfTn9KPGf/ACJ2q/8AXs/8q8EOLxnrPhy0e206aNI3
fewZM81cvviN4j1Kxms7m4iaGZCjgRAZBpfWouNnuW+HMRDEKVO3KmuupylJS1YsbK41K9itL
WPzJ5ThFBxmvP3eh9s5KMby2PQfAfxJ/siJNM1gs1ovEU45MY9D6ivZLLUrPUbdZ7S5imiYcNGw
INfPf/CuPFX/AECm/wC/i/41m3+na34VukS48+xmKXcvly4JH/ATXbCtUpq01ofJYvKcFjavNhqiUn0
Pp1o45PvorfUZpjNb2yFmMcagZJ4Ar5oi8X+IYY/LTWLzb/tSlj+ZrPudU1C9XbdX91Ouc4lmZh+pq3
jl9Ec0OFq7dpVFY9i8YfE+ztY2sNDmW4uXO1p0OUjz6Hua9HtyWt4yepUE18raXfbfNWtLXkebMq
ce5r6sjGI1HoAK0w9SVS7ZxZ3gKOBVOnT1et2eS/G0/wDIFHfMv/steRV6v8a5AbnSY+MhZD+e3/
CvKK4sT/FZ9Xw8rYCHz/MKKKKwPaCiiigAooooAKXknAFJXp/w08Cm+nTW9Si/0ZDm3iYffP8Ae
PtV06bnKyOPHY2ng6LqT/4c2/hn4G/s+BNA1KL/AEqUZhjYf6tfU+hNen0gGBgcClr14QUI2R+YYz
F1MVVdWoLRRRVnMFFFFABRRRQAuuuuAFFFFABRRRQAuuuuAFFFFABRRRQAuuuuA
FFFFABRRRQAuuuuAFFFFABRRRQAuuuuAFFFFAHIPxr/AOPDSv8Arq/8q8b7V7J8a/8Ajw0r/
rq/8q8b7V5WK/iM/SOHP9wj6v8AMKKKK5z3AooooAKKKKACiiigAooooAKKKKACiiigAooooAKKK
KACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAK1PDI9/Z3iTTTrskBY51
3EngAnB/Q1l0U4uzuZ1qaqU3B9T62QhkVl5BGa87+K/hm51fToNQsojLNaZ3ooyzlfT6VB4A+llrdW
UOI6tMIbulBI5HPEo7c+telh0kXlIZSK9b3a0D8y5a+WYpSa1X4nyUQQcGkr6fvvCehalKZbrTLaSR
urmMZP41DbeCvDlnKskWkWwdTkMUBI/OuT6m77n0q4rpcutN3PMPHf4OubvVltbvISlrDzCGHLt6
/QV7d0FM/dwpj5UUfGK8/8AGvxls9KtZbLS5kuL9htJXIYvc+/tXVFQoQPnq9TE5tirxj/wEcD8VNUj1
Hxg8cTbktYxESDxu5J/nXEU+WR5pWlKys7tuZj1JpleXUIzycj9EwWHWGoRpLogoooqTpCiiigAoor
d8K+GrnxRrEdnApEKkNPLjhF/xNOMXJ2RlXrwoU3UqOyRsD/AMFP4lv/ALVdKRp0DfOf+eh/u/Sv
oCGGO3hSKJAilAqgDgCq2l6Za6Rp0NjaRhIYICgDv71dr1qNJU4n5nmmYzx1bmfwRZC0UUVseY
FFFFABRRRQAuuuuAFFFFABRRRQAuuuuAFFFFABRRRQAuuuuAFFFFABRRRQAuuuu
AFFFFABRRRQAuuuuAFFFFABRRRQB5T8a/8Ajw0r/rq/8q8b7V7J8a/+PDSv+ur/AMq8b7V5WK
/iM/SOHP8Acl+r/MKKKK5z3AooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAK

KKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACtew8Ta5pa7LPU7iJey7twH0z
msiimpNbMzqUadRWnG52KfE/wAUouPtiN7mMUr/ABP8UuuPtiL7iMVxtFX7afc5f7Lwf/Ptfca9/wCJ
9c1NSt5qdxlvdd20H8qyKKKhyb3OqnRp0laEbBRRRSNAooooAO1FLUlvby3VxHbwIXmkYKijqSa
EhSkoq72LmjaPd67qcVhZpukc8nso7k19G+GPDlr4a0iOyt1BbGZJMcuc3c1l+BPB0PhfTA0qq1/OA
Zn9P9kewrr69TD0eRXe5+d53mzxdT2dP4F+ltFFFdJ4AUUUUAFFFFABRRRQAUUUUUAFFFFAB
RRRQAUUUUUAFFFFABRRRQAUUUUUAFFFFABRRRQAUUUUUAFFFFABRRRQAUUUUUAFFFFA
BRRRQB5d8ZLW4urDTBb28spWVsiNC2OPavlv7J1L/oH3f/AH5b/Cvqwqp6gHFN2IR9xfyrlqYZVJ
c1z6HL8/ng6CoqF7eZ8q/2TqP/AED7v/vy3+FH9k6l/wBA+7/78t/hX1Z5af3F/Kjy0/uL+VZ/U13O3/W
yp/z7X3nyn/ZOpf8AQPu+/Lf4Uf2TqX/AED7v/vy3+FfVnlp/cX8qPLT+4v5UfU13D/Wyp/z7X3nyn/Z
Opf9A+7/AO/Lf4Uf2TqX/QPu/wDvy3+FfVnlp/cX8qPLT+4v5UfU13D/AFsqf8+1958p/wBk6l/0D7v/A
L8t/hR/ZOpf9A+7/wC/Lf4V9WeWn9xfyo8tP7i/IR9TXcP9bKn/AD7X3nyn/ZOpf9A+7/78t/hR/ZOpf9A
+7/78t/hX1Z5af3F/Kjy0/uL+VH1Ndw/1sqf8+1958p/2TqX/AED7v/vy3+FH9k6l/wBA+7/78t/hX1Z5af
3F/Kjy0/uL+VH1Ndw/1sqf8+1958p/2TqX/QPu/wDvy3+FH9k6l/0D7v8A78t/hX1Z5af3F/Kjy0/uL+VH
1Ndw/wBbKn/PtfefKf8AZOpf9A+7/wC/Lf4Uf2TqX/QPu/8Avy3+FfVnlp/cX8qPLT+4v5UfU13D/Wyp/
wA+1958p/2TqX/QPu+/Lf4Uf2TqP8A0D7v/vy3+FfVnlp/cX8qPLT+4v5UfU13D/Wyp/z7X3nyn/ZOpf
8AQPu+/Lf4Uf2TqX/AED7v/vy3+FfVnlp/cX8qPLT+4v5UfU13D/Wyp/z7X3nyn/ZOpf9A+7/AO/Df4U
f2TqX/QPu/wDvy3+FfVnlp/cX8qPLT+4v5UfU13D/AFsqf8+1958p/wBk6l/0D7v/AL8t/hR/ZOo/9A+7/
wC/Lf4V9WeWn9xfyo8tP7i/IR9TXcP9bKn/AD7X3nyn/ZOpf9A+7/78t/hR/ZOpf9A+7/78t/hX1Z5af3F
/Kjy0/uL+VH1Ndw/1sqf8+1958p/2TqX/AED7v/vy3+FH9k6l/wBA+7/78t/hX1Z5af3F/Kjy0/uL+VH1N
dw/1sqf8+1958p/2TqX/QPu/wDvy3+FH9k6l/0D7v8A78t/hX1Z5af3F/Kjy0/uL+VH1Ndw/wBbKn/Ptfe
fKf8AZOpf9A+7/wC/Lf4Uf2TqX/QPu/8Avy3+FfVnlp/cX8qPLT+4v5UfU13D/Wyp/wA+1958p/2TqX/
QPu+/Lf4Uf2TqX/QPu+/Lf4V9WeWn9xfyo8tP7i/IR9TXcP9bKn/PtfefKf9k6l/wBA+7/78t/hR/ZOpf8
AQPu+/Lf4V9WeWn9xfyo8tP7i/IR9TXcP9bKn/PtfefKf9k6l/0D7v8A78t/hR/ZOpf9A+7/AO/Lf4V9We
Wn9xfyo8tP7i/IR9TXcP8AWyp/z7X3nyn/AGTqX/QPu/8Avy3+FH9k6l/0D7v/AL8N/hX1Z5af3F/Kjy0/
uL+VH1Ndw/1sqf8APtfefKf9k6l/0D7v/vy3+FH9k6l/0D7v/vy3+FfVnlp/cX8qPLT+4v5UfU13D/Wyp/z
7X3nyn/ZOpf8AQPu+/Lf4Uf2TqX/AED7v/vy3+FfVnlp/cX8qPLT+4v5UfU13D/Wyp/z7X3nyn/ZOpf9
A+7/AO/Lf4Uf2TqX/QPu/wDvy3+FfVnlp/cX8qPLT+4v5UfU13D/AFsqf8+1958p/wBk6l/0D7v/AL8t/h
R/ZOo/9A+7/wC/Lf4V9WeWn9xfyo8tP7i/IR9TXcP9bKn/AD7X3nyn/ZOpf9A+7/78t/hR/ZOpf9A+7/7
8t/hX1Z5af3F/Kjy0/uL+VH1Ndw/1sqf8+1958p/2TqX/AED7v/vy3+FH9k6l/wBA+7/78t/hX1Z5af3F/Kj
y0/uL+VH1Ndw/1sqf8+1958p/2TqX/QPu/wDvy3+FeyfDXwL/AGRAuralF/p0q/u42H+qX/E16P5af3F
/KnVrSw0YO+5w4/iCti6XskuVPcKWiiuk+fCiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKK
KKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigD/9kNCmVuZHN0cmVhbQ0
KZW5kb2JqDQoyNiAwIG9iag0KPDwvVHlwZS9PYmpTdG0vTiAyMDcvRmlyc3QgMTc5OC9GaWx
0ZXlvRmxhdGVEZWVnZGUvTG5uZ3RoIDMwNDAA+Pg0Kc3RyZWVtDQp4nJ1bXYsctxJ9D+Q/6B
+MSq1PCIFwnXAvJibYhvsQ8rC259pL1jtms4bk399z1NJO947ULQ/YWz2zXdJR1amSVNIaq7Qyk
3KiTFCiozJGiUtqwtf43iR8b9Vk1DQJ/qsp4lGUtXh0yumgJq8cFazygk9B+YjGogp8O6koeFtU9BBa
JZkUVFPQyqljdd6shQxolCoRNu2VGJ2UxWeDhm1S+BrfA9/kE6GKRdsO+jYSAwCjTTcp8cBjHST
0HNoJgOPwOUR8RnsRjTu0E1NUeEUSfumsMhp63kCGqDxGLRg+ABuJ+OxgCXTmPQyIMUq8
P+FID1tZ4xUgGUu9olyDMTA04zX0KaEX0F5AewHtB1gvoL0lwwS0F5MomNck4EITkxa8R+Oj/Q
BjCiwaNSRMGmb/g3HHib6AReGWCWDpECsRFobEhwg9B2/ANJPDuPFv8jBOQjtBO4UhTcHTF
XQm3qMEmlT2EoyR4CGt8dlDevoK/hOnkoPE4BJcYiY6jT5Er7CIBQJ8QV5Qk06luWEt6ybyCroO
AESjcU8P01EBZCDIbPD8Bv1Fsg82sBGohU5PmSZkQSI/DNwNh4A44IHwGzgQ7sODxQM5AJO

5SdgOfDzxZYHTrRHSCg+RvBLQgpAlkbxAil6dh2tBQTBE+A1aDhiswMcukm0wFgZBdbIHfhSEh9
eG34A/Gh4hgb2QoggZbzS/ARVACDxM4BajzPAh8huwx8LVglDyDkORiXwL1ELLni8zoDxbBlwfM
vnJPVhUEE8+wkMIB7BwYoOgZ8ovk4e0KqlmcNyCsMCwABXxFayDxJGTHKBIsDlsyFaLUQrsG
MhpjiSQ1lgwEFazLz7kl9FyAExBoIUa1jBthGwoTxZraqHIRKtikFHDNLc6sgQBYMJXPDrAlRBDG
6a/DJIP5EkCLFo6VyMJGZGgX/ogRENdbgHvwLDPTmGOIuBBof5Ysgvo9MoTANKfWRfeEj55Yh4
gPMFUZY0VCXm5ATwilckkVoJTGZ2gmZixCF94lGMQpiBmNRCXFGsmwHk8stoGf5mjsEDvYyA
TIHZDRGXsuMQICnm/MNwltkQfCk7LpHcOr/N4NCIQuF/TctIWNpkhcQn/mRcaEQ8shdj0ArzGIPG
MmPRetoZ5jiGrAMGQwjagwKGDtVBUzfHJMwrKc8DWYN90l+GpAcUZDSaQWi57DKh6UwONto
O+ZNPQGIIzJfZlwcVfhq6AnymBt8j0wxTgJBqhiQFB9AK/QIOoRUGLZzPDMw+grBIRj2R5l9G2Jw
TFTjOxEzyGgYzWzN0fA5lw1hnmE70MGPaMX27PMOI+uGHw2+cBbV6fXhz+O3w9p8vx8Obx4e
v7x9/vjt+Prz8Xek/1OG3jyq/8+OP3383q5ii8vZFS8dwcN29qfjmy839hWp9/fBSyVIFal+vm31IDU7ivS
7T1vBMUyVuDg8Jp9PXJIPDmy5VmvCQeBWYjEJsXQM0W5urlx5eO+oOfanTBMwwYV6/Bq2b0f
quJ+MoWjvMnjT3KbOws+hyadocQUvD6k1ruW/vhYvKDjxrNjvz13TW586wN8KoN6bZG7PNymi4T
O4N132zN8lo5nip0zZqYsY18u34uM7P40u98Tk9iNaYS512tpuYm02TdTo3Rwizux6YxetH+WD9
bOFwtz1TAvXZaRzm2MPTR27PepuMnKbtDDxms5it7PRPG3SqG1nY5VhcE/Y6doPT2n6UqdpG
+6nua1u9qY3LeTnIOC7FPCjhprspU4breOMNDXJs4d2NrHvcsiPpqMpXuq00SZmT9tMR3to53Tk+
0wYTUfWjJJwJkrxQBka6wgdBGFzfrNNRgXZHHXoUimMUsnaS502QleMdg3GmUehy6MwyiMbL
nXaUCPmpWugziQKXRKF0XTi9CiJZIYU25dxsXjUQRBHtyBOLnWaxsKc6Jrci9vGivPWJXYXOHF
0SnX2UqcN1WFKuwbqPBP7so4jq4GXbjUaUNlrekaqPMiMvUJMEpBP0zBmR/F9mVcrDN2EKT
NHbdvZu+0veVOXRKIURJ5c6nTBjghY1+DcSZR6plojZLlu0udNISPWeUaqDOJWOrtYR1NlZ6Os
mimRTF+GVguLncgsKg8iCE1lJoGw+Qbmvxj5XnLYqx5z3hdH68fxBtMQ6mNlwXb6/CGgre76mb5
fRCvayi18bJmfBVembc/PBzo4h3lZBjmZKFLdUMdXj6R6MGQzVVQaCZ1Hj9sD79PKxmIvdQNpS
ZlZBGxWRrcx1loJX1aySit4tRQauPlsddVeE2hlenTSkbTTBze2ReOVDfU4eXjqB4MM7ogj6Gh1DZ
bxHKh3dme2VzB212UixldlSfdUGri5UfK5a7eMtsYrorc5lG93dpaii18VrMZlfhLRVZmTb4MErLNE
5LU2hpShRPJTr6JXSZNtdKqZ3sp+1KkPSL4DIN0yo2lNoged7VPOjZB1p4NfV5ZUd5haVOQ6td
s0Hfc1N1y5kW6hl+9T6hnOR4aOGQpTqizrCfljdNd3o2h2rroZWx3T5xLPd357pSra23SW82OFjAR
0bWh3IPBhtE3QXcplYXHclL250Q4i1V0Orcw7Gc9s2QfcgzzX2fBmha+Vhgo6fhRXeVGfUEeYrEF
3Tba6ipH2+JW77qEFcn15umF7iG1odmDyVb1bs96EWEvk+vYYr6FgENbTakOdj/qsglyK69Kvo5y
Z2lS+Onvbo5UryclVepfgrvk8vP7qyx2KgodUxHa9Ktlnp90xXUrfvL+796OJeFudgZ60OZN7laBN0D
3los0zor+/D8KWMxfGSbBfjJd/KahN0F3LZcoYNYgwTdBonqC8E9SWmS2FZQn/ruV1FI/YhlwS/Y4
A+vclwvRYnXGetDkxewGuWhXehxkKv2KfXcMEdK6OGVgdy5JrzOsiFXv26+7mJ/ZsmwzXiwpXqj
DrCfJWua7rh+wpWGlpt0/FiafusTOKe6Urqjv3Vfhpe7S/Ozc5aHci89Nkm6B7kVGaZ1F/tP5Xj9yGH
hIYHcr5xeh3ksglNG8QYJuj4OVrhTXVGHWG+htk13fbCyrUngBR2DNCIF691jg7cNLQ6MHmHt11
F3oFq5up8vnjahTxML+caWh3lvGTc3sTtQrYFcp9eaTjvuPFKcJnQKM6ol8w3cLum26FX+1Kk3jNA
n16yTRTfvg2712G/ZH9W3bW0H78Sq4uly5GB6dfqzXatHnuFttZ29c70i/U7qtsq+zaahm1UCtCmF
KCN9C8Qb5fepX0WamT7YqUx3R3jjuq2yr6N3LiN5pWKMSXJ9Yvz5ulu0c27u2N7vHMBmZVL2bB
UHFy5/zPTdd6plGP6cs5az7bqYUSt/tZSxa2l1K1r3SHUpVydMWtqq4FXyVUNuDm2X24/fn1oDk7
ah8xmJ5t0ltxV6ebhse2W4o2CvN62Ly0sNN4+Hl+vT6fHw+vT3fHXmy+qsAQth+/zb1Xhds4a1QB
Pv311/Pvx5fGfp8v4v6Ct+9Pj8fCKP36+/3D+8Bavvjv9fXhZfP94+Pfx5sPxYX6mTn3+z/3d7f3xzacbl
uQXP92jhZvH29N9+fzwePu/GzzkT/89Pfz57nT68/Di9P7rZ2DK3/z16Xh8nI3z6837h9Pi878+4efi84
vbm7vTx8UXb+5uPwxX78794LWPDzefi3/LWF99/fzX77ClcWfr1r9sKH9tUP4QYH3Nfkgsb7o/F3P
TpUg914bLHej1jeFnYnU/97mYWykFmFJ/KZWr1ZXK52J5gfG5WF4XfCZKzaHs41fX157E8qJYF

WUzuboX9SSWN5CexPKuz5MoiWPWW106eRKR6x0LWTYL67sMC7m6M3CW67P5haxJq+ivD4f
Pcn0lu5BFf33auJCrU72zXJ+eLWRNmkV/fXpzlusjkoUs+utzglVcFdsXclXRXsialv+upq6kKuS5UI
W/XVd7izXxa+FXFWYFrJOGnWfu6puLOSqhLCQRX+9Tz7L9WZ0IVc7voWsk1aZiNe7jYVcLekXs
ixuyqWNshCtC8K6TquLn7omqRN9nVrqVPOH+v67/wO9qO5eDQplbmRzdHJlYW0NCmVuZG9ia
g0KMjl1IDAgb2JqDQo8PC9Qcm9kdWNlciij+/wBNAGkAYwByAG8AcwBvAGYAdACuACAAVwBv
AHIAZAAgADIAMAAxADMpIC9DcmVhdG9yKP7/AE0AaQBjAHIAbwBzAG8AZgB0AK4AIABXAG8
AcgBkACAAMgAwADEAMykgL0NyZWFOaW9uRGF0ZShEOjIwMTgwMTE3MTcxODUzLTA4JzA
wJykgL01vZERhdGUoRDoyMDE4MDE4NzE3MTg1My0wOCcwMCCpID4+DQplbmRvYmoNCjlyNi
AwIG9iag0KPDwvRmlsdGVyL0ZsYXRIRGVjb2RIL0xlbmd0aCAyMjY+Pg0Kc3RyZWFTDQp4nF2Q
wUrEMBCG73mKOe4elrR7EIRS0BWhh1Wx+gBpMq0BOWnT9NC3dxLLCg4kMPn/b/gz+tl9deQT
6DcOtsceoyfHulSVLcKAkydVV+C8TXtXbjubqLTA/bYknDsag2oa0O8iLok3ODy4MOBR6Vd2yJ4m
OHxeeun7NcZvnJESVKptweEog64mvpGZQRfs1DnRfdpOwww5PralcC59/RvGBodLNBbZ0ISqqa
RaaJ6lWoXk/uk7NYz2y3B2P96J+1zV98W9v2cuf+8WYq7MkqfsoATJETzhhU0xxEzl8wMMmm82
DQplbmRzdHJlYW0NCmVuZG9iag0KMjl3IDAgb2JqDQo8PC9GaWx0ZXIvRmxhdGVEZWNVZGU
vTGvUz3RoIDUwMDM4L0xlbmd0aDEgMTg4NDkyPj4NCnN0cmVhbQ0KeJzsnQlgVMX9x7/z3l7Z
7GY3953dsCSBhEBlgETk2FygxiOQaBPFmnAoeKYCXIXAq2rwwLO1XogVDzw2G6rhsEatrfUoeO
PRmlapR7HgWevB+3/f5ID9VwxUk/z7Zz4v8515M/Ob+b15bydvdpO3EABSKBYsq6o7ePqSY79ZA
O3JFiDtzelV1dMu++qKgyG+GQFoj0+vPaLuDtdtD0EY24B7fje97siKv2xtylF2z1og/eZD6uqnnZK/w
Eb7WraadWh93UHzkjrUAnK+ANzNR9SNKY6dtWQj23KwvKm28tB636PI0yG+3sH9CUdVHdZwL
VIKVAYc4i9bs4pzS1dx694D+Ji2msXzDljkf+rKXe8DrGS/tiPPr7lhFMc760fAXHpuYD11ROaF7YgFI
Fsr4vteU84+ezjl/u23ghx10VA4ar5c085a+3MrW8DVTsgyh3z5zXPffeOyfcA4nKz//nMiFviHsf933J/+
PxTFp31zBUxb7Dv+UBgxUnzTj9VvlnTlba/w/LDTz5tTvMtm9ZcBfEy/UnbeErzWS2JlrhK2r/Hcv+pz
afMe7jt6z9C7GCbrtaW0xYuMvLxKv3rNMtbTp/X8tEhW5Mgrqzn8b4A81xYy2ae94/p5xznmfSZI9U
cJmDV25MyzfiF65amf/nl19944RjOulGyvglij+5Sdh6PSiy+//PlcL/pKerCcZOa4j0UJdLiY1uDFGBxJu
0myXwFdf0nbCCsc1I9aS9hAenesP4/jtTiHVYu2WzQTSxfyjU6cxaOkB6T+sEo/uPITrS/unCFK7FN
EOMhLxDDYb651vXmksNh6XNIO6An3okN/Bi2WhYhDP7D+4l1p45v+6vfVvRd/+K5y6+9xQ098n
Rylt/Fz2wGY/W11bffi6j67o3A96zWa9kwfsrf+7A6P++Ce+Ej2W8+4vMeHeoZreMzX9pRfZqbtmVjC
/KsZzJlsN8v0TFzC8ilep495VzAd3WPjYoj5T/xS7Bs8N6cPYd/XDIXfCoVCoVAo/v8bjbDWD7UPe4vl
b/89vioUCsVQlmCsdzB4YWwYal8UCoVCoVAoFAqFQqFQKBQ/FO5j7UKlz+223gy7/Onl+p54Olb
B7ZbJM4Bhw7KtEa3kwJYdj+xxY+AaUzYuOxuBMbZx92LkvwbMb/t3F+cMWMeK/3JE/1V6GTlwX
ux3cJYZahcUCoVCoVAoBg0dujCx6rrQeB+UYt0W3YkvHAYccBg7EYUoqhNOajSijW/ggovqhpsal
9WDGONreOGhxkqNQyw1HnHUBMQbXyFRahlSqclloqYg2fgSqUihpklNR6rxL2QgjZqJdGoWMq
g+ZFL9yKJmw0cdBr/xBQLUf5rrP2oOAtRcDKfmSR2BHOpl5FLzkWd8jgKMMD7DKKmFyKeORgF
1DEZRI1BHSu1GGOMT1GCluo4jKWOp36CCSimlqKEWoZx1AMwnjqR+jEORCI1Esqok3EAdQr
1l0zFRGoQk6jlmGzsQAWmUCulVmEqtrPB6jSUU6dLPQivXnYcjCrijHzgE1dQaTKMeKvUwTKcej
oOpR+AQai1qqDOo2zAThxofog6HUetxOPVlqUehlvojzKA2YCZrNqKOerTUY1BPnYWjjL/jWPyl+
mOpx6GB2oRG4wM042jqbBxDnSN1Lo6lzsOPqcfjOON9nCB1PpqM97AAzdQTMyd6EuZST5Z6
CuZRT8Xx1NNwgvEuWjCf+hMsoJ6OE42/YSFOoi7CydTFUs/AKdQzcaqxFWehhXo2fkl9R+pPcTr
1XCw03sF5WERdlnUpzjDexjKcST0fZ1EvwNnUC6VehHOoF+Onxl/xM5xHvYT6F1yKJdTLsJTaim
XU5TifernUK3Ah9UPcZHTkKlXMXYGfUa+Weg0uMd7CtbiUeh1aqddT/4wbsJz6c1zOnF/gCuqNuJ
L6S6k3YQX1ZlxNvQXXGH/CrVJvw7XUIbiOejtuoK7Cz9nOHVJ/hV8w507cSF2NX1Lvor6Ju3Gz8Q
buwS1M34tbqWtwG/U+6hu4HyupD+B26oO4gxrCr6htUsO403gd7VhNXYu7jNfwa6kP4R7qw7iX2

oE11HW4j7qeugUbcD91lx6gPoKQ8Sp+lVRtFE7EaY+hnbq41hLfYL6Cn6Lh6hP4mHq79BB/b3U
p7DOeBl/wHrq09hAfQaPGC/hWanP4TfUP+JR6iZ0UjfiMerzeNx4ES/gCeqL+K3xAI7Ck9SXpblH6q
v4PXUL/kB9DU9TX6c+jzfwDPVNPEv9E54zNuPPUt/CJmoXNIP/guepf8ULxia8LfUdvEjdipeof8Mr1
HelvodXjT/ifWyhfoDXjOfwd7xO3YY3qB/iTeo/8CfqdvYzugNvUT+iPouP0UX9BH8xnsGneJv6mdTP
8Q71n9hK/QJ/o/4L7xpP40u8R/0K71O/xgfUb/B36k7qH2BgG1XN6d82p38q5/RP5Zz+6b/N6Z/IOf2
Tf5vTP5Zz+sdYTv9YzUkfyTn9lzmnyTn9l/knP7Rv83pO+Scvl3O6dvlnL5dzunb5Zy+Xc7p2+Wcvl3
O6dvlnP6hmtP/ozn97e89p/9Fzul/kXN6l5zTu+Sc3iXn9LfkNp6WmtP/gzl943/xnP6cmtMHdE7/XM7
pn8s5/XM5p38u5/TP5Zz+uZrT/9/N6W+rOV3N6WpOBzjjwt0UneSArutW2653asyf3fYkdm627qdw
OLnnsEd++myDjTl2uw67Hm1n0mq32Jlpw0Chf69ixf6LtvDv+/kTC8U+IKKThtoFhUKhUCgUisHAIR
Jlrq9sfbeS376+cnDbtb5yRDkcEa2Ya6tvWV/Z1fpK8X8Ntb4aEjRXyIC7oFAoFAqFQjEYUoCsFgs
u9ZXFvmz254kipvDKZPR3HNGRUW04oCdKy6HwwK7xeVg0mqzOOxwDNwNquV7FSv2X/Zh6e
3ov4piL9HcaUPtgkKhUCgUCsVg4MmKNtdX9r5byW9fX0Vzi3LJpBtwuqKdEa1wbcUVV1SUBQ6L
K4pJm8NKVesxf859mF9FdV/FcVeonmyhtoFhUKhUCgUisHA63eZ6ytH361k5Pqq9w/8XNycfeura
JcrOqKVKPMjLTidFkRZYpxOJ2xRVqe54Bowv9X6SvGfsQ/rK2f/VRR7ieb1D7ULCoVCoVAoFINB
7DA3rFbbrvWVvf70ErG+6v52B3N95e5vfWWPsg3s+sr63cVqfaXYA2p9NSRoscOG2gWFQqFQK
BSKwSA+N4brK2tU33opcn3V+wd+MdxchpmM5RLL0/NNer1EI5orLpfLimhrrMvlgIPaxozoyFXYD0
k/66t+ihX7L/uw9HYNNbf7HVp87IC7oFAoFAqFQjEYJOZ7YbPZnX23krZdH1ph1/rKw83tlUmur2K8
npilVlxwv1wu61wWePdXHw5XHaqa+DWV/08mVctrX7YB/WV+7+qyj2Ej0xf6hdUCgUCoVCoR
gMUorizGerR/fdStojHkvdm4w1v8kwXiYTAW9crDeiFbf5kRY8Hjvc9kSPx4Mot4MZ7oG7Qe3nyRnq
ydqKPbAPS2/PwHmx36GnFA21CwqFQqFQKBSDQqfEBDgcDnffraQj4rHUvf+BksAtNIEmqXGJC
XERrXjgifUgNtYBjyMtNjYW0Z4oZngG7ga1n0dnqydrK/bAPqyvYgfOi/0OPX3CULugUCgUCoVCM
RhkTkwy11cx3t6MPa2vEhGXLJOpQHxSYnxEK15447yli3PA60iLi4tDtNcZFwuvFwOFWI8p/jP24T
uv4/qvothLLJkTh9oFhUKhUCgUisHAH0xBVJTT03crGRXxtT+966skJCMhVSbTudpKSUqMaCUO
cfFxiI93ls6ZER8fD3dcdHws4gbuA4B+nkyonvym2AP7sL5KGDgv9jss/uBQu6BQKBQKhUIxGGRX
pHJ9Fe3dw/qq9wEVyeb6Kk0mM4HE1JSkiFa4vkqIQ0JCFOKiMhMSEsz1VUI8F10D5nc/6yv1zbC
KPbAP/5qn1lc/HJbsiqF2QaFQKBQKhWlwyDkka05ndFzf51HOiA9/eh9Qkc4tOVMmfUBqZnpqRC
uJSExORHKyE4nO7OTkZMQkupiROHA3qP18QKU+v1LsgX1YXyUPnBf7HdacQ4baBYVCoVAo
FlrBIL/eD5crJqHvVtIV8bU/vY9hz+KWli2TASAJoySjopVkJKclly3NhWRXTlpaGrzJMcxIHrgb1H6+m
kh9c5FiD+zDv+aIDZwX+x3W/PqhdkGhUCgUCoViMBg9KwC3Oyap71bSHfG1P96eOJtbxnCZzAF
8w7N9Ea2kIS0jDrkZLqS5RmRkZCAu1ZORirTIT7l+SPp58rv65iLFHtiHPx3N6L+KYi+xjZ411C4oF
AqFQqFQDAbFc3MRE+NJ6buVjNn1oRV2PaF6ODdfnkyO5Gorb3h2RCsZyPBlwJcVg4yYUVk+H
+IzYrPSkZE+YH7HfK9ixf7LPvzpqK//Koq9xFY8d6hdUCgUCoVCoRgMJpw8Eh5PbFpWb4Yn4mtV
e9dXedyy82WykKut/LzhEa1kISs7C9nZHmR5xmRnZyMxKy47E1mZA+Z3P9+spb4ZVrEHovuv0kt
2/1UUe4I9wsID7YJCoVAoFArFYDBx0Sh4vXEZ/t4M764/CsSuJ6jlcwsUyuRYrrYK8/MiWvHDH/AjE
PDC7x0XCASQ7I8P+ODPwkDh/V7Fiv2XffjXvOH9V1HsJfaJi4baBYVCoVAoFlrBoPJnxYIPT/Tn9G
bEy59eeh9QUcRt5DiZPAAoHFdUGNFkDnJG5mDkyHjKxE8aOXIk0nKSmJEzbMD87ufJ7wP3YHj
Ffzn78K95lwfOi/0OR+XPhToFhUKhUCgUisGg5oYyJCamDM/vzUiUP72k9MQTul2eKJNTgeKJE4
ojWslH/uh8jB6diPzEytGjRyMzP3X0SOTnDpjfid+rWLH/sg9/Ojp64LzY74iquWGoXVAoFAqFQqEY
DOpWT0Fyclpe361kcsTX/vQ+9mlSt5JymZwOIJZPKo1oZTRGI4xGSUkyRicfUIJSAv/ojJlxGB35Kd
cPST9PflffXKTYA969r1oyYE7sfzjrVg+1CwqFQqFQKBSDwayOaqSIZY7qu5VMi/jan94nqFVwKztl
Jg8DJh9UMTmilRKUIJWgrCwNJWkzy8rKMLzEV1aMkrED5nc/X02kvrllsQfi9r5q2cB5sd/hmtUx1C
4oFAqFQqFQDAZzn6pBRoZ/TN+tZEBE1/70/gPVQdwmHy6TdUDI4QdVRrRShrLJZZg8OQNIGY2

TJ09GXln25AkoGz9gfvfz1UTqm4sUeyCh/yq9TO6/imlvcc99aqhdUCgUCoVCoRgk9J6QASH3n+U
eU+IFWBdi/gj4mXlggFEYg4moQBWm4zAutJoxDwvQgjNwNlaKsbagP9UwYD5OcFfNaaxZy5pzc
AJOWuk4a1dN4+3v3OYYv/o61HVLj1d7hbChr7rQNED73xV4qBarrXfXE/FxRu/T5HOQh5Hd/5BW
BJSMGz8hog0e0/SDcEjNoTj8iNoZM+tw5FE/amjEMT/496fqu5IP9VP15d3S/3VnLVheH5w6ZfKkA
yceUFY6flxJ8diiMaMLRxXkxyRI5szPDAs2+/LysxIT0tNSU5KTIiPi/V6YtyuaGeUw26zWnRNYFR1
YFqTP5TbFLLkBg46qNDcDzQzo3m3jKaQn1nTluuE/E2ymj+yZpA1j/9fNYPdNYN9NYXXPwmTCk
f5qwP+0HNVAX+HOHpGA9NXVAUa/aEPZfowmV4h026ms7Np4K9OmV/ID4kmf3Vo2hnzW6ubqt
hcW7SzMIA5z1k4Cm3OaCajmQolB1raRPIUIRNacvXENg0ON50KpQWqqkOpgSrTg5CeU908N1
Q7o6G6Kj07u7FwVEhUzgnMDiFQEflUyCqolN2EbJUhu+zGv8A8Giz3t43qbL28w4vZTQWuuYG5
zbMaQnpzo9IHbAH7rQoln/NOyq5dNh5X2XDJ7qXpemt1ygK/udvaeok/1DmjYffSbFMBg9kGbbWc
aU2t09j15RzEmjo/e9MubmwliYvZpd88EvOouo9vXqDazGk60R+KCIQE5ree2MRTk9Yawsyzs8Np
acF1Rhfsqv2t9Q2B7NDU9EBjc1VGWwJaZ57dnsoLNrKkcFSbN7Z7YNtiPD0Jl3v3xLy+MpmS1c1
Uzcy+kRWmR4GDeUGE/HP89KQhwGMqM2VeGVrnLEaaRS0Cs3IGVkJiqpsavVONPNN+5A1x
xvwt34GXgGBD7dF5jT35NhyvJ/BTJrXSd+lxvLedKigIJSfb14i9kqeU/o4Re6PLxx1Roc2ldi9TPi8K
GWY9vcOHEMhz872zzByzuCmM2d0LIZDd37fsxODyM4pqAxpDWZJZ29JIHmiXLekv6zJsCvJL
Xyhd6YsiR2/fj8SbFV8+fGBJJ31E8r7u8pi5QM+PoBn91a1PP2NbUR+x1l5f1lfWkQvGVDXq61pPS
0nVZyotyVI9lc6fBFbLk8McmL+q5lZ0XpcwQ/mkxb9NB3drozM7eo02H3bGbUYexw7SS0S6zHi9D
Ewsi9w+M2l/wztWq019LrlZTf3RrqzOibBonoNbWaQH/tNam1uYOY9nsgN8baF2n3aXd1dpS3dR7
QjuM9cvTQ9Mub+RBzBcTebFqqGgLiEtntAXFpXVHN6zzcla/tL4hrAmtsqmisW04yxrW+TnlylytL9fc
85t7qBG80MOaQxalrwsCy2SpRWbl/TkdAjLP0ZsnMKdD687zyjxSiLb6ZeVu/T48yKDBS/UzrGTQ
EdTva7e7i4MdjOMSZBxOKiheZ3QyMbFE5hdeV7xso74Gx6GE2WvCR5rZa9qDVcUyLjmwOx4zV
sZhR3exPaHYV55GsZEMGjw9qSMYrmK4jeFRBhsdWoO3GAwGXb9bXxWe5mMLd7lhT3mCfieP
KkjdxGAw6PT+Th7Lndjek2OhV3e0R7nM7u+QVun6HbTyUL0MyxgeZNjEYMPp1NsYDAadqVUs
WwVNX6XfHvb6vOVO/VYsZdD0X8ljBHxs/RftXjk2N7Z74ouD5V79etQyaAjph6GTQWOzV9Psami
sXhMuHCuHsKbdGVPsZf3ldHo5HVnOLldShdwPMpj1l7fHJ5nNXxj2xEq7n4aLxnUn2r0pxbUchbM
g9Hn6qbyN8OILGGcxnsM4k/FsfS7c0s9gu8dbvlz9TWX1qXoiRrK4XE9CMeMqPQ3pstricEx3P4vD
l/KLecSVeoqs4tHdGMfYodvDxT7/Bj0oB//S9qho079Lw97E4kf0i3U7ElhrGWsl+zyP6E6eWac8kvr2
KHfxinKXXs/DrOew+Oij4CifKhs6NcyGymP1aj0DSSw7Sc9EluNpepaM79Jv5y2PT7+IPTfD17IBv1
ZaXWM2yu6ndF9aU9rdMcWd5VH6FJaG9Ct5Aq6Una9ozy0rRnmuPgJFDBrHeCITS+VF38pUK89
aK89UK89UK51q5dUH/TKWXYMY6Y/Rz0KKfiRUMtzFtXlaJYQ7oOpkYPqJ4nZ6qp3BgvBs4lIK5ae
1RMaZnKeG4eFktpd0VUzz1EX0hr/OFbDOoL2pPTik+bYOeLw9lVHtKumnQEubl+oie3H1qaJhknP
JH9AwOhDkwmXpWONEXKvdx37yQfbyNflrbBA6S9qL2snm6tU3cN+NneuLneul/dsdGp7a5+0Wh
vWDGXeUZ2lY2dpz2J9zGIKZt0J7gjbVPe13rML3QXtPWYSrjLdyfy3gd4xLG68PZT/k6tl52RvT9pr
A7yTxY7YlwwZiehC+nJ5Gc3pOISyouz9Ee1x7jesanvcp4OOPHtE4MY/wo4xTGndoiPMX419p4H
Mh4bU/8W22jeYlrD2sPoYxxezjGdCEUtpvRg2GbGT0QRvde7RjfrU0BbQ3SWPX+cG4ac+9uzx3u
82xge0K7U1sUzvTFITu120WD+JSVvmKLGSNOWxUuNRtZE7o963TVmrgimlwZxgYXC1XpRT
VFioWvfn+Av9pf7V/nKvdiUnkNs0vn615dRS+DVePQxBhhXaZWFLaaj8Gx6TeVwallFXylQTtUWm
QPX2le6QqanaxTiCQWMbSxiWMixJOJ8rhxXaOQw/ZTiX4TyZs4hhMcOZnE1aaNFCixZatEiLFq0
0KKFFi3SokX2vpjBtGiiRRMtmjRJC2aaNFEiyZaNEKL098mWjRji1pa1NKilha10qKWFrW0qKVFr
bSopUUtLWqlRZAWQVoEaRGUFkFaBGkRpEVQWgRpEaRFUFoU0aKlFW0KJIWRbQookURL
YqkRREtimhRJC38tPDTwk8Lv7Tw08JPCz8t/NLCTws/LfzSwksLLy28tPBKCy8tvLTw0sIrlbzy/Cx
mMC26aNFFiy5adEmLLlp00aKLFI3SoosWXbTo0s5s0zeXP0mTzTTZTJPN0mQzTTbTZDNNNku

TzTTZTJPNPYe+SA6GxstmCcNShmUMpm0nbTtp20nbTmnbsKS+vxQymbYgWIVqEaBGSFiFahG
gRokVIWoRoEaJFSFqspMVKWqykxUppsZIWK2mxkhYrpcVKeeEuZjAt9v2i3OdTo50vGhz8XastE
yNlvBTbZLwEW2R8HtpkfC5Wy/inuEDG56BUxmciV8ZsT8aL4HOIsK/UU57EKeAlhuMYTmO4jeF
BhkcZ7DK1ieEtBkMbHxxm8diPsN9mf9D+qN36oL3LrnlS9husz1oe9RmfdDWZdP85emaW86jnF
pwlDSI1O0M/CVCnSpTU7Vx7Hcc59nx3MZp44KxH/q354tN+eLRfPFgvrqgX5RHadOFRc50fpRy/e
wTDUFX7hTfFobS3LwpmJmufGhbsi+cO8HXITZ2RyODBYy3MbQxrGa4gKGUoZihkCGHwSfz8lm/
ITisp8mNDHkM2Qx+swwskJXFdEBfrCK7T3GJ1+5NuRjN95I2g3YZwXhGjjnDeEYweDufN9pVHiYe
QZ94ViV/zzK1h/GDY9w6L7++O7gv7NjC6O+wbx+jYcN5oRseE857zlbvFkfBZTNP6nriOx23GM8O
+o1htRtg3klFBOC/XrJ3PjnJYOI04B3GOT1Ww7t7CoR9BzlaFvYdYNZ2IM888cKGQumelcGM9X
Y6tH2daLCIYLTvQ9+1vm00/zsHlplfHa/4OC6NNOR3iqKDTt7HwVIYu94XLnWZ9/n5o64IDZvvr3+q
cy3w3sS2R85DvRt9o35WFHQ5mX0G/L5NdhH0XcK23Jhvjw+Yr8i0qfMe30Helr9k303dsDvPDvIm
+jaabaBQN2pqHfLVs8GAeRU7YNz2nQ7o4zXe2L+jL8x3g32iOL8q62y0t3GiOAlq7ex/F8c3P6TCv
8SNLO0RsMN++w77Cfoy9wn6gPWAfZs+yZ9oTHHEOryPG4XI4HQ6HzWFxaA44EjqMrmCB+R5
cgs1rRjaLqRaZ9mqmmm/XccGoCYeGQxCK12u0mroKURPqnIOa2f7Q53WBDuHkUsoaqBChuBr
U1FeEygppQouzGzFBpQU3IXntMQ5sQVzYyN6RdyoVKfUOHMMysi9PN9yzaBC6+In0dhEi9+lrGR
qQknTE1ZWrcIngDpIV9izT1aMEuUnZPZoZuqKlrcN2b2RgqNhNGZmNN6HzzHY11mkdzV1et02L
MqLFhnaVF81TPNPMtLVWNRPaOrMarOYbVkGdGrOaogN+sxvmkwqzGc9RdL5fmrJdtRqzndCN
X1st1umU9izDrtW3xV1e1+f2yTg6wRdbZkoPd6vCKoW1VW26urBXwiwazlmgl+KVjl2VDPh+rFPp
kFch7OtmQT8jOQmN2VcnpqTK+r8p42ZcudtXxddJGNFbJ2EE6xR8T+ZVFIj2sYuXPGG+SdQU
qJ7H0BRafsb8INCy2X5/25LFPe8e5TbNnjPjJvnhRYH5IWFfgSq/G1jn/iW4ifM4rGBqjY8UV3f0PZE
cF5VeGxwbHWguaqxfeqkhvKlvi7r66th0rc0NslsrMHsa2r5txSXm8VTzb7Kzb7Kzb6mBqfKvqoXmN
d9bUObAxWNlbO643Yt2slruCk9u7Eiysyxbyg1x2Ynblkfb0F4m5EFzSGXIGKkJvBLCosLw3i/g6
M4tizHcCe4pSlhyYnb5e3N1T5GV2bKACvUMLs1JNaPyMmlB23dEN5qUSCjZ/+zlbaCKLU1C9olo
/3F8ka7fda2Lht7Lo21i8ePFCUxYXLARqQvl1NaEJM+iJ3c6umqoamTe6N0/XZV5bVFR1h9HJwgl
6IRaZ3ZmpAlHAEQw6ueqyayttK+2auVRY1J6WWXzal/wNvpSB6zjtzPAYuXzWzmwflmOuXxa1jx
nfHXO5asbhtOxi9tBeSiMzzumOg7GFTKzIWVG4onRlzsrlaU25j60mpm+1eav0vCY1ToWFSzsH
QgmFzVysOmW2d/t4YxM2fFKM1FQ0FiwUMjx+vfBFR2D3jewC3taXSibX9R7QrrzF/Y0wjPR3fviXr
PFPuaycLE06m6ke69PdsE9wLoeGTLchQxLrvl3bMY7vWHnAuMds8yMtQ84k2d2hx7CuA+vihHC
j3bxJZLxhUgVY3Ewr85/8tbtQXyD67m8r8cNlo5rtiQciYOFhXUKcLm4yTjDeB+TcQ1WGQ+LC4x7
WX4Vfocv6MGf+ZuyFlez/pGYh/f1rWg0fgkHLkE013QzRRKa8Qq3z+jDtbgOvxHnGI+w1wRcwPY
moRzlxmPG18jH5ZYV1i1Rv8bV2CBsxhxjAe+QhqFVKzBeMd5CLhpxB+6jTwWi03IQsnESLsYvR
Kr+O6aux6+wU7i0Y/VK66Ps6WachVNxJlpxL54WcaLWusW6w/ip8S6vwniMoE8L8L4YLw7T7rS4j
CnG6zgG6/AUj9fcOi3HWO6yHrNzqnGL8ThX3w8Lp9goHrMWW6/85nzjduMBuOjPWI7I4exnNi7E
Y/gDPsLH2IJkQ5CHXt+UmQKv8jliL+ipWpLtCX6ixjNoz2W3i7GbQjxjKzHBjzCsXkDXdgqEkS6OE
TMFleLjzWXNlfbpN+kr9VfsgjLPRzvAHI4RotwJx7Cs3gOm4SV7ReJWnGiOE38XNwiurSQtk37p8V
hudDyleUba+7Orp1fGYcbn3HNnYZDcQ6WcmzvQDvW4o94GR/jE3wuvKJMzBe3i5DoEtu0KG2Y
doTWot3A1fP9+uH61fpjlvGWCstJlucsr1t/Zl1ub7bv/Hr1zmt33r/zeenH43leOzFsPxfTOKLn86q4E4/i
Rbb+Gv6Ev5rXD9s/UBwtfsxeFopLxXXifvGkeF58wKOE3IZpB2pV7PU07XSO0wXatdp17H2T+U6
H9rr2J+3v2me6VR+mT9B/ot+uh/QOfbP+N4vXkmsZbRlrOcJytMXgmSm2TrfWWe+2rrE+bt1hm2S
ba2uxvWe/wH6R49lv8r/5807snL8ztLOd166DV9I5HllbsYrX/Vqeg6c5on+kx134IGchTWSLPPp9gJ
gmasRh4kdilpgnLhCXiGvEL8RNYpV4gEfAY9Ds9L1AK9ftGZtnnaRdol2hbaW23rtD9or2hbtQ3qer
Af0An2sfrB+tH6MfiqPYZG+RL+Il3u1fq++SX9Rf1d/T/+QZY3ZkmVZbDnHcqPILstay/PWQ62ncFtlfd

TaaX3e+rX1a5tmS7NI2MbYTrTdbfur3WafYK+1X2Z/ yf6Jo0VkiHx67t/9kz0tla/BLO1eLcGyVHzljEy
uOjw88gKehzq+Kj7BVH0nz0uMWU7fErVUi/xuAlvQEjLfsxAbMF48iaU2TeeNoaULYfGm1mV5Qpu
MI0WTSLSXcpZ9qfVrLxhrORiu0jdoGUYG12iTtKO1mHWlrfytu5fV+Fq4TJ4mFWCM+FBPFeaJULM
VLWpJeJy7CJGOVZhFR4mCxQA51vm4sff/YmlOABv4v2dt1rcInM5P3XgBp7R+/CWuAdfCquxj
bObztmombPM5bzeL4Y56x3L19ISvh5TOYOcbNuEtean1fZS2xTLOdiBf+F963peURWcSd/ducByq
+Vto9Qo5CuMrzLczdfdfEznK2Yrr5JHuG/uzelr3cm5pJiv6locjbk4j7Pe1UbluNm40DjbOA3P0PZLM
Up8KVbyFdFBi0I4ittVeE0s5+tw+t5/Srs7O+eiEx+IFJEjivl6+NB6hnWF9V7rWutvrM/ZxnK0L8JNvKL
/yqvZySOYg+fxAf4pHDw3qRiFcfS3jL434GStUX8EISINLXzNjuA8XtFzJAvZygUcvZv5en6Er40dnC
dm4TfYljSRzCOaw/4dbKeG43wca6/mGbxQtDNnLmftfPydxx0jyrggH4UgW7qBs1YnfXoTf+NoG9K
vUZwXqsRRbOuf+BHmsocJqBvTPAMP4QDO rFX6sxzv4cKLCjFM/Ip2TXyFvxwPI18CGFV1/nvOuf
s2c2e9s2SSWZKZJBOSQCYbROcqMQIRArJIsCMRZXODUNm0SFzKolbQKiClJe1fi6ItCIJJlq2d
WlfX61b1WqlFgWrqbSlkVbm5n3nTiL23/d/rwn3nHvv3Ln3nG/5fb/vO5egCGrm/ogJqrKmDTeRpcw
RiDHDcL4XolcYnYe7YRROmEce+XAHqrcuhTG8hhI2P37VHsWDZOHwRma1dR36JXoUdGKyq4
RW6i50sYyDX7AeAV34JMEWL/SRrOIBHGsxSBZYC6OgyHMWYZ7BSSQBaAZQIK0PteRbpumn
W6bmW1AW9vWz0lytjblirJolMtCZ6PM0bMmh75AUfYofTOiD+z4JMRrDkno5gFShxQyzkzLnBksy
Ti5Eo5w88QmniFI4uUtClacRoiRkryYFNgkZpKEHyD3A6G631QJDQNbMIODstKHxYOxjx6DLGn
a6VzL1OPH9cHC7zT9ooWtH+VgeNmWqXr+o1x6bC1ua21rxQyMkqENBh+snfQueMhN5CSeaz2
SD1gbcNA6AaNdzzjBLLRHq6BrzcXGbqMyxA0pLM/xykJuobKKW6XwiGMwr8iivMGMGeW0KDJIj
OpyjZyVGbkP32zKTLTEjloM7iPbDqoPT6TSyw3mc3kYmT7oMpqxy93cTDcY34puD1Mf8zF1dvtw
PR5XfZo2zBPYdeaM9VmhpDj0D59gL+deoyds+Epzo8gK7knyJMdcea6DD6gG9vo0P/a6NT/xFKs
G8QSIEPZGpBDxIDGMvYwYJp4S1eB0l+bndlfm552KavDOlinE6awY5nRZCvFOQQzzTikUmhw
WveGwqPn9kw3Vaxiq0+FQFFkWBH4y3MNVUIJUXLJcH9llzideny8QQHgy8bjdxcWRCEOI6DeM
UCgsa6oqicjr8ei683xN3WN84t+jmYFQRjNLk5mshrdouzWiTYvxHEfw+WFpT+gTcU9t2Ax3hZnwt
OgPvmHL73j+OFhfi94C+yvS6dP2IRxRa4Q2a++7m2vsS+hvfmRvaPQEPTW6u5GrTq/Tf7axOkA7
53/7AbXkPln6Othinjqmj6+BGwxJuFJMAkMp3ZuerLIFl50HOt4b+rJ6Xc+1fl361jH+1N/3/EB3jHh9
+Px9e/i1Ht4g3UT3d6z3n63sMdtt7GKYTRSqsF4xp9MoekkSFI4U+XGyG+V24CYS8AieFUicqQ
VFUC54VVBevCqTB/XJTj+cHUXbq4Ok8djUjFzUhT8zn5Xkh1dDQmLgbBytXzmucPYlswsGXb/rW
8uiNRQtmgxsO54dPkAlpQxqMiMI/IAwXkLomy3g9fgTEuKYT1CQ/fZ19nOmgjMNTR20XWIEVm
NrBVwHRn3ta9a9Qe7Tf3qpTb4Mt/4A4iJFk2ozzDRhnm9iZWkfGACfxFGuFpx8n/irx2wdUthoGULZ
wewgDJu6l2wvU/cDiquR/uzfbGeE+20DnFoA3uix3xp6y8yursRLHGsQp2KHWfFaK+SS+vCpW5ne
X+Dp8pNa3z0d8Pm8iXub2iFFvGUyknFrO9wCdaC9P7VOxSpMpScmofeRuM1ZbbVZPr+6qXI7d
U721urdajFbXVpNqbzyKop5aD/H0kbsOjhk7syBziniT9Vz3ULobBJ8btPGPbq7mmlw3zbCQb7jnQ
KTZBw85EKJdzxOeZkhoOuEi6uwINjCq9v0KJJROmIDK0aZOnEM5T2xcMQH9CX6/rUWei2FX3bj
Ghob6TDKVTABojRwkE9vllB8/tnHesvkbua+v2qK9aGl4fLnf1R5yWxtU6p+sxe7e9MXzjTX/olbiHzt
wfmLH0+nnll/9ZFuTSTsC9aPOOmyi1tnS1y+31ojqblpF36tkmrwfpDzISBnHYxtvVIXzpXLFxsL2YU
qV2k0G5P8nf4lfq7ZaAhvDD/lbVO4EhcVrsdd5tTFYgqglURyQpUsp6eGI7GamMk5nKDLPVaneh
UltH/qyy/FOTYWpTrxIQYht/v9nmB98FvoiCK8wmdPcijfhJ5quvWvq4xjYum3r7gofxruPy9bzROmt/
Sct3M8w9xA0XJ560T//vQ7b1XtVeWsM+frXe45/x8797Di9wOWvTbDuzuDpiphFaYWZEDXC8To
mKt+Kz4vsjWiFtFloqIYcslJhlShSzAdZzKQMxk4SiSq1CFFaKYuqQENlgTvJX5zRNz4F9T4UD22f
yFlpy3dAznN4C86tzXwx9vODOYnkKvzu7iBM9bDZ/L30rEtGD7BHQHvjIKltzUVtxfPEVaJq9Rvin
eo3zTuCEu8wYfdhjtC7ioPIIfKi8VJyuXsLGmecg17M3tT4MbQYcdh/UXtBf23+gndwRTxUUSrBiWh
5hK4O0wJ+4vG8JLbdLgz7vYOD/aYvkDG04fLzUr/GCfwWBwNzofTKfccUhKNMjDleG2cxlOpXhk7

5RK5loY3cvfB2C27C7PuphiRG6KT108Pdrso9qJs/nQ6dzydHQTxaMl3p1tsw0c5nMvh+piLZxPxUI
Cnu7GhLsoaHBh1nPfp7rpxDY31TJbckrN2H/rI2vv40f5vvYpduK7KeqfksZ7nPzz5TO7piST8eb5v3
ubn8OLXPsrXz5/84S8ar1s39FfrC+uLyZkBmCelkg+ChIUSMBWFSYpJhWEZzIA/mLlR+lwcHT8hl
/UNHzs40psPFVXDWWH4SZT/KH0qA7blsocUshpUlIdIFRuVauTFZAm7ULpGXk3WsA9Je+VD0o
A8JP1T9u9mt0q75Rekl+XfkrfYN6W35RPkJPuh9CdZWY2tkW8nd7O3S3fLW4kwV1llrmEXS0vkV
WQtK7SSdrZVapcvEy+T5spCQK5xZMh4NiNNkLMOGSEqy0uS7CMh1pCEJ3gycdZcs4SwjCxxqi
CM4x3qOABJnSHidFHLKLSxZ+IQtlxoOIlZhTZwapep0x1FhOSGxUSQITWEgJkFxrJNhapLDtcM6
q8P0hPhvuEJ5hh4SpQVJWkcw3oZhiUQj8YxBHYJ3lZRWUJUWZYkQSxxYEcf1g7SdyQHSBPiA
Akuz2U4anrGzFkZbpxgCutFLB5ZD1o4okQVlSRJtMNLNSEC5EJF6FxJRSe4Tba2JUQJk53D6bT
esuf9ZZQU9357tbQgE9n07DCf14NyoE+mwLjJYGpo3r7CAOXbrTRlJPTIBjcfjYE0q0iUKs/dO9ol
Ds6s6BH2JMSSskwK578dPA0gX8jDVovWf90fo9N3A2wJz8Zxt72xe30A08/QFAjTNgU07gXavNM
p7r9/YHmls5vJh7kyNuV5nmCKCwTIHDIUT/vyGivyRSG+mKLI/0RLil7vwqgBT9Kyh+iYkj8HEOF2H
QUWM0QCQSQQL+MhIfHsC/w45Lb9m7YPu0a15+7gf7Vk28YlJ9Lzfgj723b2PfUpvcv/1v2eauresE
F05doMjy4c/h6vhiQxgcZo9NctyGyMbYT7fTu8u8y+DX6OmN1dIO8wbFJ3+TdHBb5iFQWCnsj3liw
7FrjJiTeiHCnsERYI6wNrS1eG71T2OzaHNoQfVDYqWxzPSoc9r/gf9PvagzPdS0Vlso3obUCz+BLI
AG5DrGl/ngqVeoXEMOTZBGATqqPXHlo2REflxEqMacrQ/rwTNPJvC5JyWRJMEAXa91Vi94g03VS
a0dpKs7KrcnlIT2VvJR+tPFVJKktSvSp2qiVqrcrQ8H6w4paffglRo7HmeB4QCWVPD6b1vAX8A1Nz
RwUOjroBnWDLIfkNAYSa4hNR5NJRDFoIRWW2rG2g8kM4SqYa/dzY63uun2g6ntq6z/qxdSvugbS
iDd9SX24NNDcfO3ToD3943Gyel5t538C06t94E8LNWXwPXolX4y1Wt/XgT7beYE78yc3WF2fz1v
Cwb0Ls0XE0EIPG8wxYmg/F0BnztmbnZOdlwjXKNepeaY+jN3HY8ZYk8ylvG6JfbnC0OdqAquuSy
+vwOr16g6PBebFzpWOt/pqsrJHWBFdFNkmbghsivOT3SsDdZzpWOu5w3O/4LwfniGqqV9NUp+r
TDH+ZR/filM+vl3i9KBqjhgwm7UMiuPYzZgppOrDz18OpXn4/f5R/hWf5jcsTOJqoTZBEzPdVe46Pv
eqcPVO550bYku2w50Kinf5srE7nHEArbRo7EiRAB2Dq42xLB0JkeGJMNUkkXK5z9g4kaNknB/Q8/
1zXumsOWt97c8WsKxa1/O6Na1o6JpU+eYlB6PjFbQ//tqhwp2PWBzj7WGcsv4uZVjr3wimXqxyNs
1OGP2L/CtZfhV8xz+t39UUOI79QxQoewWd4DF8gDXle+Y38Gu3G8rfVNxNqpzzbMTvemViiLnlvji
0tX1y1Orlhsi2muhM0ihSXZGhVlGyGMjPiMxLPxZ9LsN3x7sSt8VsTf4j/lcGn5UqtNF6aaNYyiXa5X
WuNT0xcoy1MrNVuim/W7oz/UN6jPRL3SLKk8XE+EZSDmj8uxBOyxmJjTsAMRjPLAnhZYHeABA
blQhQGZFUhqldxelyXQZMwhdrJoWimFpt4Ou7CW3EvZOxHsYj/zJqhZp3F7JhKKfDZsIEN02NkjH
YhlQxVg8/o+4GbtePPXAUFBse8OoJG7TPnPohMpk6b69ppwOn0CIAjhPPTufTxQr8ifRwcqICsNg
OOgzzCkfNBHq+M9H884GmOg3igg6OXD7jp0Sum092sRd3Nsr056bmTpkOfc1qzHKCbP/lf1iM6
R8Kfb7w8XquP14McJ2sT422JH8qPxmWU6xwlj2V+cM+CmdDf+kzDOZlH8D6v4Wdty2LBu6fgaGj
3xi33nndJpv/PXRvXf/Yo9mJDsN7yrFt36+Saqla8/9cr7x5Gz1p/st7E7xXdu2ntjMzksLt6wpy1P17+0
0V//YXWfVV9vDITVrPo+iN33fLutZiuKqPJwyfZavZ8IEDjcLe5RAiJRVzEH5oSnIQ0uex3+vsuqSHY
FrwsuSi4OLkheV/w26EfhrDL4ZeCqs8r/n8fNCf4it8nchVZAP5IX+lf4FXn828rZNI6bixriqt1ExXZ0r
NeDk0wUhmWenZUILaFqFWUOtWZs6LYBTRI/sj/4iwkUgVrkMmnKV5JEGzY2aRKxszwzo0kH7H
+siNh1hB1eQqirDwmd3Dx3YPV1TBFabpVYrHJsUKqVzrLFF3qwSi9TAEbNPhz6ihjgzOdIf3VOL
Ma6riM038PsG7jDmG8sMxgjWLB1gBlIXgBV1D+YoY0wXjo7TQhKE+jQEPAjvtm3ZCJEUkPxATQ
R3dw4WDvpR6fDRp8KRzKzSq0tJLt2Zg28AbDAOYNcQJgE6qBFADlXh4YPx+o2YjeR8Im5bQiM
QTBvEMY2hPi9FcjhZjxcOp1/99TN97Uy4zPqTogvMpldyDx2Z8537fn7J9GXts/AVDX8qbZzbesIfd
bpCPqjeeX/n5qesvru/eUIRY1Bsazuwad632ovKokUzLppgveoeF0i1TJgzLtlYuhCkMhuslQvWEER/
MGfMdXa6lZlyLnUv9a8LrA1uJ9vVF/QXAr/V3wx8zH8sfuz52HeG9zR5mnxT3FP8bYFOdakqjHc3+
hsDzGputXMjt8G5Ofile4+/333YLzlsLYUztD/k9mYcdRo9EyzOaCPRVBvALJJbh26Xgky4FJlwHar

bCroatAhtl4aOolWB6FsdQjUZ3tFgHsLpQWlh5g6G5F5yrBuSmDqYhfFKKDww/TZI+GvqC93Xn8A
iZp5JtaOSo4GkMBXWwY61PHFd1LF23/trpi3zYmz79q4+tT7B/8PkPyafjZs66d++RXZcvq/nJ8ziJW
WBjZXuoJ80C2V1p864itNUc4+7kO+VO9xz/nEBn0Q7hQemMJC0v7ikm45mMOt6XCU5hWtUpvtb
gg5LkpeuonBKl5utQBICTVCEbFQ4tiekrRk4nCm0pxsV6TAxG5rZ8OcNuyGUG8y0f2QysUKOWY9
LEuaa2IF8qL3Uv8i8KLC3ic52xWP3IBCF1MYBSArgUsAdEwF5pfXHBE/Oegrzk+QO34WDeXdN6
05Wb7lh89cZdl3fiFGCzAwfvJ/rZ5XsvueHhh576/m6Y7wUw3xTYihcV4f/qR/rwGbNNaX5Q2qlt0x/h
9shPS09rfSFR9OJJ5GK+Te4ofkQ7zB8OvSi/pL4pv6WeET7XtCJnkc8EL/GZDIfG6XvW92sf47Oto
Thr9w4DevltEwiBe7qjy0EcATeNIYeD4Qyuc9Ok8WakmrH7eEWht48p9IEiuzedACm99CUaHYY9
3+2mq9es4g5QcZcqAorhGI/BiGqK5xcvK95dzBY7Y6KpOTMg8BFESBeyRzCq0xBaBunquTdglnu
zAbPYCQ3AUIDilR0Bsnk7xLhhEHCFTQjhlvcIXNH+wOilADV21LC/gOADdzMd9AGDdvsPSvL59u
EFsay9bN15nKJlzn68wwQpOehDHfTxDhOEZS9tdwIVT6ch0AFzqaO1wW6US2Nq4IEIM9TGER
Ozg4+nEGSm8k8caPh4n/XJN5di72uD2M3nTea2Ky+cl2LWzPlaSwvGI9bs/P6he98DW0hbL1pH1t
01CV930/qJE79u2741g+2yo0gNnmYuWB3ZGCFuVVvs+doPWM5aNYmBeTC2uI3WmiSeSiczlk5
vZ9mcijkw1GudZ1xnPO4JWp1/QnldFVAOf3t5a9UpNW/I9wBuK6qmVKpayuE3fGM0FYJioJRawC
HbAmxFO1y2kA4qaqEvrywYQKKs0l/NFAxB8oVt8J/PUYcrcaZo55DHUENQfEIgyFdWKMIQgDqd
FAyGQlvG4rHggn2mjOpKY+5g7Zfed3rE//RBPX8cFdwPEGZFgQKMxgBkD85++AFJzjqO0fk6Qa
MuMUm9KAk22+dS71LyxZXLEovrQG/RTmD8xuj2F8PLjyiQKM+BkSaJKIQLDzec768FI8gRsrn3N
BY5tFuOfrmugUYP/vzHiycv/zpLdZfPzh7e9fiezYtWXh7W6rJVxzzj01c8Z3HD215Ays49KMHzi78z
MA1Lf33OMijtj373+997uJf+n8X7IJ/sBFzzowNm2oLcDNVpH4hvtD1e/wPLAmcnyslc11LXBzGxON
1uT2MI2AnFWqEESRZ9vpkP0KKnBQIM1qa2SfhYQILIGZQIT9emtka6A2Q5YFTAfJZAAeQN+n32
W4L1/b68Ckf9gWNbEHwwN5GilewNzRyZOMfpeyDIFPDDrEiDbEQYbEL4msx8UH+nLHhnqe7+L
FNR67c1RGxTkRnnNd2Q50FPDz/4e5Jyzdyd9Lxu6ZV9+6eUP+U5g02Pa3gYY/Dru0Yry6H0kws
qxLzprSdln0SPulo9lr0mcSVyJ1SeulXjjBMbyAOJYBFDfRK+gYfDNHMOI5XmBllkDMsG0xVpphg+
Llvm7NA/IOyDnslpw+QhRWpEdLz98ulJ7Zw5i1zn4xhU1+8Q5oaDNoaD6MUEF/60fM8HsHNVeW
oc9YFxyTERid8fApaRG/T35Wfkn6pfyOLM9kuhiiCQGpjb9MXMVzh6X32UH2LPt3npsmTBMX8ev
Yu9nvsLu4nfxOYacol7BuPs2muUq+UqgUa7R2tp2TgZdAGiDKnCwxPKtwLE8X6xRFFGRGIhW2j
1xvhrgasblEwMJCjShJ3IMwLe8F1ezNlzSLzjuoD3UHwKO+ujhSqJelkG+JLaPexAy/dECKZZBdOK
FFErQCWFVdoTYC/wTXZhyExHae9QD+pvUb6++3cwNnh/Aq6xv5K/B7m63H4dHntDmzn5Z/zAq
qS246R3q4/dxR7hXuM44r4bq49VwvnOAKyw2ESWI0qjUUP9NayN6qivoiBv4Zxs86xaE+B2Aiik8
oR9VwLdz8CxAYdXH+9UMkxEzgUyilVwkXhRoTahRpqZiptRV0VOxu+lhfo/wQ/UQf0jdX/FKxbEK
B6qqqZgOHzxb8X4FX2GGijJZOO6xP+SEGCuElN4KZrJAWbJZzAq6y5UKFxUIUzKYnlNPul3mvP
ouF14GhtRH2kxnKJyMFMG5ZUW4qwgXwbkny5LJFGUcBxBK2UFYytLebIBxp+DSIHkBBc2wlaY
yKXP8eZma1K9T76cYZ6ok1ZNIUCqaqk0Np9hUsPyPLaNEeiQtKmBlyxDEO0iph7pztBt1XbtEVqh
yjK49rEhTiozTnpiPcmTDZsqG33bl1JeufM6rb8HMXUcXbatt+8HXVv6gHHw7kpoxYUm1daI423DB
kjHWCTZ576OzZs+eNf9rrTvynWT+96pbJt21zSKk7TvzqtrueDB/tICRZTtBZ3602wwIHsMzT1wisn0
sBm3prWkr82Od421ocwkOjVcVBagawUk/sqEN4WG4yf8EbbKSVB1UvpqmfolwKj4FUe5fEc6W1
L+BXMEXRlle7F8gzRYSAB3baZ0ondE8+cY0AAV312u5nR0lpPjxhU3T7zhglbDJXU9OXHLHzRT
XLgX+thNmqqHb325OOolPiJ97PvexL5KTHHEHuaBEovU5njin+zsB2soPfiW5X+6Q3yO+4d6U31
BPcCf6kpu8Rf0n+F/9T8QWVWylu5u8QGZdthYpBReRIBW+zEOoKLw+TsCOG/oWed9urdQXSO
hr9pKX6luCsSwMspqEPuUSMG6aFff6UiJcmy74S5y69M7/rLzhjvfpdbnd+LothtueOCBG27YRu
J3Y/5O68XP/mL99l7hR773yCO9ux55hM73Lus6djvMVwd+vtOsbvJM8hB3hmnWmj2ZcCszWZvs
aQ3/lyzN4ed8yduHhH+ERfCfkE3QBcrTTb+i6E6HERNDy4GTuyocDmdS122irixHPbQyEckW5tk9

ddBewz0+Ot0CTRiNTTTeL+IXfZWn0/U1H7V0mopAJpiiVP3crO/CfN2Pr+nHxDrbP3dLB6jYf8+iBb
dtuGrxJIDt9Kut31t5a8h6u212/mOm/+Bj3z245we7wSA3lsQ02nN/xCzfzmHJgWdyi7iVHFPjnutY4lj
uZmWJVViTJFnVYJVM1QyVqH1ltVggC2DdDeLkcSbpUKy2XWCm03r3bTea717v3uV9xs24dJTFjz
5+QHtyLCQ66sv24CBWE8BVzHsoFpx5HgUK+AtbdPK4gim7Uvt+YSV+upSuN45pADjHbpkEShm
AbuQv3UoueeG1rV+dIF5834dlaNrn92tb6v1dfsNf6C8yxFuxZhziWkufNo7yLT4gpw2Ukdrh3eLenH
qiUBG+bl7if1vodL8Y+TJzRhuJ8hTZbW6g9oGx374n3q8lFCbO0Nbk4fnVyo3ujd0P89IKpMXkR36Z
M0TqcbEL40K8NJVsVOtjtKpTXyrwMueSYgEtpcbj8YRQGjervq6u8a71rapYWbnJd0fiTt8DIU/Gn0
xoPXiLcXfgwcpHK/dX8UbMb8YSGb9ZVJlp8eP3/dhfJ8aml20pl2VmIJpC1XZCxAutOrcG0VrqnC
VcWxWh3rdZD2jiCz3cMlhbgkaRCX0mv6qMjPAtra1YsRBLFXgSkOD6KR0IQ9jzGP/TgZb4i1xWbh
TuNqvNQYwj2CBuKxUm5R1NJeWg+i9m2cmV6ClfaPEI2n4N/IL6ObrnuMC2k/Zly7lhfOY/bhcZSe
nzsYElp4TgYso/NMOxcq+GGeFt8h3Z//Gfx1+N8LK5qLBtCI5we1VF2f9AYk8UjCZB9HC/L2LXDC
MQ+hAvVQ7YL9+BTmEFYt2uJrH2lxw9XYmxORSyez55iCZ2C34Rb++sME+5rmHBTw6xvzBi0O
mWYZRXQwH2dRoldCGKN2SET0NsZwtNDwyEyMnm7nGj/HE/Tw9PpkehGczlqjJH6X6G60w0/u
Zyd1pUOv2xKijvrLlcG5PDpYa1Z9arNdPeASiuKf3pCaUYjLzN3Ah4WaoNA+SHUpUrt2iCNfl8tDdK
/+0GLRbU45L7hqusby7y+ydbjl9/yzofvvF5ufe6aP3dZbbQoiZ/rnHv6s7fzuCZ96ezyopqoz+tgP3/Og
3c+c89dY8+/sMSfKPYVLZrSvuG+V/cj+l+sT5J7ue9CTPiVWRFFkLrJFc7xjimOTqcQ9KEA4/chw+3
xYsNNvDjASllsqAEqbicyeo39BtMF3VGDMfowe8CHKWQeRD76ztaNpkNVpBq5BkGWOB9QAq4
wywNM0nDP9mW9u737vEyXt8e71fuK95SXQ17dG/XWellvMLSmd5RMtO9vBjYADjRj7zDR5s6
W6bS97pO51r000EKLYP2u15w6XGgEa660bdjsC/h8toyNajQaLnVlaivqy9zkZuOKqmi1JTAgm9cc
IOZlt16Kw6xyWPWrNvSReF3KutmxDT2AfzrY689ZG0G+XwLUGYmmwR+sMs0LnMtdm3jGlkP8i
2kxdVO2l0niGBnPi5W8SPZ5/XKEu/xJn0+RAHS4bdZgh8Pg8//P1iCJH5JD0R8SsTi/5wAFULMf2
MHuUK5JwmTjBWm3dBA5lp448svXbvJThYcml20opKHnW9e8EVe7eRXitwbOGEjpXH8VFIKW
CeCvCgeTBPBYdNH1ceqskltOFpI9IGEoy3DkJvJzPR0PjMThbzjCKKsqpAxkbcTEgKyXE0RnlRUc
G3T5n+SDQjl07xoqBShiqVDBqvbERSAZKeLLGm2vdSJCPDYiRhHskoS9d9m9P2AkzYdCtIzhVZ
kgjBPOxLzbRyaAaKyjOKVqLVaqbGaoYR0uWs3GEv/9eaCkuaFTbLdrAMO0BqgaD1mE61HuEo
QAIgDg+rPwLaC1LjSgamDOYhUueA0+7U8emzz08KLWhiGYLt2OkerKbarx3DMY9BSrQcSkKesW
Tj10niDd+i/wDELpJf/4NBF/jFjSHFBpirItMuW6Q7zrnLhJZbsEPXu/gN4ZTGiuKIDfDfCNqEifhTvw
NvFKQkzgtNODxQhueluxQzvBnBKmMTQqVcoYdL09kp8k/ZcVL5Flsp3w1e728Bq+T72e3CQPyG
+y78llZY1gBUjQ/G2Ur5To2K7exko8NyuPlafK18h72KfZleYiVBFDQXEAavKtgz6D9sdMn+rKYFY
WWETsTkSSylDOjx2uGJMZtl9ROGY6/aUZJkkkLyESxyvKyMenFEx3TQM+VpKI8yLE8RwHLEK
UJAVxkBge4Oskmh8q4slObbd2TGM0hp4mdQo97T5VKNcVXiZbeE5H3QFaeglO1XND9h6qKdj9
yBty6e7R9fXC3mitxWguWNkhOQoJJJ1glZGkWE1hN9fdvQLTpg7ySA9VZoxhVLzeuhdf9swLelq1
A2+29rz1DkkQxnoXl1pS/jd4svUU1arDmsFeClr14MyT7nIOe+jUA6ozl/o1Z0agDU8bzb/nSOEII/EZ
SANZTXHwOkEenvUQImEwEE1PFwT5PrwPTN2p1TjKUdRX6+vyMbQIYkfBZMaujbiLijM+sHS2m
TEDwQx9W6EPp0yJ2EcEE3rkxs3ILGrIjKyXeX82ggLpqfkgNTi87apg7RWTNVPHwc+nqsp2DtYe
2Gdk9apmgWHXaUasfpc+34dQHg8gPABVkcDw6Ct4VNPMdpugp9OO+RxdJfMc2U9uicljTuQ5a
iZwQHtD8Bx4V6dHlvegoNJxFMpuxTvwGnrDE5YmyeWTbxs/fQZ04IX1i+4lgju5CB/PUv6cwvOi7v
e1b7eiYaH0UpmDn6YO8wl9A+Xsd9Fkv3ndSpqgGg027ff9ptkJmhqP5LIM13QkRI7yWcAS/7TFL+
w3ktTfLv0AtMewGV4G6mHJwSOlla5AYi2ANuKJzhco59GFI/p2xt4m1UKnl6G9cJ32I/+99hP/rn61
zVue9g9B88B1kDuO3cd8T/4Dsi+j/sfQ94VNW17zrnzDkZMich/CkClgwUMSDEEDBiYuMijWmKA
WOM08yYG5MwCZMQksmcM8nMMJ0JKaWUC5hSSmIKLZdLKXJTSnPTIIsYKUVFJCpULRWliv
zTUKRERKTK/vY+Z4YEsD997Xvve+99sL+11+/svfbaa6+19p5zZsLMJzutsTFPCben/DfGpNAHT6X

cz8ewf5lmWUlvCXs/V66wlt5xw1LYt0hWIG+ivGipvVbkdawopHQnRK2//OLSr7NfZ2Jh4gHbldshdT
deDb6X9L3kO5L/uf89/dtSxqZsTNk4YPDA3wz62qCnBlcMrvhS/pD+/w+Wkf9QefBmuVlulpvlf6pU/
W8p37pZbpb/C8tPhvxmyMGB5Wa5WW6Wm+VmuVlulpvIZrZbpab5Wa5WW6Wm+X/h2J8uSrNZ
v/hhtJJlpXGU/+e22gKST1HUPe/eokeQcvjqDspj5zAHirmdSdkHkft6OIA3Yma/XtYep9iP1RUzWuJ
z5LlrxgWyUoxGYmK6ICJ2ffvHTWxTEOFsSZWgKebOIHK43qssDjZxP1ouVBo4iRxq3A0/mtDd1u
WmVgg2fK0iUWyWP5qYokmWs6a2EKJlk9MLJMqDzexAjzOxAk0Oa7HskMjt5u4H90vZ5g4SXhE
Zl8/LVgkzJWsHDaxhYYr/Ne0LDLaE5XzJrbQEOUkxwralQSriS00ULnCcQLzW8JwE8NXCckcW9
GuJqSZ2EJDE8Zw3M/0v4EN/xvY8L+BDf8b2PC/gQ3/G9jwv4EN/xvY8L+BDf8b2PA/w+z/rycn5Jg
Ya0/I4tjGvh0podTEFhqZYlxV+RojJmZr9HGczH43IWG9iS00lqGF4xSuJ2JipseQH8R9uMPE8GF
CG8eDuT37Tczs6eL4S2gfnHDCxBayJxjxGsLkrRYTM/ILHA9j8taRJoas8SDHt7KYWmeYGDG1Zn
B8G7dnulmZPYbfRnH5QhMz+VyOx7KYWqtNjJhaSziewPxjbTYx/GNt4HgS17PWxEzPcoatvfxv7e
V/a691WXutS+0lr/aSV3vFRY3F5UmyUwYyYDJNBSokD1WAz6E6qgXpFCQvb/kqrnzArC5DexW
XSEPPfVSDYqcCtM3HeJ00flUBXgHpBtRu/v2UPkiUcdkqLIMG0rk+N2QWgvtoAdrqqPLvsuV6yaw
+c85CTw1myilHt0szx9npboydTJlAqdBRRfPQW4d+ZoeOk/RG8hn8u/li2ifFtc+B3Z+3uDCOsrnNjR
hXC2/Z6UHMUcnnZL2TuOfqsDur+ExzeY+H6y6DBRPRI/X7OM9VdyH7Jff/JB3m3basZp7aRosd
GKkH9fMt0FwP4+JnX/noBGDSm6rztvqULT5u5fPF+QxYnrtaPFxm5jkPHNMhXldxjV5+ewLiaXzPja
qnOvQzUjWmOusjVthljZ4esl6+UxcMPieXwOwx+N3G7mkRuvwbhmsvMwm597xM2z9HpPsBE1
HKVCfjw4y8By0+4b6679B9Z+Tbs7Hnsf3yOxWMby+EYriM3+ebu+0itGbCXGWnQ+X2yHMP3GW
t1oaeQrr+O77m9lQlmfqFfw6NSZtbEqA/tx5eW1nVvbEM9mQw+TrlHE38qhtCftGemTp9oLPRX2O
XW1dXrQW2H/ap3PW+cr06vqatPs99XU2Auq5nt0zV5QoVX4Gircaff5qspq7FWavcyu+8rcFQvLfA
vsdZVfrCXWmGWMnFVX485yVPg09NnvTpucaU+dUzXPV6fVVerjr7VnZHDxSUx8TmFccSGrn1l
jVW18+0PVIWZauwT7IX1JVX1drnVs3z1NWUaRPt+WW6r2peVZn94TJ/rRs67ZPvnZbhrPPbF5Y
F7X6twq57sILKuldrtrfZ3VWatwYdZbVuu9dXhcZ56KkAL9Ps3grfwipdr3Dby4MYVmGvwZy1TAU6
mA4fb/X66tz+eboddjR6YEivGcCraufV+N3wnj1mRF1tTdCeWjXeXrGwHLP7Sdf+zdm5uJut3lehsV
UyH1+bgA2P6/oKX1FqFWbRKxaygPiqMKu7rrG2pq7M3dcJZcbSK3x2rKgOU6H2616/bndXNDA3
Q8ZTUEPt66G+rzBzkVnz+Q7T0dK7Rye/kISse69PayXfZ71bcvhYvXebtEx6WtorPYP6Vzecrepzs+
UBeYAbkO+s19+n92t8f2r8PNT5fulrwXvgC+giRr+H9t59Dj6id0su5w18JX178qGXWefnryts3wW/0
Po+FlhGWWZYvmL5qixTMs0y0/I/LA9Y7u0zsvCGvnyAcWEy2vu2smh4sZ4+cwgD6Jj0ZZxWfb1W
x8/JsvgzCPXcQX/4gi+tIYjdPSeS0NNj/MLtHPGZTPFeyziimW+w//46x25ueLed/+vBP/qnnqv3FeTN
Sk+XaCaZT1Aq2DnhErTi4+FiBQniSvFHJlmtYivwj8UfA68X1wP/RHwC+KfiOeAPRdx3iZ9KA0iQB
kq4h5QGSTnAX5MeAM6TIsBR9qthUpN0Afjh6QrwX6WrwD0SbLaQRcM9n27Rgf2WIHDIEgJeZP
ke8GrL94HXWNYA/8DyA+C17AlFniJPJUm+W74HeJr8FeDpSjYJyv0K5IXylDnAc5WHgQuVR4Ad
yjeAi5QiYKfyKHCxogP7FT9wg9IIHFC+TaKyVPkO8DLlu8DLEzaRkPCzhJ+RILA54dfAndb7SLTOs
oZJsn7TitVZo1bc1Vt/YsXzl/UD6wXgj/thln7Ofnh66hew4cnFlmhLlsmWbEsFHm/Dk6ltqu3nwFtsvw
Tebvst8B7bXuBnbS8CH7B1k2h7yfYe8Pu2v6D9rO0j4Au2i8Cf2PB8Z7tkg+dt9ouA3+G4EmqoP4
Od7d71eeB96nngT9SL5CofpyUQkLSgKRhJCUNT3KxnycxYy7SaO55w+eGt00/Y40FWFGhFX6z
FlmxlquL3cNby6zzUFdavagbrEHUIXiD+WEx6mZ2b2/9lvVbwEusS4G/Y/0u8HLrvwC3wFfMS+dNn
4jwxp3AE213YS3ptnS+3j8Dn7Gd4Wt5FvVz6nNY0fNYF1vFENS3JN2CtQxNGgo8jK3LXE8irRHa
SC7zlZWTFv7QV0Mz5vsqFtBcT0W5j4pryvRanEjYLY8UZNVxrMj+/hjP0ez/SnCE52XuG+K7iT0zJ8
WvRcyQEL8SsO+gJ68w105DTHkrZ5DJJpbQ259SFIT4aqmc1x5e1/JaZy/XFOJ1M6+X87qF15t4v
Y/Xby5csHABneP1JVYLxGsrr1OI4u8MXF+L5k/PxLjAfp8PVsvsFyNgaSJWrfJ3H1JoAA2kQfDHI7
CWW2goDSP264Yj6Db2a0a4k2487kZtlkncR9c4vPA5Ph6nZzFOuRqcumFaQitoDa2nTdRGHbSL

9uJ5/IV6k47TGbpAVwSLoArDhVQhU8gW8oRCoVjwCauFVmGjsFVoF3YKe4T9wiH+noAgLMWs
AgkDdsM28BGXYCH4qBaDj9aN3B8z2OCZqwx+T6nBp3UZ/N71Bv/aUoPn1hr86wUGz99OFvbjP
g9lkC+Tv0xIRQkjFCmG/PPS2PWkOBm2ZUAnma0u3NN3mXwys1czuLp8hzynPRcNq6qPFXhqp
aqTcZVdWV1qHpV9UbjakHKgrELpi3IM8bXJJv8ssEXvsqrlrLxbavfUHq49W2epG16XXpfDW5O8E
70zvYVejzfsbfFu8u7wHvC+7b1Qb60fYVhvbv4/V4GcNbfUXDe57xeB6tsH9IUOu4azJL/IsExpHktBf4
R7y0BulnMijVihUCjridE4cJ7rFajEgNoNWiKvFDWKneFg8i+2SLE2TZkteKQR6VbpqGSxdFQ9bCi
wei89ySa6WddYnt8rHleFKsUQ4pVuULvAu5YByEuVMqlqCJ2Gbdag1y1pqXWrdbu207rW+3y+73
6pEa2Jr4u7Eo4lnEs8nXrFI2kptq2zrbPtVizpSTVcL1Bp1o7pLfV29IDQhqSipNqkl6ViymPyRPDN5dn
JhciR5S3JX8mF4aGxPB03vOSJ82NMhfAr6rKdDFECJPUfE/j1H2G9A9TyO3SBB7p4eD90LykLL9
J5scqLNBZli9k4j2iT+HuUA0FhcJfD3Kq8fE5O18F72PuYA0FhISJA7ArkjkOvA7mW6UiDJ9N0GGs
n1eigDEtngOaBcUB6uC8AfgawDvAjkyu+GgpKgJdvUkg0tj0PL41xLNnguqPfolszL7Evi76ym9PwB
ox7HqCMYdYSvKRu9ucB5PWcx6ghGHcEo5gGsxVhRH6s9GOnByOxe83Xw+R4GLuTWPg4/svc
qO+I27sHH+azsXV2Rv587AMQQfNLD/hvJr1FL/D1e/g4vSBYn9UwRM0F5PXniQz2TxcKePMS2f8
8ojBmFul5GXEcjrpfQT1d4nDQHVRcMnpeQM8L6HkBE8eEX+eLKxV7NerRRlygMb2dlorcJU
opPU8L9wFmgYagp4U0C0gO2gMaBzoTq6JjRvRcyo+tp8wETNOgm6Zz+HAHDmYl0ccgpwbCWL
5KMZnHgL5dsi3Y04P5uzAnB7M6REuYWt+2pMD63NgfQ70tYtJoJQejzgQegaBhoKGg24F3Qayo
+8OKhbH97DXS2Thf8samUldL0H90boRYzfCsoOw7CAsOwjLDkJyl6w4Cl8cFleBRoHsoHGg8aA
7ew7iHO9/9Tly/zLW4MEaPFgDogx/vQU/vUWjeeyMPBqFPBqFPBqFGY8wa01ftAnp4HF/0HITVz
HTBSuOwB9HuD+QQViRB/7wwB9H4AsPVufh/vgyaGzPt8XbTd+kgsM/4gTYM+I6K3g2/0MRGdL
T9jejsJLJ+ChE8iUNKz2LtBk0BS+kiOQPQKJI5A9AqkjNBBSK00rVsJ7HZBcCcmVmPFIZNaF2V5
GFNiML2PGI7HuLmhZCS0roWUItKyEx7HnoWkXNO2Cpl3QsgtadiGGuzD6XcRwF0btQnbvwshd
GLkLI3dRCkadwqhTGHUKo05h1ClxcM87GHkKo05h1CmMOgU/v4ORp+DfdzD6FPz7DjSc4rvp8x
quH40RPDUE57vIhjGdGNOJmZ0Y04kxnRjTCfIOyHdiptcwphMzvBbfva/xsQpiOgzxJMTzBOJ5gp7
saaetIjXAPVNwH9PZs06cQXni/TRMzEU0vg56oKcbZ0urOafxnAv8UI9b/Aa4CzKPon8BqAbUiLus
ZDELUjPRex9a7gd/ADwPfA5oLvCDXEM7Tql2rsUJWghLBmBkqzgDEjPRExudyzW0Q0N7fP4Hw
Q0NrXw0m7sOVI91DP+iFUJ7FNq7Ta0nuF0uUDGuS9D/GFZeDlwBrZW4ng/sAa4GXggygxpAAV
AQOq3ifVj/HBBb1UOgx4DLalroAV8IX/TjErkxD6JnDughtBXCw8x7Hpbr3FvXrOrus1a2TqyREqErz
/ClqYfPCv4N6HHxmYexSOAJ1sU+KeRtC7hX2BxZ0GX41vDrHNOPTlyX8FWfwJ1y/6ulw9XumO8
wKg+jonyUEZGomAN+zdIT0HTCjCu3IPu0BHKPcf91w1cnaGB8/lhcDS3dZmacMGPajZieQFaciO
dWqbmKWnMllxHfKYjsF7x7Y7Hdb8EcuauqbqXMDfPL6tiG+76AZVQL4S8faAV0GGxXgBMltzH
UhDmx/UAAqAgojA2F6Z1Y2Zu3tlViVmX4eZ2+MzszUwbzxqzv7PIMMrUbEUfWWmNfP6WNSKrG
s3rWqPW1UHHV5gH7csloGt8agaM1/LGiMbnT0vQms3JXCf5+LqAd5r7DpkHt91LE/yeOYia3nus
M/RcBZgfduefzqxmtu5w/ZIN58tp0++Tkb81mHWSuhu5yMTjXjEz4H2+N538v1mRjBZZzP1d5v7o
D1+Yhg5YeRnTFrkeV4PpBrT3JcnWB/8Z5xFC/I50M2kRJ2dSzyCJ3A2sdxkWfUYm5+POsFHefhI4
yRZCG5kWztGG/nLzs08xH4KPykqTP+x3WaJ7Tu249IsOBceNbXH7KmJ23SCW67w7GY2IPEzo
NvcP92UZFrYHbewLG5ltynZfb2F0FdhnCW9dwpbNXpiq431LjRnssZ9UXZdb605VjWidAP/1hgndtz
HMf/abhiR61Yfl5bje9zoedHwHjxdCU/n9cqRXvbhZBsck4ifAFux611mnjxmjrmtxPcb9WxCKOPrbA
OPGZJlreixPRUzG5PL9vrzTzoZ/qz2/SaGQ9Tp7k68+yNncdi/KQXoS8PUnm9Zryxfw2f9LWt9AZR
MCLQzmeVP+eXJ1luoDWK1so+rVvZvvvCO/kR/L6s7928SFP5e0AE/w/G3cJYYp+Hj0eR6C4UCyl
zBTbcjaLQPSgJdC9IIZ7TURLp6yg2/tuPKjnJhaesYvYXK/RrPOuk0F6UgcKdwiQaJNwl3IU7/wxhCt
0ifCh8SMOEj4VPaLjwqfAp3SZ8JnxGI0X2E1yjRFmUabSYICbRGLG/2J/Gi7elt9AE8VZxBN0pjhJH
0yRxrDiW0vGUPo4mi+PF8ZQh3ineiXxJE9NoqjhVhL1iFu547hFn4hV2upgt5tAMMVfMpfvE2WI+z

Rlfxqv110WHWEQPiC5EdS6e+CvJlXpwYjrx7O8ll6iJGpWJDWKAysWI4IKqEJeJy6hSXCGuoPkk
KG5IK/vrAToKL5J3OaiFBN8y8LUmXgW+HrSRyLcBflt5zWibyQ+AOkA7QbsxZg34c2Z7jA6BDI/Xd
o0EX+t11xtMvhn8KOG46H2TzqG9Dfwi6AqRfwRoDCfBnwqeBpoqlHkz60/Wb9fSva76q95MXwpo
DKe1vITrI8Bcln3nG8H5xd9O+rX+dJAWd7p9We8s0C5oLn1Z+p1Tak7Vn/eW1h/Pi7jq7+KtjNoO
+N1+SyVJfN99bN9id4roFKfhfV7K8EDfO7YvGm9cKJ3CTij9cDbILsctBZ2rQft9hVxuxj3gbt8Xlw3x/l
y3zJOa32rOGX61oBasa5V3udAh3B9CNe7wY/72jkd9rXF8fvG2uutvv2chvte5zTOd6x+os9Sn+HL
Y2uoz4dt7/sS6x2+0/XFvrP1IU2J+aB+naaCBrK1Gz6ov6pl+toxph3ztsX8FyP46xLzYcxv9U3QtSa
Pm8N1h/p5TfErbIEcSuFDT7fkHj79f29/bgcNFG4Gu+nton9i3gjLb0kUn1dkDvNt9MUA5wDo/HNvi
BtcdopxmftdcRa9toxA02uU1ebcavGv5p4PHsHb8tiBOjFsSoxYwVSMvUptfna9O9u7HG4+CMzDh
qs7hvmUwuZHI5nmW2s/ge9bXVi5jnfTOvWX63gF8xOfIcfavQ3hXLe86vcN6G673geeBdsXaeH3b
kBqPkXnhwL2z1vYL8eZNTBvzp842on+bLq58BMq6H1GdD77VYZdWXw//neAwucNrmu8DzixHLj
RUmNZm0unfuxfYh9h3PIW0orIPBR4LG1j/h21C/SZuA2Gyo3wpuxMEN/4f5ug6Y+2S5bwP2yWae
9/m+y/UTNeL9jNLR3+Y5wGi9EeP6WuwDD/aByb0u/Qme/5VYR4Dvg0TE/bKZsyaPtYPHhzbTD/q
mXjm4HLhF38rbz615BrC9z/b0RmNP6yuQJ2be6Ktxve4G18gjfbveycjAWi6j2Lr0Xb4djOLrZDIwM
X6NmOh7OPI8xxAfi3kmll6wDWvR9yGO2Kcstpz3Oj/YHkD+xW3Hq4uNf3JJ/DNLK/+0sh//TDGZf5
qYwj9HHMw/QbyVf3Y4mn9q+GX+id04/nlbGrT8TvXAPActo6RRJEqjpdEkSXdl48ki3SndSQnSJGk
StN8l3UX9pMnSZEqUpkhTyCbdLWWSKi2Wvk3J0nekf6FB0krpcRrKfr2WbpW+L/2ARkg/IH5lo6Qf
ST8iu/Rj6cc0WwvqJ9FMal/2r9G90u/Qz6eeUKj0pUI3Sv8u/TtNIH4h/YImSb+Ufklp0q+kX9Fd0n9l/0
Hp0q+IX9Nk6TfSbyhD+k/pP2mK9JT0FE2VnpaepulZ6RnKFP6rfRbukd6VnqWpkkvSAfpXun30ms
0S/qD9Ee6XzoiHaFc6S3pHfq69K70Ls2VTkgn6EHplHSK8qX3pL/QQ9IH0kfkfFPlifSoPF3OplI5R8
6hKjIXnk3Vcp6cRwvlufJcqpXz5XyqkwvKAvLKXhXlh1csO2UE+uUgulK12yS7S5WK5mPxYiVxCDXK
pXEqNcrlcTgHZLbspKFfKHgrJ1XINfVOulb3UJPtKn4lN8gBWiqH5DB9V47IEVohN8lNtFJulptlpx
EXkKPy0vlpdQil5OX0ffk5fJyWi2vkFfQ9+VV8ipal7fllfQDebW8mtbKa+Q19EN5rbyW+HfX0l/kVrm
VWuX18nr6sfyE/ASTlzfIG+gn8kZ5lz0hb5l30U/lzfJm2iBvkbFQv8pb5a20UW6T2+jf5G3yNtokb5e30
8/kdrmdNssdcgf9XO6Un6lt8tPyM7RN/q38O/qV/Kz8PHXIL8gv0m/kbvll2ikflA/S0/Lv5d9TI/ya/Bo9l/
9B/gPtlv8o/5F+Kx+Rj9Ae+S35Lfqd/Cf5T7RXfkd+h56V35XfpefkE/IJel4+JZ+iffJ78nv0gvxn+c+0X/6
L/Bd6Uf5A/oAOyB/KH1K3/JH8Eb0kfyx/TC/Ln8if0Cvyp/KndFD+TP6MDsl/IXvo9wr7nqXX2ddg0x+
VfoqN3lSSICT6k9Jf6U9vKwOUAfSOMkgZRMelULylfoneVW5Rb6LgyTLmVTii3KWPotDJWGUtnIX
HKOPpASVVS6ZwyQZIAHyoTIYl0XkIT0ugjJV1JpwtKhpJJHyvTIGl0WclSvkKfKTOUr9JflWKIWJC
UEqVEsCilSqkgK+VKuaDgrnG+kKBKUVWCTVmg1Aiq4IM0IdnWz9ZPSLH9yrZDGKCKqigMUy2q
RRiuKqoi3KpaVaswQk1UE4XbVPwTRqrJarlwSk1RUwS7OIAdKlxWB6uDhTHqEHWI8GV1qDpU
GKsOV4cLt6sj1BHCOHWkahfuUMeoY4UJ6jh1nDBJTVVThTR1gjpBuEudqE4U0tU0NU2YrKar04
UMdYY6U/gndZaaL8xSC9QC4SG1UC0UCISH6hAeVovUIqFQdaku4RG1WC0WHGqJWiJ8Qy1V
S4UitVwtF5yqW3ULLrVS9QiPqtVqtVCi1qg1wmNqrVorlJlgThMj1+6fK2aD8kmYnwvuMPFc8GJQ
OdH8UnCPec2o1uQrQDooBGrCmELwpWZ7jFaD1l3Xdo2E+a7rrktNXgn+BGgTaKtJ29FeA94J2k
VU9SboGCeh6jT4WdAFupWmUY75a8q15q8pr6UNeKrtoC7aR4fM30K+LFiEZGGoMEaYKuQI+S
S5Oh8d49r1aKprz6M4uV3LXUdd613HgZpcb7tWu04CNbj2u5pdrwDVuA64Aq5DQOWuTpfHtQuo
yLXDVeLaCzTXtdFV6NoClO3a5JrtwtOKK8u1ypXjWgOU7mpxTXetBRrnWuea6HoCalQr7BrjWgY
00FXpGuqqAbJCb7KrFmilq8BlcRUBqa5C52WXC0h0zXCec2WT6Lzkmuk86coBOuaa4HzTIQ503
DXReciVAbQHvftcl4B2uKY7u1wjyel86poNiXxIOJyHocOCejZa89HqcL7vKob0cudRZ4sT63c2Od9
2LnWe/F/2mijzv/ch/pc+xt/U9ON/z3lL/2uUYSQgKs38uxS7aCJROWJfvh3UaXLEvnwPkXs3+D7zm
tFLJj8DehX0Buht0EmzLUbnQZeua/tbdJVongWUaFIKaAgIz2/zp4NmmYQ9gz0w0TnGmepMc051

ZjlnupOdec4C5xDnCFAJKMfphYBVTu94KxuclZRmp3LnKuca9DS7GxF2YCyzLnZ2Va5pXJbZUf
lzsr8NwA+B/WixfEj0kUP0EsLDwWCo9FAo+Filjci4h8JR6RAYjIQzRUeRhxGcHjcpviUlw0CnFpl7t
tG6Jzu+0z21/pDIsPYjTh/+BMAs0kncea/T1EPLY8ri/1jSPOHGuRo6gYVF7kKaot0iuq2d90iB+JH8H
Si+JFEuQsOYtEpUAplAm55ySL8igyULb9wvYLUmXbVcp4e8alww8O6iA/d8hoYtwBIVeRqwJpIB
UEqMs/gNBQ0EjQWNBE0DpoMxeeZJr0lxTBmcmOwPZuQcSgjuhKxc0F1TIOeEsFKPY8dHSXsT
acEZEfSahrTIPPGCOZxQxickuAS03qQW0Ni5/zSZmN9Y23wcy9ND8iDnGmJfmL+F9TE6Mrjfblv8
d1GKuP0ZrTWJr3gjaEifWbrRtNMet52RcfzHF5MHFtxyV/vJvXtHG6CMjosOndUWs2hj/+UiyozKQ
HBmspQYGR4ZraWixQ/J8ZJw2NTA8MIFL02ZGMrQstDAZe2SaNjMwLjLD4QtMhEwO2rO1MdAw
m+N8yCRHHHyW2Y4O4GLOz4DkmMA0tCwPzliUa3mQ9PCWWq0gkB3RHccDsyMhrQj6m7Qsr
RmzuKFhqWNLID+yQqvWUiOrtZKAi7JO83lbGvxbI0/wepPmRstSLRwojmzVmngPIke3asoAHoney
Grg20gmsR3Zpq7QNKt3amkAosk9rDTRFXkJ7LSxfE1gaeVXbgLHZwLWwZE1gReQNbXNgdeRtr
S2wDn5rDzzxzSvcbfRsgneaAtsxbp2BLZjVFugE/hs0AL7+9aJ8foCa3Fs05qjFu1yMCWa+Plap+C
QalquBEgdgVODY6JDUKdGR2jeYBrzZ3AqYvfDdaNaI0NpkF/V2BX5lyWE9gDa5uDM6Nj9IFoT9
Vmhlx1bB/H9bYFngJ9d7Aq5FsfWQwL5qljw0WRGfC/rRomrY/WAT5tsAbkNlvemCfgeMtrwTeRqR
eR31eezNwMnJJOxY4E7lq6OxTtwVYjrUFLvEaWJ/gT0YE3f5y5EZYqwY+HSyJeBxHg+7IPj0d+pv
0THgph+dbMlsXvHo6cBXysBDx8gZzlrP1obB2qj49WB2ZobmD3sg6fVawlbLHsSSQHM3T1sD/BX
puMBwt0ucGkWI64TXMM1bX2oLLYI8aTlyWOCqDq6Ju3RVYEa3GXXK9iv3Qge3W+d7AjgmuiXt0
VbEVvW3BDxOE4HMyKNujpwc0Rj14abluGle+IJGthxxJ4hmVvstagZ0abIV8UGYcYpUR02GDgdu
DK4I5INnAXZjkNnxdp1cG9yNv9Jh4TOa/XBDZFI+m+4H54JqB1RVdprcFXomv0CNpbgV+PbtCX
MJ368uCb0LnE1H8Mml1t1VuCp6M5+lqOMxl2+Ijn4dX1wQvRzfrG4OWIXd8SeAnRucXtA0arPA
Js3BbiGK45u3GLZHBegcyfIV2+gvwXqylxSlr2gZcZMQFOCekRNv1ncE3o236bm7tcyEV+g+EFOb
/0EDgQ42BaLN+ODQ0skefFRqJ+jA8wPBYjINRH4U/d+hH/dtRHw9NwLoOh9KjXfr70L9XVx1ro/v1
c4F8rOJiwB6t1q9A8hU2S3SvtiyUGW3DaTYNsdsZWBpt84uwJNulBbcWODQ9ovut/vNRrz85NKv
sgn8wy3aszg/LX+T5EA7IRI+P+ZnJ61tMn2f18v9U6BmOed385Jnmt4fmRt/0jwsVRrL9E9n+0teHpk
eP4cQrimz12xd14rTMWrQLvfZFezjex3G8vSExdCDiaUiB5ce0nEaKnnbsDJVi71cuEiMOv444iuzk
b7ri2BaYvVjUt2B/jWsYsuglnHipoStoxm72NowlnQF7WMWvYp47WdYV9nZ25C66l3IEw1pi97m5
+32xcn6FW3V4mR22i8ezM7YxcMbpGLbG7IWnYzoDTMDGdC5M7Bi8bjCt3HyO9B+Biu1hpTFE7
U1i84vzmjICUyLpjbKLTotw46jkm/Q2sl5/BRNbKpsOB3OQ2SvhAuQ+XvDiEID2+PwZLGekpg2O
ELVUbPwv4VkrXIIproBZxmSvQyXoMmNpGeGfLhbE8NBZrlnxGKoDc1tCSS758WWH6Z4Z8Rao
mlvMXhzw6tbVK0rNB6aOslReAxRGf+ZceS0MYm0saEtqD2hrY1qf7ZoY4mxZ8f2tk00O8l7W4aC
m9UR19H9J+Lvu4vDh1oGqm5Q4dwCl0JHW4a6zjqz44s9ZeHjjZNcFSGIKZ09io2/zly/3j0tN8Ter8p
018bOgc7s0IXYYM3dGj+ZfjqieucloeylNXxlC0wR/C2XXB36S5m6bjbFxt+ioyeSLsr3ZUNS1iOLrX
vzS0Fq/dNYFpTbn+FYusTXM19yKxqdCfj/Mh0b86sKLJpbm1rqZSnEI4CR1HYec+bVXY3Vzk1wO
Opo6GC+Hq5pKGy2Fvs7uRwg3N1Y1KONzsbVTDzc0NGDs2t44MLwMvUPDq5rDjSPDa5qbHV
sWZUdWNI4NtzYva5wQnLI4n5aDk2QGxkMd0bbG9PCGylyGzXy/70fsBjdmhjdHG5A/Z6LTLH+ilo
aScBswYgrsDrc3r9LXh3dgzx4Od0W7GqeH9zaXNM6CVWsac7IVc5EPM3DSQqejJbwfpyLuOppb
kT8lel1mZw7O8PArRi4hz+O4OQs2vA6Zo8ifFb1xw+nAycVNmLEgsjqGg0OQ8zz3HB3hN9lp0Avv
Dx9D5mwPn17cdA1Dfn/4LO5/3NCJVxaGea5O0y+GX2/eoIX9M5o3N2zGfpzuX+c4ED3WWBii5rZ
GV/gCXn1acMLs1by480nWvNinOotdczuLXfOO3rtDS9W6Fr/Bdm5zV8wqdrY0loYv81fMROymkuB
ISJYsSm6qgZ8dqDsQkUR2b9Pkc2xEXfnf5H1/WFTZleB9j6J+8UMsCCAiQjUapGICCNTQNK1Wv
VfGYdQYVouqCqFtYowhxKERCslvoWAchzUuMYQmjnEc2iHGYWYwCvYbJR1jOy7x4zO2bbuuM
URth+UjDss6DOsSAnvOee8Vr0poTU82/8x3v3Pueefde+6555577r2PejzcZcFupLx2+IARxjGq8TD

gOLBMT11iU01VF+DGqi4o2Qh+eLiptepE3doD12GOtDa1geQ0xDD7wuGuoynPcaMuE3qBbbVh
WzBDzyENcaPrgLf1P4jPnzGce5AdFWf41xT+/76SktT5/7yuhxYnb1165uOE+6WcTFJIOwYqBNp
X4eSL9bIN/ZVdtdtB0/L3b+x6fT+l3WOAyer+vaXHzi5/0hVZdPZqvN1RU3nCHcihnYHHQ9hFqDOI
OdbM+CHHXUwFhBvzbsGaTYNVF3aX4pzqmrvmrmtVQ3V7G69VztbtbQrF/mKtun0w+iDhgMfx0DH
cWft1ra4S7A8zHTlgYQ/NtYuEr+zvgFEYInDvZsQxqQJwSVNZ1c39mxrvVd2pK8F1EHdILlKWdrb
FdSWNQ1X36prAB4phjTjoOOeoaLqBeh64WjW6/1DTbYgww+D5l462xiGYxabGw1WP6g42jVQ9r
jvc9LDqSd3RpvEqESSIVXN1XeBpJXUnmiYBn2qa3r+p7kxjn6Oirg9mkBQBmpvpgOawRjTn0Bqvx
rITPLCosgb2KrB2VCZAPD9Xk9gslt2cT/T2qihY9elct+FuBZ4Xmh1INxcRXaKi91KZPKL3Eb8S11tz
racA1h2iD1yWaNjjwRnEA7O5uclTXDvcfBDWBUfTrspWaMvhwX1+Jc6R5sOkQ5ynFM8pnnLiVw
D/KPG7qMwJpJvaKkdq4l6fgr3BmaZzITtwz+bxwB7AQ+VPwRqX2Fy0PwXoMzJdjLg7FPUHO6p
R9rjdbTCeof8PpBW1Hye6EGklWYeqh1pvgTWftl85DIUN4d0vQbol/XG5muejvql5puVO2CNhJgA6
ynoD2sEro/bm+MCaZjRQIO3X62PhhVhN8TG3gX6azMeI9nhWF1R873KcdB51HMS6EdEPyb6S
WVZfTz0sQfOUEOO1vr45rnKyf1eWN976+O9Gk8/rEdDgXTznOfCfq/X6Hm33txY5Llcuc0bAfHK3
NwHu9azlBNpB9GPKPZGI+2N91ytfWacdfU1IY02PxpehP4akrlofBeD6ykw94Uz/W6kmZ9VSKe
BB2N9SmNekdnzXpv+v6T9SneLNhjpzXMVh6vTIZTGNCwywXam+sxwg4tDunmTPATOG9WWiq
TYS7DWubdWNUFZTYh3ewgeovjdpXYHLf/en26t8Bzqz4L9g9363NhvxFdvxF2Fw88Ewe8Ladbzr
Z0V99o2HHAi7ilzHEb5sLlytlKy+tTtDZtqTpTtf7ALY+mrulAv4Sr9zQw7+Xqsgat92p1Rc2Q93p1TU
Oo91Z1Y4PJe1c6l1e31pzyPsCTpncMT5Heieq2hlg4rUgnXDrbyqda1YIVOqtKp9Tq9oYE/7OqdBqt
7mxl9k5VH29l9c5UdzdktLDq0w0WWHnPNuS1aKvPNeTBGkRyqgcabC2h1RcbNreYsN2WWGr3
PLbbkiCfnpFOncezc0syatKSipp4pxY0acmQeyFFSDgpt1jwjNySJ/ULT+4tefL5GuJSYxeuSgeycAV
pseEK0rlZOS3b8LTesqP6yn5vi1uShrusll3Vww3bWvZU325wt1TITyfoiUH1SFVJSw09i4iqftiwq6V
RfhZBp/7q8YY9La2OgYYyGGXpmYN0upeeKtD5vYZvaGvpIJ9aSM8HJFp6XgG1mo9WTzZUNOu
rpxtqWtqqrzQ0Ng5Vzza0trTjfyKht/6Y6q0/nt760+hteicLpjf94ulNvyR60y9Z79HXS8/oD+j/I7PQW3w
CvcW3PWRdSAbbETle8ogV0ZuHr9F7hl+BNjJZMnuVMSayL7M4VsKaWRb7C0g7WDv7DtvJTrK/
ZoWsB5KL9bl+5mbvsAH2GrvCPmSvs3vsn9gb7H+yR6yaTbN51sDxxCr7c66NO8z6uE7uQ/b33C
+5h+xfNKWab7LfaE5pfsjmNYOan3JBmmHNB5xBM6b5NbdcMx0cxH0qODI4DfeCtk07yK3RXtT+l
HNq39O+x7m1Q9r3uS9p/7tOy31VZ9DFcN/TrdlIcKd0SboDXl/hgOEgH2z4C8NRPszwpuEYH2P4
gaGXX2l423CVf9HwgeEO/3nDLw3T/BcMvzFG8V/HvzTxLSHhlcV41hBTSAx/MORXIWP84dDy0B
N8Z+i/hvH8P4atDFvJfxC2KuWf/mZYalgq/4uwl8Je4u8yDuxSSk9KE/D9KaECoAagEaCVxQk1Qq
PQKrQJ7UKncFzoFk4LZ4VzwoBwUbgIDAs3hNuQjwgPhXFhUpgWZoV2kcd38GhsmV7QC4zX5+
vz6S1FE5/GpzHG5/A5jONz+VzG8xv4DSylt/EC09DvubT8Vn4r0/E7+Z1MzxfybmbgX+NfY2F8Cf8
VfK6/54rgv8l/ky3nq/gqkFnN17Jl+j1XDNg7mcVq39e+j8/72W02Qj0z4ZuJwkZWlImwUNglbhALBK
RQLu4VSoVzwAF0veIE+JBwB6BCOCSeFHqFX6BcuwL13hcvCveG6cEu4KzwQxoQJYUqYEZm
oFUNFkxgLKUFMFIPFDNEi5ok2cbO4TdwMdRbSXSmmJreJZ8TSkWF86J6cdQLvFAWFKHBArxF3
iHrguA6pGbISrbqjZCpw2sV3sFI/j+266vwFrRvv5Of6PhyxWDI6by74FPm8jP/8T8O8+thU8/B22Dfz
7Q/YFNg5pO9noi7oXdGtYge7Tuk+znboXdS8yh+4IXTor1GXoMphLZ9FZmFuXq8tlX9Ll6fJYke7zu
s3sy7ov6YrYa7piXTHMF44dh5mEVjazYLDyLhn2AJSRP+UJR4Uu4YRwSjgj9AnnAQaFS8KQcE2
4KdwB6p5wUBgVHgmPAR4JT4Q5USMaxQgxWoyHZBZTxHQxS8wVNwLeJG4RC4DnBF6xuFss
FQ6L5alHoFyoFQ6DpFqhSSgR9gr7hEp8N1D/hr6K3v40+lnrW5Cy2M8hZbOPIFIg1v8T+yM2BilHt
123nb2s26nbyXJ1u3W72SuMC50KM9LfNIOZjrGdvQD9jHOWQn4B4F2gywE8QZk7e5zFftDr3L2z
31m684KznK7fdXp2XnbWE438q04v0XgfyrrlHoKfd15iHKFjzIC81vOI0QjYDtKW5jfdXb47inwwHn
Mrx7S2D7mCGPOkzsnoP0JVv+QnolymCvIngcUfRQI1OXjYMbZQ3opfUB5il6oC95H+yBf1tHBFnl

CrdOjBqqngCJrQh4f2X4kA2Q6QuEabSGPmcl3jQXKkOtsuOTs9Y2voqcyRgpPdc9hcvb7blttqduU
dXHEOi9QnuB8l+ShLCVX2sZrHE859+mI92B8Ud5i9ZW++fJk52VqP9V59ak+KHmgrpkbii5KfmFB
N5Kv+HyAL9B4qHmKv/SofEyp4x BhvO6XxtK3rN4/3397gmwn3KN/gMyfPWgrcJaiReY+/ptcd5y5
DnvOlpyY42V+qS9lokL2x6vvvqcoH2fp4c6/uuA+18IWC8Pian8ZOvCw9K/V4qV+wSaOvCw5Kdn
pUv6V+KHZR+qH0ffcLmfKCMuWOzc4zmFtJy7ovJ8hx0bHNO+MrscE5huw63c0bxJ8cuF3PscWI9
9lLqyj7qKHOF+vql5StcJkeNK9bR6EognhKvsWy7K8PR6bL4Yq2cO4678kiXbpfn568Ya5TYh3XP
urY5Trs2kw1PuMMLT7mjCs+44wr73lmF550dhYPO+sJL7rUYB/Ea6xQOQUzEeBk4xopPBfBxfAvj
XOfI768ttOEb85vutMI77ky/+HH5Y3xTGzC3A30qIF49FddkGxXec+cUjrrXKzGk8JfBLHzszkfw2So
gLvN6pPiNYleVTf14qC+MGdkZ4Zxrh2PA5Vavp46Lrl1U7oprj58sdRwbdpU5brgqiL7tqiH7K6DIGX
E1Uv7Q6XWMu1odk6426v8S4Jh2tSP41mkZHLouTt81jg/vOI6od3Wr1/DCcNdptE9hIOssji2Nb+C
6jH4LOmF/sY+Fia6awrWuAaqf5rqotldhputKYY5ruHC960ah6LpdmO8aKdzueljocI0XFrkmC0tc04V
7XbOF+9y8XyxUx9meRfxnsfsB+VOxoNc/V6+bhUfBJgGxQd2u31oUsCb59gfKPAIcs5W9Aa6n2o
W9gplXdsF+DsZbyWl/h/mz+rlErPXzZVXumzc9AfMocP3rfXot8FsTQhdi0aL5UvpO+NszsD1lrQxc
V5/Klb1V/yI5tq/aj6K9v3nymz2+sYL4XfjEvR3nUuGc21FY6dYjODXulgS/eK+e+yDfaXsXqOOM3/5
YmX/KnJP1cUa49zqj3fuc8e5K9T4W5x3OP7U8p9ldu+jeW5brTHE3+e2zA+ORHluc6e6Dvj2RMtd
hHjuz3lf9zhi9qlgH5Z257qPq/ZBy37lJtq3is8hXblTyscxGdxee4vXf1v8nxml+S/856p9D/pnhf9hd+4d9
vhlcxObpOcp9BzIde1F7XtcBz1B6alnKN30BOUGPUG5T09QPjlcMEbxNnoucpuei/wPei7yC3oucp
+ei/wan4sExeFzkaUfC4StA6fiwRI4HORoM/CifYUO7Pw9CB7gG3Onsyep618Ba9JdwSZYmzJF
rWZI8EnGaJyx6w8ACZlhxLOPDWW0Qok2/Znn0Fk8UBUJQ9DrgE0I7LPkulpTb7SpbZ0mQ5aDls
OZo9nD1s6bKcsJyynMm+Qek2pAFKI9kXAfDqBsFDKD2SPY5PAvQu/P1kwNm2FkakgR2AU+1Z
SC/TOTeXvc9uwEn2JqRXuZ9xV9l6zXXNB2wjPq+CmhxzsmJVf68wM2g+THqMQlvjAHL/sx+qLU
D9x35jn9OgzzmA06DnD6Gcg3TcDTrG0LuIDLwH/xd0CiQeZtKp+B+kiWIYOvsMC2afZZlwvs5mOc
wAOoksJG2CFM42Q1rG8iFFsC2QlrNt7Aug6RdZAYsCn3OyaPpffnHMA2kla4QUz5ogrWLDkBKg7
x+w1Vw4F86S6NehjQt9tRqDMq1Ga4Q12hpnVtTrOnWrLzurDlrrnWjdZN1i7XAMmJhVmdeZ3aK
NSLbbC227s6a2xBqLbWWQw1P1p31D6z1UDvF6rUeAnzEGp01au2wHsvrtp585ZbVx1v3W3tsf
Za+6Edo/UCtlJSI0DCQrqVdUdK2aXWy9a7kECKkqweOT2wvgs1x7JGXxkjWQXWKeuMtdcGvQC
pCGZo9bL1qvU63CugVjCIQF9S8jqtJ0HvftA6foPF2gsIJ6C39XndNq0tlPqP4AE5pTaTLZboeFuCL
dlqzrpj3WhLRSkEW6AkQgHo5gQ5zrwKIG7LsFmyU6DPEVATAFujFrY8mw2lOtrBSUqADog2DZ
DngV3AUBmigK2bbYdMB5e65jNbdtl22Mrs1XYaiA1Uvukg60V21e3jWBrS7VL40W9BUoB5FDNC
OpvAen2NCzGL7CaBd5Pfz8QeNTZ2iHohXAhyqehChbjl0+lExLV2iuAfGGttRd1Jr1BD2pD0T9Cu
Gctt3WC5coJesjC8Ta31Wg7buvOa7X22E5b021nbefyuqFmLvqpbcb2EcblirXANmy7YbttncCNW
COyRi3Jtoe28ewUS4JtEvpaDqPXi2MltafBP+dss9YOsUfsFfvFC+K71nrxsnhVvC7cE29Zx5SRxBb
Eu+IDBOhZnNUr1cB74pg4Qb6jWFS2nDLac2MKNPqWbAVxSpyxM+EoeoddC3qVZ41m0/jYQ+
0mrEH2GRVGreWvPsqOWD8jPLIWCPeEe9kRWXOQ7gmPQbN6a7kwhM8ZrdGiBlqAZNG+esn3
zNFsjYZ60dDLI/jsMa81rxX0nxAhJuS1WphtVswVHguPszeKG/F5JMx/aAFkFVijs+NfPzVdiBOL88
pevYnPJ3H+ZUeArofEXLEcqGN0xyPWi17xkCCKR8QOmLNO2yx4r5FmazmMdpqQKeRAnLiFM
1BYb0nGa1so5KKQr9gLyUUL2wUHlUuBGJSuzB4Yd3kWCUX0NHQv2RVHPRfujYHsLEGPgM9J
hbWvxAu11iMK4NgITTa3cFA4DGN39CkPBt8moHkvdCHg812cf/iMF3OFxvklfn1eOPxKCj73tU1L
T34xV8Y125wdj2XlmZILsRIAR5NiR5b1EGhyR8iEub0bvB98/8uzGG3tsfYEe7l9VTwmnrRn2C1gw
ehso3VjdooQBz6RB3b3gF1mwBYd5PcYfj9Xab7ax9M3E67NvsO+xu+y7oR719j73MXmGvAW6j
vdXeBpx2sdyutXfaj9u77TX201l3Xsmyn7Wfsw9YD9kvYmQVinAsMDrZr9iH0SbkhY98kdIJ+olP2W
/QWvjVf0c7qL2snJ6Z4//MZ+kRjEs/xKLS8aUkE6TY9Nb0hPS29GRlqent6RmQLOmd6XnpX9Nt6d

1An07fDGIb+tn0HcBzQ9qVfi59T/pAetlLF9lr0i+m16Rfwf8yqX9Nv4v+i6adfR7s+sfsT2BfsRV2B1r2
H8B6lWDnL7NlXoWOHT4mjehvXbZpxtndjL06CvmuoEzbOMwxNUwCTAPMStcCD6CX7wEf5oZE
z8plZwPqyTT4v8ST+SQjllE5LNWZltuR26l8ceGer8xa/3oka1bmIaQt0pdxVV/0vwPw/hCoy8cBxa3
ZhT747CfrgvfRPsSfDNBVtltT7c+qQNVHRTfKJ1X2VWwxreKrx1axec6Cfn759Cl8zNerxlgf0Kaiiyjn
+Sodxv3bJllrVblaTtzH1Ff6JuewFkjj5VhCX/0iuir5bIBOswFtKT7PPwNkf1H7mM8eiويipdtatP+B/Q7
UKUrwzUGfv8q8wNxXpgRgL0AfwPmPsdvK1/K7s+ZLzoHFrPhM3Jfv5+RP2Vj2U7Pypfsx7h/PwL
9C/YhC2NeKc+t6YXcN8cVWbWqMk1yOwcX+gr7hRzhqKo9dfuYd/nPQ9i/5AinAM4lvjjhs/0gwCXV
XFTyIvnuNZXuaaq5iXXvANyUrsUegF6AfoALAO8CXAA4KIACxGuqgzExcYn5sdhcRdmbpL6p21D
uw948R7zIP6Yf55vP9LXAeLVYXAlbiXcBHizwRdh3ihMSqOPyonFI0UW/INNnU33A9RnZzgj3AEY
F//X0kVzucYAsdT+fAMzJNtRl9lfAZ2OjnEcARAPES/1fCkSzBMo6rYCYskCTvHSArIX+Yw5nDLKP
uFEaWxrfgHUZY6JO1F/oo7gFoECu7/S3l1gMsBugFKAcwANQD+AFOARwBKAD4Ji/fywVd5913y
9/3linzK0l1p4l43+gvy4Vg9Xr8il57udozAPyZ+rxrNg7GZAvMn8WXf+f115L5Z9wfJZaMxdtP22JPG7
Bl5WYsvOxapwgfoT8IyaATgpAZzvCfzivXqug3y7VsWL85+jvvmnzDIZH3sogAkgVuUn09K8w/mnl
mdPUMXBQNNAtyer+pW2il5yLLKnCgvrjTLXYR7bMxb657fOyG3aLQF+lt+32/x9Vtkn+/qKZfKEHP
zdE33lgP37OWty7fhf8VkoF44fFlNxD3AB4F2AywBXAA4D3AK4C/AAYAxgAmAKYAZfGpdBK5VJ
xRfJTQCxMiQsIEE5qckAcJGaAWABYJNp5Ns+AWyWdZMhdZsMoXIO91N3yOD+GNjMNq4rXrd7
Xem68nWeFY4VRevqIXIXNK075EtHkEPpyLqOFZkr9q3tWOeB0vIQbv2KohUIK0rWHUOMuUStO
yldySX3rutZkQawF6UAr1eV+vG3nk//0pe+7KGhb3p8ir7dEU3f7lhBX+2lp+91rKlf+CbSb3xfom90fJ
a+zpFF3+Xlpu9yWOiLHDn0RY6X6VscG/7g7XGciZN+NTvAXmQsJR9gewA4AloASuR8L8A+gEo
ZauW8CeCgDldlOCrnXTIUyfJOAJySoUuWrQDcM1f6A5U/A9Dnu/9iij7G8/wpJXxx2q9MVAp9HZJ
+yc3oyy3SN1uC6ZfcRvoldxh9syWWvtMST19oWUXfZkmkb7CY6esra+mLKyn0lZV19H2V1P9vcjl
2lp1b+BvQqnq2dU3OqpOY1qxuF7JGXJO/ZvuafLp2YC7RkBetKZJKAYjlp7SxeJWQRDnlYPJJ3A
ccnzy1JJ+cEpmXv+qIT75DqoFPDvIO/Blmf5L/MYT1n/L/yBL4/8aPshe01dpqJmD0ZGLIOyEXmZ2
+xxQLYJK/yZLkq6+B+uA7fA8/wIL5QZAVR3XioUQ0YdkeK1MYh4BfXUKMXxNiOWy9qoSJmWlb
YxtXxiaNJ02ujl3xrKxYaYNkWpkc27YyFSBjpWVIHsnAvyYb+R/yP4S2/47/O+C8zb/NeL6f72dB/D/w
/wCa/VfQJhj6NMT01BsjaPZjFhLyE9AvAmbclW6Int0VsOXgyXrGvVmFd+bwJe9x5ny2dUWWOSK
x3Rwdl2qOX5FiNq8wm1NWeMzpK3abs4hGcJpzk66ZNxLcNG9ascVsXGE0b0F+Yqu5gMpA3aQ
7ZmfSPXMxykoaNe9escclSnLjoQ5CsbncB9CGWibJjTf3+wHopgDoGK/SsTRpznwE9Qa6GGmQc
4wA5RjNm2S9elAH0KmUyi7og/Kxnx3APwT1TkKZXpInFYsdNXuwf0mPzPV4nfTY7l0FoD4+MXtp
HPGbXly+YMUZ3lYvM97wmmEX0xp2G3YzvWGP4WvMYPI64essxPCnhj9loYYKw34WZvAYqtm
y5/Zhjuulb4KFMg/sW1hSuAxRAHidKINCr5VpgFXg90k5Uq6GpPUqOo2xhPMSwDWXBFE44SjbG
pMYk5hQHxsdG50UHz2SUB8THpOTeCdGTOil2Q50FEBm4j24PhYbvXpg1bXVw6uHY3JWD0dP
R08njgK/PKYEyqyPyYyJWj2++grcv5L4CMreWH0jehLqPo6NjtFH34jRx4SD9HwFol2S1WdXDyQZ
Vw+ARD0CIJv0g8SFJOkYGx09lukU82j1eFIE6gs1gV59FnQj/aDmdlw+oRw0C09Kh9YzgZeZUA9l
9T59JIGfxDvR4wn1SWbQImX1cIIHskTF5MTkRN+IHkl8Ej29+jaUyom/mDgXPQLt6pM00MeH0S
M4Svy3eYjR/Jv8m8zaF5//PjMaXAYXeECxoRg84CuGr4AHIBr2sXDDG4Y3WCR9dSwqZCpkisWE
TldMs1j6rtiK3ynGOQG2A+yjKGemd0zc9FuGPDny0fecWS394oBjm1TIMtke/G6OrxwH0egvwaN5i
EfUPrWWQK3ht4b15OmMPF1Dnq4IT9eRpxvl043k6SHg6R4WRpKwD4z6EEEx9WEP6dMp691Lb
LxDPS1pz7KKKd03WW11ugLTmWLnMw/+e9W+xPVo9dslea0kSI0kcSeJJUhBJ0pMM/Cp28NM6
UCshJD98SVvw9A0utlY0DvTdbFYj26Lcx+NZkTyK6nJ7ZFtslnmfZJSeNe5L6d3JLqj0IngD7JTK9y
TePnkU1byj8igqvN/XGD7PKPxbRnkxW+A3kYzPv4Df6GZRuT7YGpUJiY/KiVofJQLOhyuReNsJS
3Qm3M2MckDKjyqia6Qz5bQXUmbUPhkyFyRGTKZORmUi+OQpktRyHJTjnRJqf7t0jX0xvG54Hfp

cbgAvM1QZ0AOee21i/TSC8I82I48AdLCtke2QUgl3+vJ2X+qMPO6juyEBNvWYvKbdfMQlZ5p6CJ
RrSdJxyhckHPdJkuSUmqYkjmKLwAWT03Qh8nTkacQm9FTE8FXD3k/aQxOcR00P2FbTiOmhadw
0aZo2zUbygDHXR4ZHRiEdGReZCPlk5NrINOCFR2ZG5kSuh/sipe1A6aFMHPAoUf1pRWKkl7Kls
D6yJDIOqL0AkqR9spztpodwDzhwZ70MmXQnn3pYYqj8HdYPHvb/tyi6SvNwLf7nfC6Ty2GX4LrLj5
vCpVMU9vpxE7hkiuVlftwoLo41wbXDj2vklug9y41+XMZp2Q64TIVxeTZN++woH2+hb8+e4Sa+m3
8LSvwN3wOR7Uf8j2Bn3cv3Qs0+vg9sc4G/wHRgm58yPX8ZLGTgf85fh/hzg/+AhEf8h+yZfxt/jaL4
O/wd9hy/h5/D2R+xH8EMWcgZABizo9hV/4p2JX/BHwD9/bflfxtwt9/iv6Oij6qojtU9PdkGvrOJXLQX0
75TuiniRfLJcDVpB8vgsPW7/rx9Fw4XA358dDCHly0iseesDm4OunHmwSrc7AWqXljbJWlZxvHhu
Fq91+POk90+1+vOvkW3I+vCG/tUDiXWRXVGP9aTqj4bgyiskcxWSMxmW04vIZ1VD6IFWPqvjfJb
pERRerLP9tleW/s0DLZb6nqvs9IUyJ/obfqEk09sVMv+reC6TUm5SF0qC/dAZFjN+xMrJg2O0ZfVy/e
BP6GOAJ2xo6FTToTxgC0YaFhpjAt5bFwnRCWDCK0LBVwRpgF+HmQtMC3hW3GZzSQdsI5LNV
TJxOU04algpw9IKMMcizD5LsZABVh2+ieVBthG6XksB2Ad9C3bJV9w/OeZ8K5HdTDCug3W34JY
EgF1wBuAtyR6XsAoxKYIGaYNqlgC0ABgBOgWMqXPwIAqy1/II1H2NhWY5mxcTlvLFvetfzE8IPLz
0A6Yexc3gd8SMvPGzuJaoVSg1Bm0KgFKIOSmM4vH1o+ROUGpSTV8km8RhKvyfl2kyyUtCDnJI
z1GRtD8uFOxvJLxjyzZgBuMyY93vb8TzvavaAokUo/ZaYhSQvAqlyngFgkWmYvSE2+V4e2LMiJD
wkCnoUF5IYsjYkDVJmSI6x0ViBCehMytOgVByk8JD1ISJdQ4I8B8rifVFKcq0FiQ61PJQIS1LkJlaE
Q8lwIEW2bg3JD9kOeYWx9ROeTz6R5y47EwB9AHCKXjYIAF69DLx52TX53ugSAN667LFMg9cv
uyNBxCHGMS6wrpb+3eXwzbr+8JHwcUiTkMaX1YZPAX9S+KzuLuWTusvL+PCRZbxuCOA/IztcFr
UsfFk43I/GS0mqFSBxfFm4LO+uJEklZxyupgEm4a5nmV7n1R3SeQD367x/cM/F7+POqHYAeN7R
zpX/9qGSnrFiYHmORg9j8NB8jhKTg1q17UCPanFsR3VthJ3I1w0yTtMYfBci84TuBLYfdJNxxXe1c
ErWxCe2pAeNMU4Xr9kCnAfag+AjxcEM687jCjeKGEpA/OfyJTxxjndpRRiV6N/ewTKINY3I5wfp7g
xiaBuw5qvEn0CsK5vrBn7NPKzmQQWlubT5Utwp6K4j1uNZcFaXS5wyxNoLRJ8gnlv7AsKTONwa
OqnvolorcZ3S6QnH093rRONqxYInEWt/THQN0eF0N5c4UcQ5RRypFrQ1f40sxrB1NoM6sxmyOZs
LJZxFGPd3DPs+H4ky535Jkg2anyBfcx56fR7I8G2EP9RCW/xPCU8QvoP8oJVEdyAOukj0daJ/TjiV
OC9q3gMsEv5jwm8Q/w2kg75BnBiiE4n2EM6IMmsJb0DOfeUQDssEQ780RzS9OGoaGGXNm0R
vQX7QGN39jeaXgPuQ5tKlv43orYiDzxCOkiUg7iP8Oar1OZLwt4QbEWtvER1HeDVJKwjqrBAm8
CLgsBWQR8G92NfkMNvCL4K9EPNC4D/C5bk0jS4o/wM4iAL0WuxvNZEcn4e9LckrQDwe3iXP6B
ZhT1FD+f+VZMN9I+p7ncRB+8nOpHwccTackYr/SuS0Eoyx+nuN2ROAfo2SuMnZH3Qw99GHHQ
DsYY4fBlxxqjdSKTZfeJcJdyKeP4a5wB8mfAAh/6Dz87xf6lk4MziLqH8IPDqoJfxZMLd47G/lxDzD2E
mcXwG0Sel/rOgzVAylOgJwr9CDv8W4evlyfeJ/4TomaBExEgHfyb4g4Q7NHFIeZ5NnGwqc4ToMs
lewiLhFzmlbPw20uRFwrmkp5lo/AoY9EXTh5jo+xlHWw/aQ2U2EHYSf5LqThFeRXiO7v4K8fxx9BD
wz33kk+doXHCMqlDboD1EdyOt+Qne1VyjMt+lvrxF9Bjx41AC0CinUuacI5nl+WukuVVUPhRx8H66
66BaF6nMm8jR/i/if5XwOqktqjtH2E5ahSJm90nOe6RhLeFW0uprms8BrSNfigx+Dcr8EdV6WeoR4c
0kgTERjiPuxoOiEYP1GfkGQ22J8yHhNuInEv2Q8CBhOtmCHRGXEBYJO+eKcLyI3iBxCL9BkiOJpl
MA9yPCdjoR/JroNKIbIMFcxmd90O7X0Uq4RnGx9H3LiTmc1730X3zuUszvov/gr0EMq8MKqou1zk
sxlmq9xTLR2+nufyaOhrHm88R5maS9A2c3TsORJnup1yMk5yZx6HSutch8xGMOh2QSjVjTQnQV
4b2av4RWvqR5AGUc/Mtl8wU0K5sIPyTL3KfZhz4ZBpbmeLQo6E+4izw2TfNrKP9Q8xFwfoAylcKg
ZCfRY4i5KeIMEqeVcAFiTRzx1xLnPOGfE/4G4uAUKvMm0VFE9xFdQzlvEWcLle8iXE78NM0p8kb
EsYTbCRcT3oWYj6eS7dS6HjGblQ5J4zNkOYiHCV8gfgfhMsJNx9GPJ+DMtkQySHd2F3CZwhP
EmaEOwkfJryP6u4ierdchp7Blg7Klvk0RtwpavE69fEmWWCTVH7+LuEHtHt5B+0w34e9IzyBGPgY
i84hhl0Hcsbo7iBhkfhThO8h1jRSmQLCiYRDib+Fyr9F9AOSOUS1YgnX0902otupVjmVmQ32ACcT
MfsXwlOE3yN+MOEQ4tzX4nOS29rVgMe1tN/Q5BOMdy41nyF6Fe1PXiT8KdrJvE74MljyAslhiHkn

7nm4S7ivY/dpj3SS8H3aHZ2UMLblrw2mKBE8RHuGdsS4k+SKtWmEpxAT5xJxLhHnEnGmsC43
Rfwp4k9JJYN3U5IB4gySPILJduLQukZtOaldJ/HTSEI7crh2ktkuceS2gM/e1uB+rCEI/7rQQPQM7Y
RnJJ2RAXlosqEOgIIGOYDTqBVqi/Q5RXruiDoteAY5dHcH6fZWcCdhRhj1fAsIAKa7pNuc1C9sF3Y
LtDcmrYZwP8Nm2Rdx3NIWHF2PEkV1C2HpGQs+PTKwKow8829jhJEjMD6hGKLYFSIHSDf2i3A
a3Z3B/SfsTktpp3oe7yKHS5OwtMcmCWm4YwefxP12+9xNwt04ysiHtQDLTJHkYipTjDveYHqmFR
yFcgCXUiR002qCJaeolUtEHYn8iVo8RniKdJ4kfrWEqXw1tfULauUXpP996st9SRrusbliSRNJT7LSj
MSXy6Dklao7RHvyGdqNDyGGnobinP3taeC8P/9/aEx3oyYkLQ09hE1SXUbPrIIQz1+b/xngB/M3kJ
Y5v0EOOtAe4vOEuxDDqoRa0XNK0BZ9I4PoKeLTs0W+g9aXcuLXS70g/DZisHsUSUb8K8I3CRc
QrkMMK7iRYtTPCN+g1V/iXCP8GzxHlle7QLhdpiWtojBuE24naRtw5WJ/zrVTDPmlsGQH2D/P/+g
PyPO+xQ9zIFsSSNMvkqnS6bdSJE2GUeZ5sglmsXHcBaANcgzpTILlzlPwZ78oZ1KViOf30D0Rtl
H7WdP046n5L3LbjvnSM6l/jDZEmdvP/ppJ0DI79CvBP3mRA9jmEkRwwR5hjFlmMUzRCXESen8I
8IRxl+RXcHCbcRp5jwHHFYiR4megPheMLXEQfxRE9Qmf9LuJrwKsLthP+KynglZxD+kPh28r0ZG
q80CeMoA0av+yV54GXkQLxAn1IJPY0l+j5ZTEs2+QFhnnY1EUQXEZ1O5XcQ/ns5znTSPEXO/2P
vusOqOrb9nD1n9jkctlhARARFVKzgoQoWRCmilhallYpKU0RRCGCnNyqW2GMhxxhF7F2xRo29
xNg72LvR2MCSiA3ezG9vT7z35rY/3nvf+7735ctvr7NmzZrZa9b67dl7H46vgQsQz8FoZbiSegBloDe
wPq7OYyBXwFXYVcNIHMejdZiQy2RWSnTv9+FeuEi25vhEIM8ovPvWH0P+VICcDzwCvA0cLvZa
+rWQVVRTfCEjRr4LM74/KnlLxSwSE9oVmB3A7NJ9BXgFU7XGnL+6kuNwPOBK4DHhSzE3c778/
KK6ApYuEXNZzfQlgZwCjgagUeQH3y3xdyAdAVNfIG8mRgMHA091mp7EvUyATgSeA8IM66bCaw
B/ASclulPzicIM0CfG1UNTWBB4Gq54vAZWls9YnEu3UC35clNiv7XexMkAljgVuxyiXlin5Yd7yZ0SN
/9F8C84G1wfazlceh71HIB6HHuxE93ojR4cBA4F71vkNcRzgPD8c+n2CHr/J2Onb44iqzEZZ9Mle5
0ldrfacRnUH9pttu4FngFPW+RrSqb47oNHgeqnmOFc+voEEr368LPF06Fle0acJSyHoZuEUglQTq
bNT7HVxNxou48TkMF3eFGKWL6KsbARwvkN+JCf0ilZ2g7wn9Y8TEXz0v3E/NFXq6Bq2tgeB/egg
4DIH6WVhSRcgsD3q8v5S3qeslWtlugfp8+LyJUSLE9YuPzjVSpVLcS2p+gNqKi/2eHrGVEPMQ4E5
o3mnxFBonTX8esRIYCc1pYCrwKXYI6IM77FWky4heT+h/Uq+nAhneqfLqAj/D2yRgDnYdlzC6i9D
wVRP6UfBfCa19IY/CmS7B2eXijFwwyiRtlyJ6/YRzPKbmLfr6oe9l6OdglHT42a+NGCue7agZgmhfb
eLtnXQXHpYAX2LcppDHi7tLfv0SII00+8d0zErgCi3HxAx9hMaArGPlc6MP5jYBGi+1moRePxXeVg
m0uij0RqyFEe+MDbgrZ2fhAbwtY1bsZ/g3anMYixE5coZS878P4hMurvgCpRhxndXHaWuXjpUVo1
9ERTihCg5CdsMT2jdq7sEyEueVKp5HMexqJGQdVUfvDz8VENWjiFJPgXI09I7QRGvxF3IQ5PNo
PQicAlxQcwz3+DZaLQj9SfQqwrk7IP8Jlpym1inbx3EErxWdFC++mWy+W5zWHB3qyrQP0dgX
KAQAmoz4RmlUDDVOhV2VkgBerxlJgdFk+S5QCBRlXTVcgSNPo7Atle9KoMvADPbWC5GngEm
nOQ84GnoNkA3ISxLkNPgQ6YSQawCAgNewH7VsDF0BxD3xOYP+7a5H7A+tAvwnzewPIraCYB
MQf9SfRaAbwEzXngl3jAzOX96FsGPTT6zsASlPrqZ8AmHblfEBGgsKe5Ao0tcRYTBVqth+dsWIZ
BngN8CfSFN3WeoUBYsgJ1pRDzTKzIOeBirMUqyEAT0Aqrb8S6WyEHDE7whqiamqKVqPaQgzF
KF9hgznRkC0AKjiXUughGxZCDxupGPqrmP9bYB3Y74BNIOqb0HtDsx32OoFl3cVzibJPYngOI0U
gn++U8jtBaTXwc4HUSaAOKAVA8wnwJ+hVeRpkoEQE6oEEKN1E6yTgQ9jXgv4C8C40KZAvAu9
DEwP8DJoBwNHAKdBvB74CQiO9huwC7AzNPWafjJsOvS/QDnrY6N5DPgAMBWJcKRmt0cBUa
PoBB8FDPOQyIClg1QOOAKKXFA7cBqwGPzhf3SnEpwb8HIRcCLkVLBEZGgEcDnSE/hDQFaha
PhLIMLoeq6bHeumxdswZck1YrgR2BKrnjll4zEf9SxkjHUacg70QyEPHp4vwErQX4N+AeQi6KtCc1
WNnmBCzlo+YCofcJQPGMwHrOWDCvJBvfigpnxQQUJ2FkiBeiLQCCRAthetIYEX4KEN9KuBR6A
5BzkfeAqaDcBN8HkZegp0wlgZwClgNOwF7FsBF0NzDH1PYJ63oO8HrA/9lsznDSy/gmYSEHPQn
0SvFcBL0JwHPolHzFzej75l0EOj7wwwAaKvfgZs0iH7AREBCnuai1i1xFIMFGi1Hp6zYRkGeQ7wJd
AX3tR5hgJhyQoEmrAWVlhBI9bOCutoclIlmZqitZg9OoCPeagR0zYAqCCuZVCD9mwEHrYSMXQ

X8V83gLrwH4HbAlh34DeG5rtsBe7GM5alsc4iivvYrxD3lk3kgECda6QDcAuQBfYrAAulp05joMcg3e
RMeg1HZpJkJ2BjdB3LGwmCdTth+wGORSt1aApB7wg7tE4hogRDb+KOx11dOh34j1yl9zHvYFsC
3k9WhcDW6rvQ8X9ndQQc+grkNyCPgSYirG+ELI+Av63Yj6N1HnC80voAyG7CqQN4L+t+oYXchP
1LNBaH5oc2Jeqo6jvcKG/DPwCs1KgHwyNrxph7dzFVUaC3qzGVry5ll6ofYHHcEZ1MYqDKqurhta
76FVJfUeMVlu02uMat1Fdl7TmqRNS30Rj/hXV9YU+hpVh9SeLIWJYQS03hGaUGjHYl0B+AJwPzQ
X4X6xlhfhmSpB4pkQbw2Y45lCKuL1Uowd0BvZE637gEqEhxcDZWMFf4KGimi0Yq6n6rpzfnfB7K
CGzVULP7IAz4Oel7CAyB5YbcXZG9G0rj0BGpSlrRog54DsJtkLm+Sbe6X8Om6dylvoKjRs0Zsz2D
OLmAg8xavxh0xIYAg12QXy9xMp+CvtYnEUsPKzEDFdibpXU7wmgbxUVoUnWcl5obkDjq34TAD
NpiZlPh40zRkwWNkbkrQHfRpALoMH3EGTkjPEd5CKgXo0YZpiOuYWr+S/QiNkaED25COP6YJT
TQDfgWvRvV9VwWs1keSjiNhS5NBmRFHKO3IR7vq6tmnhmOJQp4vqof4pKF/e2TkLm+SDeQUc
DRwAD0KsaWm0RvSLhQVqt1bVZXEOfx4TM7jFLK6nsCGiV9kO8axen4+q/xKRH4l5H0UcbKB/i
2gQoDvOHVWs/xawixCBUwL1jpiDCFi+gR7npbcG+kMzSGU89oLLT6B5AHwGmwjUxQOBfFZD
MXoQsjclC+BowPdA2FCtyoai11DYCNkRUZ0E3CnseX4ORV+BSUB3gXQRqvgB8BQDhwnNbAp8
CdwrU14bNdcjWAuUIDBki0PADYolvorBPMleT8D+lqfPERJhad2L0MLRuhc/XkF8jnmeAEuKwFvqj
OAsnyO+YWr/CW33M7TQ8zllcg3hWE6j3xzy7ovU8euUhVn1UbtfmGQQUmjaqf+TMXdWzFj0x1hj
ITeHtHVbqCWQDcq+hGNHwNc7rKsbNgs8xGGU3xr0O3AD9PGB9rGAT5NhxypVg/0aVYXNN9Q
OcCUtEiWVDXoBcRbXyeNph3YUGMZT9oJ8CeR3kTPhPgmmwCHkDrZ+jbFTH3Bt7GOc7H2TIBU
x94DdgGVR8EWQfZBp6vwj4Z+B4e9iLPr8B/Pux/h34O7MMEGsbBw2vlqeocMLdVsJkBzWPldrD5
Gq2PIOOKw3bD5yJWFzlcF1eiKKxXXWRsXWR4XdTaTD6WFUbE9VGOhhyqciDGOonZ7gE+hv8
8nNFPqqz6Ae4VqPdHfU0CpmrZHoQVEVU8UvQ1dROy1UwhG32AEkbMEGjlgdrBt7bYVciL4aET
GMAR8iqNDQTqtGznaMqEPThW3xutt4EyU7MrCBUh5PbQt8EoXkKWkV0y1pQlInubsePcJp6d47
3ulRp39OKtTbF4xq6/qP/E7cNRIwFC1ocAS4B+AnU9sRZfi76TBEfdkpwr36kXjy3vIbVCUOFIhv1
z7Qs4jz87ox4cl52sGw0UMjnNNwCjYr9gH7APcAxwFTgPKCHekcA+x1l56GZKRBvTw6K7wxlOc
Kz9DnwMjASOAqYAHwP/A7P838RqNsPjZNAag/5LjAGeBq4EdgUNrMgJwN7AH3h5y/vKM9ANa
MQoxar/9+LtwC1o1kO+Cm/hqr3QSC2hPwtvD4E9ga+BS0Sr7ow6fyHTV/C2CrgZmlnWEPEkXAp
Br3ZAW6AVRoyFTSo02zCuUaC+AlrXqHo1eniifgbejkFjr40uNPbAjUCK0cuAhbAZrJ31CvFMQKD
UEJ6jgQvQywl4F6NY4R3uFGHGQfMQmt2QXTG34dqcl4sdEfw/BbZE63zIkyB3gaUrvMENHYvW
C9A7wOYB0ACbnWilGGS5HfO9VEzBPrD0FeCfTIVRutNLYvGiPMSyFsF+qJXhjoihAPn/Hw8B
Nao3GOhcDX7+9glRFPPI0v0GzOiH2j6MuQM8wk9AzwfJ3Fxp9BfT6Dm/MHwJf4o3tLbyfvQz0Bd
4tmyEiD3kwLI+jbyrGWgl8jdYQtJZCbgqkwBj1nS96JavRAPbAWTzCOUqlzwpYbgRGaHer66vKQD
PmeQJPks2lUmN4CAS6iFYaCMsc+D8Bz7mYWzT0pyGHQk4GfoGYF6nfH1ArF+fVEaPnAhPge
Rbsb6hnCsuGkPerkQH6wdtcgYYSIPzL0MgLS CJ5wqc8GuvyFb5BvQzeqgjZMEplG4zr9l58OyhloJ
SrMgbW+jDeiRgRAfAYzUFt5oi385zNaoOjaoOFaqM6wAaw8YXcHN68wCHISQl5LgWpo0BfrDIP2
CNcZRv0fQ18CdwPXKAYQ2kDkUXQvlcmBvIN9MVZ667Af5rKeljnNcStJ+Q8RA/Mr+uLbzbvUhmW
c/i6P0mB9JMd9uKuKw/O3fXhq0ZAQ7S23ieTqVhEWnxGfQFwSh2WkkujkjN79Sc++vRMySL/U+K
yBZKjw+0mXEBdSnXASEv8eHLEi1qQisSXlxCeumXlxF04KKU8qETiWz+L96aihVgknfjmviZLRCZ
U+I2lDncRv9uBdr3WxkgFUjKxcUA6GQUcD5wCnA3MBa5lSk1JJhv7pAyMJ9uAu1MGpmSRg8Cj
KZlpqeQ0slAbxpOrwNupaYmp5AHw6YDeSSnkJfBNBm/WESDe9xK9BSkk8c0BMTv5rzR/SDqC9
2c8On9guY/Q+BEqHyG+q6H5MX2E1hpWJLVJI+JDmpMQEkGiSSxJlqkki4zAX5PnkHlkCZHFS3
wyQZ2zrpJ6xK/+86NR/P6v+DXm2tohx4i/EtSZlgn+Wsk0GfPVmU5qx6vqsYLox4+2G9Wj3UVVX9
ms+nE4yMfi/h3Oa58faGchfjEF7/3xCxgSn3UH8XcEhqb49D/820Wsn8gonavkQ8P0McSJNCWtST
vShXQjCaQfySDDSTaP3HQyh+SRFSSfbCW7yU/kJCkg18ld8pi8JO/4hksxbCXUsMaw1vADjusM2

3Bcb9iOY75hBz+u5dKPOK417MRxnWEXjusNu3HMN+whEj/u5Z/Wcet9OK417MdxneEAjusNB3H
MNxzi1usMP/FP67n1YRzXGn7GcZ3hCI7rDUdxzDcc49brDcf5p3xufQLHtYaTOK4znMJRPOMTx3
zDGW6d/zcREb9iPZSM+rcichZnvsZwTovMeS0yF7TIFGiRKeTjrDFc1OJzSYvLZS0uV7S4XNUick
2LyHUtlje0iNzUInLEbmtReSOFpG7WkTuaRH5RYvlfUTkgRaRX7WIPNQI8kiLyGMtlk/+RURmk1
yyjKz7hxF5qkWkSltsRaRZ1pEnmsReYGlVnQI8puWMb9rkXmIRaZEi8xrZMwbLT5vtfi80+LyXotLq
RaRMjUinGgQEaNOjYhRUiNipCliRr0aESNTI2KU1YgYDWpEjEY1lkar/yAiB8lxcP5c5RF5SJ6TNz
pJZzKa1lgYrdWIGBU1IsZyakSMNmpEjOVFRlwV1lgYK6oRMVZSI2K0VSNitFMjYqwsImK0VyNir
KJGxOigZoyxqhoZo6MaGWM1kTFGJzU+RmctPtW1+NTQ4IJHnKnRRYtLTS0urlpcamlxqa3G5T+
OyGNLRNy0iNTVIIJPi0h9LSINtlg0REQaaRFx1yLioUWksRYRsxYRT0TES4ultxYRHy0ivlpE/LSIN
EFE/LWIBGgRaapFpJmWMc21yLRAXgRqkWmpRSZli0wrNTLidXjFvHEFmsmvBAoZKF6W8auBE
3EjZh6vEBJJYpRznOmDjZ31M5XzmjRLuQCpC9cVaNIspZBLobC7qEmzIEuQhN1ITZqF3+KoTTy
IP1+PCNKVxHFWzyJfkgNKFctINy0jXbOMdNUy0g3LSNctI92yjHT7w0jKly61MQZz3WNNmqU8gR
TKdU816Z/N6I5IRnctM7pnmdEvlhndt8zogWVGv1pm9NAYoyLLjlotM3pmmDFzy4x47es8dB58A+
MoOfL9YC2pFq7FfOdWzge7gCwifll/qvV4rsf2oZi0itl4RapRUVqZ5HaQ2L4vTQHvlesjZ7P0esFery
E9W+w/F1ki/Sc9xDZkkOq/n2syFy+r1IHtpGzvH5KeOUoOnudi66BzkcXqAvXie/Y6q33c1/fQTpgkQ
5+kKQTXJoD6aRFomWRTlukM5DErISRzgpZusNxNtrOWazOW6QLkCiPng2xkwrQQ8xkqiRm8Q
1sCj+ysZfEnGZLhwjllOlixZPlyzSZYt0xSjdtUjXLNJ1i3TDIt2EZOD7ZgfiwlFpg/iR5hLfG0jz+XhHMO
p86TC3mi/xnYKUyz8fhTZx+plrc6VbFI+3tVgYpGnSdJ4vedlybrlCWkNM0jppHSkv5UsbSAVpk7SZ
VJK2Sjv4jp9iZ2zHs0b84ofY91XQfn1vIW9YLa3mPjdzeyrtknbxvSLPPCkHf1UsfltN5CG/6oh/T5vvf
DnPSnOlucRZmifNI9W5jz2kBv5KuCX+Sjglv5JG5a/k8ZK4W6AUw1MT5fcTVKEK/HEL+qvsTExm
6+Qack0xQ10sWU0f0hq0Hm1IPagX9aPZdCwdRyfQSXQa/Zrm0G/odzSXLqLL6Eq6mq6l6+kGuo
Vup7voPnqlHqUn6RI6gV6i1+gteo/7ekyf0GL6nNVjjVgL1pK1YsEshIWxtqwdi2RdWFFWjfvkCSyZ9
WdpLJMNyV+wL9kols3GsvFsApvEprBpbDqbyXLYbDaHzWXzWC7LY0vYCraG5bPN7Ae2g/3l9r
AD7DA7xk6xM+w8u8iusBvsDnvAHRNi9pKVslsTKayQbaWy8sVZVu5iuwoV+fn7SLXIF3l2rKbXE
9ulDeSPWSz7C37yv5yM7ml3EoOlmPIOLm3nGm90Xqz9VZFUmTFpNgolRR7xVGpodRS3JR6S
gOIkeKp+CoBSgullRKqtFU6KJ2UaCVGiVXiICQIIUd5OTWKb8Dx6Nbg61CX1iUSj3JDvg7u1J3zgy
f1Jlz6UI8i09F0NDHQMXXMMfLojyNW9Cv6FTHRiXQisaZT6VSi8NX4mpSjs/gK2vBV+YaU5yvvzH
aA59P5pCJdSBeSSnQpXUps+UqtJHZ8tVaTynzf1hJ7vmrrSRW+chulA1+9LaQqX8HtxJGv4i5Sja
/kPuLEV/MQcaZH6BFSnZ6gJ0gNvrJniAtf3QukJI/hS8SVr/I1Uouv9C3OZvfoPVKH/kp/JW70EX1E6
vKVf0Lq0SJaROrTZ/QZacCzoB5pyDOhEWnEmrPmxJ0FskDiwYJYEGnMWRPWxMyzi4R48gwJI1
4snlUTb54p7YgPz5ZI4sszpgvx41nTITThmdON+PPs6UkCeAYIkKasD+tDmrF+/l6mORvIBpiWLINI
kEA2mA0mLdlwNpwE8ez6krTiGTaKtOZZIk2CeaanJSE828aTUJ5xE0gYz7pJpA3PvCkknGffNNK
WZ+B00o5n4UzSnmdiDong2TibdOAZOYdE8qycSzryzJxHOvHszCWdeYbmks48S5eQKJ6pK0g0
z9Y15BOesfmkK8/azeRTtpVtJTEie8lnPH/3kO48hw+QWJ7Hh0kPnsvHSE+ez6dIL57TZ0gcO8fOk
XhWyApJAs/vKySR5/gNksTz/A7pze6z+6QPe8QekWRWxlpIX/aCvSAp7BV7Rfrx/H9L+rMyVkJZSe
R1QMoDXgoEM5PVgTdJ4TZQn6bwuKpLPeW3YkgxeH1VlplXvrkqyZGfZmQziteJKBvNKqU2G82
pxl1/wiqlHRvCqaUD+lou/1PiSV48HGckryExGyV6yFxkt+8g+JJtXkz8ZlzeVm5KxcqAcSMbJ4k39e
Lm13Jp8xSsslkzgVRZHJspJchKZJGfIGWSy9QbrDWSK9SbrTWSq9RbrLWQarZ6JfM0rUCbTeR
WayAxeiTzkJq/GSmQWr0h7ksOr0pF8o1RXqpPZiqviSr7IFepG5vAqrUe+45XagMzl1dqlfK+YFTO
Zp/goPmS+4q/4k1yludKcLFCCICCSp4QoIWShEq6Ek0VKhBJBFvOK7kSW8KqOJkt5ZceQZby6Y
8lyXuFxZAWv8iSyUknltb6KV/tjkklr0vrUTH3oCzqZzqDf0u/pArqYLqeb6A/0R7oHjHmcnqbn6UV6hd
6kd+h9zpePWX36gtVnDelkFsE6sWgWw2JZHEtifVkqS2dZbCgbwRaxZWwVW8c28lZazhgy3Ww/

+4kdZSfpeX4sYJfZNxAL3WMP2VP2nP3O3rBSWZJl2SSXo/dZhFyZusrV5FTZj0VzqaecI CezW9b
bFL1iVBSlgmKnOChOiotSW/FQvJUmsjOlP RKstFHaKx2VLkpXpZvSU0IQkpWB/FwzwGkEnKYD
m0lgMwo204O1GPhKBIMZwFRGMJUVMmoEprlGlylgpHJgJBswUnkwUgUwUkUwUiUwki0YyQ6
MVBmMZA9GqgJGcgAjVQUjOYKRqoGLnMBFzuCi6uCiGuAZF/BMTfCMK3imFnimNnimDnjGDTx
TFzxTDzxTHzzTADzTEDzTCDzjDgbwAAM0BgOYwQCeYAAvMIA3GMAHDOALBmgCBvAHAWs
AAZqCAZqBAZqDAVqAAQLBAC3BAEFggFZggNZggGAwQAgyIBQMEAYGaAMGCACdTAUDtA
MDtAcDRIABOoABIsEAHcEAnXjt1yCdUctdUMVRqOJoVO4nqNyuqNxPUbkxqNbPUK3dUK3dUa
2xqNYeqNaeqNZeQNY4VGs8qjUBtZml2kxCbfZGbfZBbSajNvuiNINQm/1Qm/1Rm6mozQGoZYGoz
zTTUZjPq83PUZgbfTcv4F7/iyCKyht+D7iVH+R3odfKA33++w90YduekAd+t8V0i/Y1nczZ9xXEsfc1x
An3LcZo8gUishTyUY0t5OMdW8giOwX/i4Xd4KIGHN/DwDh4mwsMwePgCHv4CD3yXKX8pLCCN
tEijLNJoi5RtkcZYpLEWadwHSXlHkV5C4nt5eoPeJIS9Z6VEknUy34/KTJaJLFvJJmJU+ijJ+LuydrhT
dSM+2P9XsD7OK5r3pA8/SPj7Yr31Cf7pBd8lXoOdDR3J65+3qUf6EDtRsXMh2lPoeM8b2ll+YMrG
1PufMuUxeoqeo4WcKW+AKXIFa0zZ4N9mym2sAdvF9rFD7Ag7Qc/x4wV2SWPKX9KT9oz9xl6z9z
wajEfhA1PW5EzZH0xZE0zZhZPID3/KIF6Kn9JUCVRaK2FKOyXyl6bs8/9M+f9M+X+NKfGewoi73v
v8jnSNeA4i5al3j+SCdXlrm797eyH4QryfciWNOPcFac8MjuF+9rjl3v+u+LU8SPcs0i8fJHmlsP6n98d
4a4O3cgreNnEak4ppNX2yvvq8+RXt7p1OtCHE8zl920BLHPEzSxx9lqwbjw8e/KqczSHnZjmu4aoWk
03lam61k1tCGSo6MmONIU0NZp9dln5F0+rwoc2dzo480Touqj3lizfFR5JAMkkaSSW9SRb/P1D8Z
675kTO9ndv0Nduijufe8Pt83Hb2y959odKgC3nZlbpN2foD5my6Oo9KOkmy9eZTnFL2/J5Ddfuarpj
wFHM5y2x1jM9rCKZJP9HLttlnUZ625orig9HW9GI8Zt+UgclZaQM9K5hthNJga+jSO2IA2sAkz+pmJ
6Ex2VbukJKYkZaZ1ifLJTgtlz0tlz4rhfeooaa4h2qmtwx/t0SkDertHZcUPSHfpFNzKXL1KOU9fT7O5ia
e/p4/Zuxv/2MTsafloHr3pv2Vm5czWot3aVt+hY6cunnXNddSP1QcGp6T37Z3hEhIV6hlaFdnU19+ni
XtYEz9Pd58wPrE65lrqGTn96RIF9c4YnJLY25ytc/04wjpGaLauPOF6k5St05FbaZMrDXnaJ+0n15e
rol/vLqnfr4cd2VL+6ePf+8c67QsdkZO7KnPu2/Pyy8prXpntX5wcYfCx7G/hY+gPWfP+Llm68nDCyt
OP1Jz0g/UPsbl3IPM02/v3f51YPM04707+M9NmnJq7hGlwwlh5v32z/sbjlQ9dyDR0+bT/S8d7k6c
ei7ly62a/9hlctqvVLleV/m9a8Rje2ftLwVulJrytUXSn0fuZRe69JsQ3POhkLFj5Patx8/F1MmOddbMA
T/+loTGFkTHGNFOIFXPbJcS8W5hWcPHOzVUZsxsB6y60KvRhOPXzmypmuj//fzs132HKzFL4w
7l9EtaUhR69PWmqE6bh3mkh7S7WZ+kjam5Uhl/Srs4W2fF18LMzjyKzjZ6e73dL21Xpu+z2p959/nV
9b0PH3WplrA8CTnkXEvvYLYfZVfLp+RSI7B005Ogt4PfbmqYf8B3U3lztDCooe9gbm9um9cmL3R8
cN+srPSmjRsnZqR6DPiwTh6JaQMap/dPEdrG6RlpSYMSszlBW5ZRrCIWkWelBzcxx8hGXpiMGX
Q6fYS5nTn8w2ezNL65NsCQIUP+blDeGf/Ec5bZVsy3jl4xmz64pMa/KUgqsmRyxXFWPtvGjk5rd61
X7pVLbxvHL27d/+Jf2rZ9s7vHjqF7XWPaGfd6zyl7by+seaSldPRw+jdvpieXb6xEZI+dutzgZmtVmlbt
7rPgRbMCRhbcje5wlnvFoYnXbS6Mfj0n5ohSb7N3//lBe1IT9te9T3a8Se19qmfi/JO/T6/b5NahwPND
baKWlq8aEtFj3sFHO5ZtSoyzvRXe/ED3GjanRnxdPnxw5vu0mDdTsj59s42+OfsnfQq37R4TXX9g
TZ9aphWS8/S8sZ4/+YRZ+3T5dW9uW+61/S+Zzrm4FaY6f0g9EFhxM6JdgeCwrauNvYOa7e6v82
skJiA7BI2qiDC9laLfu2Wel2Lb70ye3v5Vxm5jT3mNHb+DxrTmRoNW1c47l4qLBP/0tjQ37byGLWu
aaatE7fye1NslKiV5IPf6B5GZ/cw+nk18vHx8BZH5mQMSh82jx/xPEJlmTv+B+b8kppnv0vx+vEW3
1ysIO7MofsfSsLeJ9oEer9ucPfz4yeHvNrh1HrTr8vHysl3FJaIvc/f26hA98W6HThennlgYv3SI7Vyn5
U/KZb1a3nXYg3qvoss5uGJ5487ecb394fKINSWqLF3UmbNppOqRfPm3EuPDBTvFhq6oeHJ4wed9
+31VvY9IOJVrPamseXe2LGyNHdNzcZkCvoU7rt76abdulaPuZDgH3Mq+HRza3W/VtuYDjkzvfij3Tr
Hha8kNz3MoO3eYH77pSa8fe8oXtK8yf16G486Kxq+7OW96iYHGRRySF8xZsNkUvn2LTfU2z3gvyc
H17QvdS/cGJFd7YjWOroSmbXWfsX34Fj+652cfCvV+q0qMKGP+IKY5HJPbPCpV+xFYTS02eD
91O/DLg8lwZ23OmLHKlB+Jj7iyaK+o5XywJM4f87fp4mz3FR2bbwNvTN8C3odnH3CTA29fs7unXJ

97dJ9HP7J7gl9DHPsDJK8EzMcnsG+Dv81cEeKzig6NnN9vH6l408fC2t98eMddUw9xVJcCOZk6B
eZwCx4f+RwTlc5lnMk/iXmZ/d29Pdy9+AQYFdvuIAiPNnAQ/osDAf48C/4HvrD/ju2UrombcaKYrje8h
d3vUp1i5+OrS2HOks02FM0vOVKI3f4q3f8OLwT/RyYMeec98ufJ28nvpYLnyODQHtXa3rnZ0b74y6
+LJ1Q8mr12ydtIK3v99m3cz18c3DPi+5QnNbL3PTsxbWj7hN8KyzkVRIUqmN2lyHdX1Wl5QbMW
mpa628/bH5ZlfHTl5cWlbf2iK1X8hG76wv5tm9K3fd/tDY293bLSMO9FRdmHrgdVNTytfMj0/Wes1cr
T3+aOzqXd37W74+jB1nRq03jKm2EXq7uUsLf1+zvYvcnQ/2C9fG7S44o9Ooa2nVrf0f3t6a1WUub18
cm7aH9z+MNP3wadFtx7ZH3A4LG9qdip+7O2tIRNzFo83Z7OdnO+WqHxnivd2c8RuzfNvaa4X2M
NkNdNt4qznjZJ0Ve0pD7xnVXOVv1JaWdbF093cUOWF2n/wQpe0NE4OfKFS+qQkxmf1dmk1KKt
vWkZK1jCQmdns7+3p5eUZ4O3FycxL++glPv5v8uy/YrCNGZ/FVjUn7XGeG+fi0vq7wVGpgdUK0o
4fe/awf+m39hVu3miaNcbxh8Z5Xo/Lru9vHVnrQga54vupaeLRdS5tXxb3XdOh/dSlu4a1//z7NobL7+
vcmD9owqlVmSEjC0dfefHrud+SI7GhV9evbXGzXt9vHZcvzcs+qxKzt33vjZeQWDe1UfEjpmnL/9
6czubEdyl6lLN6Y0vlzVunRmVv3bgxtHX7Mzf1ZydmrC+2NHeoV5dtpe1/ZukPIURv0K9VwPN4lske
fVYvqJhf7yuNjlrt1GjCvH9oXdky8f9Y94Vloi/trjOT3sIW5Z7pPcYt6MHxVu+dhp5o098/dPCR2aZXc
qccqft21+b41Vr3ouQ8M1pNHpJu5vGAGW52uTM/MIB8+Yq8/ZRJBVs7l9XqegePNIWQR7daks7P
4JhvfY06SXh5f8Yz8pzbP9uzYlrsIzbVnznRfdzVUtRnaSXqlulIFkEL+dCSat/orLbNZkxwV1rfvtvTq
27xrcMkV989ndJeZOKpe1Nbcxh+YF57Ua3/Lf5zJLcwZPbUFBYlHoj1gs3MxJ+SMW8/9PNnKiYIJ
Vr3/PX5KOfBYQONltbP2jtKANXlv6PbJpPHBF21ePeg16EtHMvTB4rXXpsV/dPRfXOj6i05xRNXus
adE4YseiFV3n3Un/cdvmkmFb2ma8CnzYauTRW0qVIGNL57m4v7HudLDrCfc77c7uTL+/otwiurTrz
W2T2n/6/JvW8569KHp6Z3wNn+bbus4tjqo1rsGSbKdZt3MMzs9vR5ZMWXj0ge3SGZE/Vzv7dcY3
DT4f8L1jiVNxVEHycdeyWOcTi6bsqrtXWGLXkEWdT7z+dXFM12vfS6Eh/1WdmYdD2a9xfGYYy4y
mYTIkYqzZxjOGpJpTUbbSjSpSkV2YJGYoy4QkhiP7PIQosgxvMuG0eBtyULzxVpODsa9JNlh36px
Ozns6yz/n6jr/Pff9u37X81y/3/f+PPEidWrx1d1eqnbA2o10FGfKZ6y8WKOZpY5EeNCyX38oXhFTFv
bYk/YuTNaM+XzQfrybnCHp3KaLPsW+LmNK02yu1DksPYsUlwK5sHWdMJ1ZT4VnYxGJVv4IICX
hoqpJXtDz92fbH06TShxSHS6lJRXtMOFzXO4q8YIF39w9o6klwRoN0hNbDKzZ70Xl2tYm4dEeOx
EJbORb98XATqPeHokJyhP+up5VjQHZhllK2CpK5WAlhztYHmnEFDxt7HH6oGW1wbTIDCOE0g/
TEfaXjsLJDiG17BH66ogxstl9a8Majb3YAsWEDaUfUvF5fD05vS2pPxdzd4tz3nX3TjvyyK+mswQP
5BMRuUCOnwJfVnxfnyXb5kxTivnzfA5Qh8ows34eWd8W4PkCilo6WEJoQpy0HfDJzjCFmGrNOz
Fnr5mABQBQR5/J77ym+0t84Xfkv/CH4DerzcgkdsXTyg/5nfuC8mHvhs/rhy/z/Ru5B+tmbgtUmq2kU
/7PbBpqHh1mwbBevKTrakpeLW2eelz80rgwE50SnBX4jp4qZpOwxS72Y5A8qvQH7j4U3TVwW3Li
P4s+avdsg+wytEY9Y9JLWWAsfi5eZHLMSOT9UsGtPWjnSJdx9sq72oC/mHvr7HWvvl1vjOyq47p
HdhIhVSrirl7ZinD4NFZ9U1KAgCvTwD5KxEvMxnjmMyljy9Q74Xu2fnb1h1JKTQBmRI7iqqoepZlc
noEos2KuTGlosbbhKmFMTPHyOvgHBlroVgQEjCauFdWwYj5RJNYWLWtAgX2pE7sO/ydborpF5
mS83acm4tuFP+KHGDC338SA7+ld53eCdS+u/o/d1C+B/ojdxMb54HBERn/RW+0SIAdNL38Us/c8
P1fy5PKpJSiaabFd2sND9/fFEQhfX4v6H+f1W6884amZnw2Jnv8G72RF1l6OtOio0FuAYbfM7JXw
R1p7M5PLkB2ytWnOjv1uAAeWYph7LOZocdHHJgVh3PkR6UAcdVMMkL17qn94Fnh5qTYVBWk
snQvJ042+pOKmcsyfeXqlejaQsCWrf8E39UU5QnrS6tccjZ2C3LgkOkB5KW+TQ/WFB6A10/z0uz
1QYx6eZ8AJ11Te7AkKCUNrcDZxaCl6gHwVmTJMJGLAw18AjmSpvva5CYsrwW2aqrfrKkZerBJb
hBeK9dEGYWaGeSPZydwBKwbYgXr7Zlfdh/3/M4Q1NnrjBsb12FjP55PSjtboW/eu0RpuS0Z5qY6V5
yrqiMQKuXWRtjpL0udhz/VYHYZMka405fqh2+UBes2WLaeUxBTDohvt00852hkuO0Bg1Ft4cUqN
NilomCiCsQBz3EDsZNSrAJ5TLfhhPoEc9GkQ6O3XzvKXFnNRPGU46T93K232fntewObolWCBU
RnQzAtudSHKsSfanwJV+khrnUBdNStltvG82KBnxK0z9auD9iwEhXaPJvyZa6luUMlmlUnkhs4mJ
H66vYzdWQitPcQ1roirfom+Q6jKOOC1K+pV1AX5LW0y4QCipwSIVqK5mLaMS+ndlq15cya/mUZ7

BF4FX6J5cMaDZgszezEqW4gWp2c+y120PtXtAoOYI+h/dpQJZ9wVH5eCPOXQsBggBduPy5f/n7
b5FsTuSj6yed07W/6FebDiWzuUPM+4JsFxyGAzavin5PBrxv5cTwoHQm1rVhXkrcdlqxS9AlaoQlbs
58C7pu2iODsAWKRWtQuKAXIB3QGFAQK/NLk9gQFg+RARBAFROJZXjy/K+/JG0ShK0cp/stgJV
JlgV5BriRvitbvfir8VDBIXq6vVvpPgy9C70vVG+0OGGEIcQPWYE/0vLtyDzvZ9UeE5gRoHF3fsJGN
Jbpl+Djylb8KafeeTXZZNyTdLfv0Rhoqti9Vs/Wla5vvUUqG/1hT+EzboalW6wZ4Y6jQPWSon9n52hn
ak5Ed7viNHb32ToP1WmmM7SngAAw7ktAG14XoS//a1XPxDwWqdbeZSCVo7evLJLJS6DDnY3J/
66xVj5qAd7ljQ1Oo4KI33+gzqnsOh6MhG59cLxp08O0AIY7A6ohRWMedUVsK+wmvVsZvvmaMy
OINcJg7ccJ/kQbbLaeDXRIOeYXQS4xRd/G73QjNlxWPaL2qy99s2ddC5oh3PaBTlbsAKkTx2x0J4
KgQcZ5L9lsqaT8sC/j+RGKTJl0Ayc2ShH+brlB5L//7ChS39cv0QBuno62N36O9x/GfInfTUvuhtRo
p9zlyxye6NtuM12u8Dtef9YKtciRG92hL+J33cFK4VF4vHEpsO8jG2qYN8gu2xfzzlwJ3Z9+4Pjhjole
SdO5rU0XTk28y80Qt3gKK+mstmKmhICyVsdz8miCtJ5ut/4suQt7fKA+51IFuRGNXts5RtzGMhnR
RkZ31DtX/SSnZXxggwrLRXWJiC/3ky8djD+3XTyyK/fTno6Zn8NIEq7UaJ0C/N5MvdA39jcNBHpry0
lqkVnwwWsZxMHEvtWf2b6jJCT65odSNclxLDcb/ec5wXzpd8ZvxpT/giaBTZTlzkjQ/VyWgZ5eUFJn
QSNz9Q1jBu9b5Tku/eYjsGK9OmykEKXhWauhC4kB2jVFL03KblrCfQbobk6Tg0KZW5kc3RyZWft
DQplbmRvYmoNCjlyOCAwIG9iag0KWYAwWyA3NTBdICAzWyAyNzhdlCA2WyA1NTZdlCAxOFs
gMjc4XSAGMjlbIDMzM10glDM3WyA3MjJdICAzOVsgNzlyXSAGNDFbIDYxMSA3NzhdlCA1MVsg
NjY3IDc3OCA3MjlgNjY3XSAGNjhbIDU1Ni0glDcwWyA1NTYgNjExIDU1Ni0glDc0WyA2MTEgNjEx
IDI3OF0glDc4WyA1NTYgMjc4IDg4OSA2MTEgNjExIDYxMV0glDg1WyAzODkgNTU2IDMzMjY2MTEgNTU2XSAGOTFbIDU1NiA1NTZdlCAxODJbIDl3OF0gXSANCmVuZG9iag0KMjI5IDAgb2JqD
QpbIDl3OCAwIDAgNTU2IDAgMCAwIDAgMCAwIDAgMCAwIDAgMCAyNzggMCAwIDAgMCAwID
AgMCAwIDAgMCAzMzMgMCAwIDAgMCAwIDAgMCA3MjlgMCA3MjlgMCA2MTEgNzc4IDAgMC
AwIDAgMCAwIDAgMCA2NjcgNzc4IDcyMiA2NjcgMCAwIDAgMCAwIDAgMCAwIDAgMCAwIDAg
MCA1NTYgMCA1NTYgNjExIDU1NiAwIDYxMSA2MTEgMjc4IDAgNTU2IDI3OCA4ODkgNjExIDYx
MSA2MTEgMCAzODkgNTU2IDMzMjY2MTEgNTU2IDAgNTU2IDU1Ni0glDQplbmRvYmoNCjlyMC
AwIG9iag0KWYAyNzhdlA0KZW5kb2JqDQoyMzEgMCAwIDAgMCAwIDAgMCAwIDAgMCAwIDAg
MS9XWYAxIDQgMl0gl1Jvb3QgMSAwIFlvSW5mbyAyMjUgMCAwIDAgMCAwIDAgMCAwIDAgMCAwIDAg
JDOTRDQUE5N0NBmJlXmJg5RjIjCMz48RDFBQzUwRDRFRjYyQzk0Q0FBOTdDQTlyMTI4OUY5
QjM+XSAvRmlsdGVyL0ZsYXRIRGVjb2RlL0xlbmd0aCA0NDg+Pg0Kc3RyZWftDQp4nDXUVvdV
ARAG0HOxWxCwA1sRu7u7G7uwuwu7u7ELLOzA7u7ID/HFX4CXs3UeZq95mJm3LwiilZsbifbYIM
gjHTkhkT8hsfXwKyTuEX6HIEICJt6ExE9CdkhCVkhir5Cy9YMGJvqhQvAN3/EDX/EJ+RDBv72f0Su
pA/9PMSiBAsiPwiilQiiGllK4ohHSZRGKcQhFmVQGQkoh0SURUWURwVUQg1URRUkoRqqoyFq
oyZqoR7qoC4aoD6SkYlWalxGalYmalqWal4WalVualc2aluOal8O6lJO6lyuGIAe6l7e6lle6lc+6lv+
GI3BGihBGIYhGIqRGI4RGIUpGINUjMdYjMMkTMBETMY8pGEqpmEmpmMG5mAWZmMuVmAB
5mMxMfMlRImEJlml5NmE1VmIV0rEGa7EB67AeG7EHw7AZ27EV27ALO7ATu5GB/diLftIEAziloz
iMziGCziB4ziNkziFczlDszlPG8jCRWTiCi7hMrJxFddwHQ9xCzdxF7dxBw9wD/eRg+d4jEd4iid4hjd4
iRd4hdf4iHd4i/f4gM/4Eg3ctlwwwvadPC5lxKI9lclplg6khKQuC4C/QKXqFDQplbmRzdHJlYXW0NCmV
uZG9iag0KeHJIZg0KMCAyMzINCjAwMDAwMDAwMTcgNjU1MzUgZg0KMDAwMDAwMDAxNyA
wMDAwMCBuDQowMDAwMDAwMTI1IDAwMDAwIG4NCjAwMDAwMDAxODEgMDAwMDAgbg0
KMDAwMDAwMDQ5NyAwMDAwMCBuDQowMDAwMDA0MzkzIDAwMDAwIG4NCjAwMDAwMD
Q1NjAgMDAwMDAgbg0KMDAwMDAwNDc5MSAwMDAwMCBuDQowMDAwMDA0ODQ0IDAwM
DAwIG4NCjAwMDAwMDQ4OTcgMDAwMDAgbg0KMDAwMDAwNTAyNyAwMDAwMCBuDQowM
DAwMDA1MDU3IDAwMDAwIG4NCjAwMDAwMDUyMTYgMDAwMDAgbg0KMDAwMDAwNTI5M

CAwMDAwMCBuDQowMDAwMDA1NTQwIDA wMDAwIG4NCjAwMDAwMDU3MDMgMDAwMDA
gbg0KMDAwMDAwNTkzMCAwMDAwMCBuDQowMDAwMDAwMDE4IDY1NTM1IGYNCjAwMDA
wMDAwMTkgNjU1MzUgZg0KMDAwMDAwMDAyMCA2NTUzNSBmDQowMDAwMDAwMDIxIDY1
NTM1IGYNCjAwMDAwMDAwMjlgNjU1MzUgZg0KMDAwMDAwMDAyMyA2NTUzNSBmDQowMD
AwMDAwMDI0IDY1NTM1IGYNCjAwMDAwMDAwMjUgNjU1MzUgZg0KMDAwMDAwMDAyNiA2N
TUzNSBmDQowMDAwMDAwMDI3IDY1NTM1IGYNCjAwMDAwMDAwMjggNjU1MzUgZg0KMDA
wMDAwMDAyOSA2NTUzNSBmDQowMDAwMDAwMDMwIDY1NTM1IGYNCjAwMDAwMDAwMz
EgNjU1MzUgZg0KMDAwMDAwMDAzMiA2NTUzNSBmDQowMDAwMDAwMDMzIDY1NTM1IGY
NCjAwMDAwMDAwMzQgNjU1MzUgZg0KMDAwMDAwMDAzNSA2NTUzNSBmDQowMDAwMDA
wMDM2IDY1NTM1IGYNCjAwMDAwMDAwMzcgNjU1MzUgZg0KMDAwMDAwMDAzOCA2NTUzN
SBmDQowMDAwMDAwMDM5IDY1NTM1IGYNCjAwMDAwMDAwNDAgNjU1MzUgZg0KMDAwMD
DAwMDA0MSA2NTUzNSBmDQowMDAwMDAwMDQyIDY1NTM1IGYNCjAwMDAwMDAwNDMg
NjU1MzUgZg0KMDAwMDAwMDA0NCA2NTUzNSBmDQowMDAwMDAwMDQ1IDY1NTM1IGYN
CjAwMDAwMDAwNDYgNjU1MzUgZg0KMDAwMDAwMDA0NyA2NTUzNSBmDQowMDAwMDAw
MDQ4IDY1NTM1IGYNCjAwMDAwMDAwNDkgNjU1MzUgZg0KMDAwMDAwMDA1MCA2NTUzN
SBmDQowMDAwMDAwMDUxIDY1NTM1IGYNCjAwMDAwMDAwNTlgNjU1MzUgZg0KMDAwMD
AwMDA1MyA2NTUzNSBmDQowMDAwMDAwMDU0IDY1NTM1IGYNCjAwMDAwMDAwNTUgNjU
1MzUgZg0KMDAwMDAwMDA1NiA2NTUzNSBmDQowMDAwMDAwMDU3IDY1NTM1IGYNCjAw
MDAwMDAwNTggNjU1MzUgZg0KMDAwMDAwMDA1OSA2NTUzNSBmDQowMDAwMDAwMDY
wIDY1NTM1IGYNCjAwMDAwMDAwNjEgNjU1MzUgZg0KMDAwMDAwMDA2MiA2NTUzNSBmD
QowMDAwMDAwMDYzIDY1NTM1IGYNCjAwMDAwMDAwNjQgNjU1MzUgZg0KMDAwMDAwMD
A2NSA2NTUzNSBmDQowMDAwMDAwMDY2IDY1NTM1IGYNCjAwMDAwMDAwNjcqNjU1MzUg
Zg0KMDAwMDAwMDA2OCA2NTUzNSBmDQowMDAwMDAwMDY5IDY1NTM1IGYNCjAwMDAw
MDAwNzAgNjU1MzUgZg0KMDAwMDAwMDA3MSA2NTUzNSBmDQowMDAwMDAwMDcyIDY1
NTM1IGYNCjAwMDAwMDAwNzMgNjU1MzUgZg0KMDAwMDAwMDA3NCA2NTUzNSBmDQow
MDAwMDAwMDc1IDY1NTM1IGYNCjAwMDAwMDAwNzYgNjU1MzUgZg0KMDAwMDAwMDA3N
yA2NTUzNSBmDQowMDAwMDAwMDc4IDY1NTM1IGYNCjAwMDAwMDAwNzkgNjU1MzUgZg0K
MDAwMDAwMDA4MCA2NTUzNSBmDQowMDAwMDAwMDgxIDY1NTM1IGYNCjAwMDAwMDA
wODlgNjU1MzUgZg0KMDAwMDAwMDA4MyA2NTUzNSBmDQowMDAwMDAwMDg0IDY1NTM1I
GYNCjAwMDAwMDAwODUgNjU1MzUgZg0KMDAwMDAwMDA4NiA2NTUzNSBmDQowMDAwMD
DAwMDg3IDY1NTM1IGYNCjAwMDAwMDAwODggNjU1MzUgZg0KMDAwMDAwMDA4OSA2NT
UzNSBmDQowMDAwMDAwMDkwIDY1NTM1IGYNCjAwMDAwMDAwOTEgNjU1MzUgZg0KMDA
wMDAwMDA5MiA2NTUzNSBmDQowMDAwMDAwMDkzIDY1NTM1IGYNCjAwMDAwMDAwOTQ
gNjU1MzUgZg0KMDAwMDAwMDA5NSA2NTUzNSBmDQowMDAwMDAwMDk2IDY1NTM1IGYN
CjAwMDAwMDAwOTcgNjU1MzUgZg0KMDAwMDAwMDA5OCA2NTUzNSBmDQowMDAwMDAw
MDk5IDY1NTM1IGYNCjAwMDAwMDAxMDAgNjU1MzUgZg0KMDAwMDAwMDEwMSA2NTUzNS
BmDQowMDAwMDAwMTAyIDY1NTM1IGYNCjAwMDAwMDAxMDMgNjU1MzUgZg0KMDAwMDA
wMDEwNCA2NTUzNSBmDQowMDAwMDAwMTA1IDY1NTM1IGYNCjAwMDAwMDAxMDYgNjU1
MzUgZg0KMDAwMDAwMDEwNyA2NTUzNSBmDQowMDAwMDAwMTA4IDY1NTM1IGYNCjAw
MDAwMDAxMDkgNjU1MzUgZg0KMDAwMDAwMDExMCA2NTUzNSBmDQowMDAwMDAwMTE
xIDY1NTM1IGYNCjAwMDAwMDAxMTlgNjU1MzUgZg0KMDAwMDAwMDExMyA2NTUzNSBmDQ
owMDAwMDAwMTE0IDY1NTM1IGYNCjAwMDAwMDAxMTUgNjU1MzUgZg0KMDAwMDAwMDE

xNiA2NTUzNSBmDQowMDAwMDAwMTE3IDY1NTM1IGYNCjAwMDAwMDAxMTggNjU1MzUgZg0KMDAwMDAwMDExOSA2NTUzNSBmDQowMDAwMDAwMTlwIDY1NTM1IGYNCjAwMDAwMDAxMjEgNjU1MzUgZg0KMDAwMDAwMDEyMiA2NTUzNSBmDQowMDAwMDAwMTIzIDY1NTM1IGYNCjAwMDAwMDAxMjQgNjU1MzUgZg0KMDAwMDAwMDEyNSA2NTUzNSBmDQowMDAwMDAwMTI2IDY1NTM1IGYNCjAwMDAwMDAxMjcjNjU1MzUgZg0KMDAwMDAwMDEyOCA2NTUzNSBmDQowMDAwMDAwMTI5IDY1NTM1IGYNCjAwMDAwMDAxMzAgNjU1MzUgZg0KMDAwMDAwMDEzMCA2NTUzNSBmDQowMDAwMDAwMTMyIDY1NTM1IGYNCjAwMDAwMDAxMzMgNjU1MzUgZg0KMDAwMDAwMDEzNCA2NTUzNSBmDQowMDAwMDAwMTM1IDY1NTM1IGYNCjAwMDAwMDAxMzYgNjU1MzUgZg0KMDAwMDAwMDEzNyA2NTUzNSBmDQowMDAwMDAwMTM4IDY1NTM1IGYNCjAwMDAwMDAxMzkgNjU1MzUgZg0KMDAwMDAwMDE0MCA2NTUzNSBmDQowMDAwMDAwMTQxIDY1NTM1IGYNCjAwMDAwMDAxNDIgNjU1MzUgZg0KMDAwMDAwMDE0MyA2NTUzNSBmDQowMDAwMDAwMTQ0IDY1NTM1IGYNCjAwMDAwMDAxNDUgNjU1MzUgZg0KMDAwMDAwMDE0NiA2NTUzNSBmDQowMDAwMDAwMTQ3IDY1NTM1IGYNCjAwMDAwMDAxNDggNjU1MzUgZg0KMDAwMDAwMDE0OSA2NTUzNSBmDQowMDAwMDAwMTUwIDY1NTM1IGYNCjAwMDAwMDAxNTEgNjU1MzUgZg0KMDAwMDAwMDE1MiA2NTUzNSBmDQowMDAwMDAwMTUzIDY1NTM1IGYNCjAwMDAwMDAxNTQgNjU1MzUgZg0KMDAwMDAwMDE1NSA2NTUzNSBmDQowMDAwMDAwMTU2IDY1NTM1IGYNCjAwMDAwMDAxNTcgNjU1MzUgZg0KMDAwMDAwMDE1OCA2NTUzNSBmDQowMDAwMDAwMTU5IDY1NTM1IGYNCjAwMDAwMDAxNjAgNjU1MzUgZg0KMDAwMDAwMDE2MSA2NTUzNSBmDQowMDAwMDAwMTYyIDY1NTM1IGYNCjAwMDAwMDAxNjMgNjU1MzUgZg0KMDAwMDAwMDE2NCA2NTUzNSBmDQowMDAwMDAwMTY1IDY1NTM1IGYNCjAwMDAwMDAxNjYgNjU1MzUgZg0KMDAwMDAwMDE2NyA2NTUzNSBmDQowMDAwMDAwMTY4IDY1NTM1IGYNCjAwMDAwMDAxNjkgNjU1MzUgZg0KMDAwMDAwMDE3MCA2NTUzNSBmDQowMDAwMDAwMTcxIDY1NTM1IGYNCjAwMDAwMDAxNzIgNjU1MzUgZg0KMDAwMDAwMDE3MyA2NTUzNSBmDQowMDAwMDAwMTc0IDY1NTM1IGYNCjAwMDAwMDAxNzUgNjU1MzUgZg0KMDAwMDAwMDE3NiA2NTUzNSBmDQowMDAwMDAwMTc3IDY1NTM1IGYNCjAwMDAwMDAxNzggNjU1MzUgZg0KMDAwMDAwMDE3OSA2NTUzNSBmDQowMDAwMDAwMTgwIDY1NTM1IGYNCjAwMDAwMDAxODEgNjU1MzUgZg0KMDAwMDAwMDE4MiA2NTUzNSBmDQowMDAwMDAwMTgzIDY1NTM1IGYNCjAwMDAwMDAxODQgNjU1MzUgZg0KMDAwMDAwMDE4NSA2NTUzNSBmDQowMDAwMDAwMTg2IDY1NTM1IGYNCjAwMDAwMDAxODcgNjU1MzUgZg0KMDAwMDAwMDE4OCA2NTUzNSBmDQowMDAwMDAwMTg5IDY1NTM1IGYNCjAwMDAwMDAxOTAgNjU1MzUgZg0KMDAwMDAwMDE5MSA2NTUzNSBmDQowMDAwMDAwMTkyIDY1NTM1IGYNCjAwMDAwMDAxOTMgNjU1MzUgZg0KMDAwMDAwMDE5NCA2NTUzNSBmDQowMDAwMDAwMTk1IDY1NTM1IGYNCjAwMDAwMDAxOTYgNjU1MzUgZg0KMDAwMDAwMDE5NyA2NTUzNSBmDQowMDAwMDAwMTk4IDY1NTM1IGYNCjAwMDAwMDAxOTkgNjU1MzUgZg0KMDAwMDAwMDIwMCA2NTUzNSBmDQowMDAwMDAwMjAxIDY1NTM1IGYNCjAwMDAwMDAyMDIgNjU1MzUgZg0KMDAwMDAwMDIwMyA2NTUzNSBmDQowMDAwMDAwMjA0IDY1NTM1IGYNCjAwMDAwMDAyMDUgNjU1MzUgZg0KMDAwMDAwMDIwNiA2NTUzNSBmDQowMDAwMDAwMjA3IDY1NTM1IGYNCjAwMDAwMDAyMDggNjU1MzUgZg0KMDAwMDAwMDIwOSA2NTUzNSBmDQowMDAwMDAwMjEwIDY1NTM1IGYNCjAwMDAwMDAyMTEgNjU1MzUgZg0KMDAwMDAwMDIxMiA2NTUzNSBmDQowMDAwMDAwMjEzIDY1NTM1IGYNCjAwMDAwMDAyMTQgNjU1MzUgZg0KMDAwMDAwMDIxNSA2NTUzNSBmDQowMDAwMDAwMjE2IDY1NTM1IGYNCjAwMDAwMDAyMTcgNjU1MzUgZg0KMDAwMDAwMDIxOCA2NTUzNSBmDQowMDAwMDAw

wMjE5IDY1NTM1IGYNCjAwMDAwMDAyMjAgNjU1MzUgZg0KMDAwMDAwMDIyMSA2NTUzNSB
mDQowMDAwMDAwMjlyIDY1NTM1IGYNCjAwMDAwMDAyMjMgNjU1MzUgZg0KMDAwMDAwM
DIyNCA2NTUzNSBmDQowMDAwMDAwMDAwIDY1NTM1IGYNCjAwMDAwMjYwMDcgMDAwMD
Agbg0KMDAwMDAyNjlxMiAwMDAwMCBuDQowMDAwMDI2NTE0IDAwMDAwIG4NCjAwMDAwN
zY2NDUgMDAwMDAgbg0KMDAwMDA3Njg5OSAwMDAwMCBuDQowMDAwMDc3MTY3IDAwM
DAwIG4NCjAwMDAwNzcxOTUgMDAwMDAgbg0KdHJhaWxlcg0KPDwwU2l6ZSAyMzlvUm9vdCA
xIDAgUi9JbmZvIDlyNSAwIFlvSURbPEQxQUM1MEQ0RUY2MkM5NENBQTK3Q0EyMjEyODIGO
UIzPjxEMUFDNTBENEVGNjJDOTRDQUE5N0NBMjlxMjg5RjIjCMz5dID4+DQpzdGFydHhyZWYN
Cjc3ODQ3DQoIJUVPRg0KeHJIZg0KMCAwDQp0cmFpbGVyDQo8PC9TaXplIDlzMi9Sb290IDEg
MCBSL0luZm8gMjI1IDAgUi9JRFRs8RDFBQzUwRDRFRjYyQzk0Q0FBOTdDQTIyMTI4OUY5QjM+
PEQxQUM1MEQ0RUY2MkM5NENBQTK3Q0EyMjEyODIGOUlZPI0gL1ByZXYgNzc4NDcvWFJIZI
N0bSA3NzE5NT4+DQpzdGFydHhyZWYNCjgyNjQ3DQoIJUVPRg==";

}

}

}

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Optimization;
using System.Web.UI;
```

```
namespace Web_App_Master
```

```
{
```

```
    public class BundleConfig
```

```
    {
```

```
        // For more information on Bundling, visit https://go.microsoft.com/fwlink/?LinkID=303951
```

```
        public static void RegisterBundles(BundleCollection bundles)
```

```
        {
```

```
            bundles.Add(new ScriptBundle("~/bundles/WebFormsJs").Include(
```

```
                "~/Scripts/WebForms/WebForms.js",
```

```
                "~/Scripts/WebForms/WebUIValidation.js",
```

```
                "~/Scripts/WebForms/MenuStandards.js",
```

```
                "~/Scripts/WebForms/Focus.js",
```

```
                "~/Scripts/WebForms/GridView.js",
```

```
                "~/Scripts/WebForms/DetailsView.js",
```

```
                "~/Scripts/WebForms/TreeView.js",
```

```
                "~/Scripts/WebForms/WebParts.js"));
```

```
        // Order is very important for these files to work, they have explicit dependencies
```

```
        bundles.Add(new ScriptBundle("~/bundles/MsAjaxJs").Include(
```

```
            "~/Scripts/WebForms/MsAjax/MicrosoftAjax.js",
```

```
            "~/Scripts/WebForms/MsAjax/MicrosoftAjaxApplicationServices.js",
```

```
            "~/Scripts/WebForms/MsAjax/MicrosoftAjaxTimer.js",
```

```
            "~/Scripts/WebForms/MsAjax/MicrosoftAjaxWebForms.js"));
```

```
        // Use the Development version of Modernizr to develop with and learn from. Then, when
        you're
```

```
        // ready for production, use the build tool at https://modernizr.com to pick only the tests you
        need
```

```
        bundles.Add(new ScriptBundle("~/bundles/modernizr").Include(
```

```
            "~/Scripts/modernizr-*"));
```

```
        ScriptManager.ScriptResourceMapping.AddDefinition(
```

```
            "respond",
```

```
            new ScriptResourceDefinition
```

```
            {
```

```
    Path = "~/Scripts/respond.min.js",  
    DebugPath = "~/Scripts/respond.js",  
  });  
}  
}
```

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Linq;
using System.Web;
using System.Xml.Serialization;
using static Notification.NotificationSystem;
```

```
namespace Data
```

```
{
    [Serializable]
    public class LogEntry
    {
        public LogEntry()
        {
            Time = DateTime.Now;
            Entry = "";
            Href = "#";
        }
        public LogEntry(string entry)
        {
            Time = DateTime.Now;
            Entry = entry;
            Href = "#";
        }
        public LogEntry(string entry, string href)
        {
            Time = DateTime.Now;
            Entry = entry;
            Href = href;
        }
        [XmlElement]
        public DateTime Time { get; set; }
        [XmlElement]
        public string Entry { get; set; }
        [XmlElement]
        public string Href { get; set; }
        [XmlElement]
        public string TimeString { get { return Time.ToShortTimeString(); } }
    }
    public class UserDataBindingList : BindingList<UserData>
    {
```



```

public UserDataBindingList() { this.AllowEdit = AllowNew = AllowRemove = true; }
public UserDataBindingList(List<UserData> userData)
{
    this.AllowEdit = AllowNew = AllowRemove = true;
    foreach (var ud in userData)
    {
        this.Add(ud);
    }
}
}
[Serializable]
public class UserData:Serializers.XSerializer<UserData>,ICloneable
{
    public UserData()
    {
        Name = Email = "";
        Log = new List<LogEntry>();
        Notices = new List<Notice>();
        Attachments = new List<Attachment>();
        IsAutoChecked = false;
    }
    [XmlElement]
    public string m_guid = System.Guid.NewGuid().ToString();
    [XmlElement]
    public string Guid { get { return m_guid; } set { m_guid = value; } }
    [XmlElement]
    public string Name { get; set; }

    [XmlElement]
    public string Email { get; set; }

    [XmlElement]
    public List<LogEntry> Log { get; set; }
    [XmlElement]
    public List<Notice> Notices { get; set; }
    [XmlElement]
    public List<Attachment> Attachments { get; set; }
    [XmlElement]
    public bool IsAutoChecked { get; set; }

    public object Clone()
    {

```

```

        return this.MemberwiseClone();
    }

}

[Serializable]
public class Attachment : Serializers.XSerializer<Attachment>, ICloneable
{
    public Attachment()
    {
        Name = Email = "";
        Files = new List<FileReference>();
        Date = DateTime.Now;

    }
    [XmlElement]
    public DateTime Date { get; set; }
    [XmlElement]
    public string Name { get; set; }
    [XmlElement]
    public string Email { get; set; }
    [XmlElement]
    public List<FileReference> Files { get; set; }

    public object Clone()
    {
        return this.MemberwiseClone();
    }

}

[Serializable]
public class FileReference : Serializers.XSerializer<UserData>
{
    public FileReference()
    {
        Name = Link = "";
        Date = DateTime.Now;
    }
    public FileReference(string name, string link)
    {
        Name = name; Link = link;
        Date = DateTime.Now;
    }
}

```

```
[XmlElement]
public DateTime Date { get; set; }
[XmlElement]
public string Name { get; set; }
[XmlElement]
public string Link { get; set; }
}
}
```

```
using Helpers;
using Microsoft.AspNet.Identity;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Net.Mail;
using System.Text;
using System.Threading.Tasks;
using System.Web;
using Web_App_Master;
```

```
namespace Web_App_Master
```

```
{
    public class EmailHelper
    {
        public static bool IsValid(string emailaddress)
        {
            try
            {
                MailAddress m = new MailAddress(emailaddress);

                return true;
            }
            catch (FormatException)
            {
                return false;
            }
        }
        public static bool IsValid(EmailAddress emailaddress)
        {
            try
            {
                MailAddress m = new MailAddress(emailaddress.Email);

                return true;
            }
            catch (FormatException)
            {
                return false;
            }
        }
    }
}
```

```

public static Task<bool> SendAsync(IdentityMessage message)
{
    // Plug in your email service here to send an email.
    return
Task.FromResult(SendEmail(message.Destination,message.Body,message.Subject));
}
public static bool SendEmail(string email,string body = "", string subject = "")
{

    MailMessage msg = new MailMessage();
    msg.Subject = subject;
    msg.IsBodyHtml = true;
    msg.BodyEncoding = Encoding.ASCII;
    msg.Body = body;
    msg.From = new MailAddress(Global.Library.Settings.SenderEmail);
    msg.To.Add(email);
    string username = Global.Library.Settings.SenderEmail; //email address or domain user
for exchange authentication
    string password = Global.Library.Settings.SenderPassword; //password
    SmtpClient mClient = new SmtpClient();
    mClient.Host = Global.Library.Settings.SenderHost;
    mClient.Port = Global.Library.Settings.SenderPort;
    mClient.UseDefaultCredentials = false;
    mClient.Credentials = new NetworkCredential(username, password);
    mClient.DeliveryMethod = SmtpDeliveryMethod.Network;
    mClient.Timeout = 100000;
    mClient.EnableSsl = true;
    try
    {
        mClient.Send(msg);
        return true;
    }
    catch
    {
        return false;
    }

}
public static void SendEmailAsync(string email,string body = "", string subject = "")
{

```

```

new System.Threading.Thread(() =>
{

    MailMessage msg = new MailMessage();
    msg.Subject = subject;
    msg.IsBodyHtml = true;
    msg.BodyEncoding = Encoding.ASCII;
    msg.Body = body;
    msg.From = new MailAddress("provider.service.secure@gmail.com");
    msg.To.Add(email);
    string username = "provider.service.secure@gmail.com"; //email address or domain
user for exchange authentication
    string password = "@Service1"; //password
    SmtpClient mClient = new SmtpClient();
    mClient.Host = "smtp.gmail.com";
    mClient.Port = 587;
    mClient.UseDefaultCredentials = false;
    mClient.Credentials = new NetworkCredential(username, password);
    mClient.DeliveryMethod = SmtpDeliveryMethod.Network;
    mClient.Timeout = 100000;
    mClient.EnableSsl = true;
    try
    {
        mClient.Send(msg);
    }
    catch
    {

    }
}).Start();
}
public static void SendMassEmail(string[] emails, string body = "", string subject = "")
{

```

```

    MailMessage msg = new MailMessage();
    msg.Subject = subject;
    msg.IsBodyHtml = true;
    msg.BodyEncoding = Encoding.ASCII;
    msg.Body = body;

```

```

        msg.From = new MailAddress("provider.service.secure@gmail.com");
        foreach (var email in emails)
        {
            msg.To.Add(email);
        }
        string username = "provider.service.secure@gmail.com"; //email address or domain
user for exchange authentication
        string password = "@Service1"; //password
        SmtpClient mClient = new SmtpClient();
        mClient.Host = "smtp.gmail.com";
        mClient.Port = 587;
        mClient.UseDefaultCredentials = false;
        mClient.Credentials = new NetworkCredential(username, password);
        mClient.DeliveryMethod = SmtpDeliveryMethod.Network;
        mClient.Timeout = 100000;
        mClient.EnableSsl = true;
        try
        {
            mClient.Send(msg);
        }
        catch
        {

        }

    }

    public static void SendMassEmailAsync(string[] emails, string body = "", string subject = "")
    {

        new System.Threading.Thread(() =>
        {

            MailMessage msg = new MailMessage();
            msg.Subject = subject;
            msg.IsBodyHtml = true;
            msg.BodyEncoding = Encoding.ASCII;
            msg.Body = body;
            msg.From = new MailAddress("provider.service.secure@gmail.com");
            foreach(var email in emails )
            {
                msg.To.Add(email);
            }
        }
    )
    }

```

```

    }
    string username = "provider.service.secure@gmail.com"; //email address or domain
user for exchange authentication
    string password = "@Service1"; //password
    SmtpClient mClient = new SmtpClient();
    mClient.Host = "smtp.gmail.com";
    mClient.Port = 587;
    mClient.UseDefaultCredentials = false;
    mClient.Credentials = new NetworkCredential(username, password);
    mClient.DeliveryMethod = SmtpDeliveryMethod.Network;
    mClient.Timeout = 100000;
    mClient.EnableSsl = true;
    try
    {
        mClient.Send(msg);
    }
    catch
    {

    }
    }).Start();
}
public static void SendCheckOutNoticeAsync(List<Asset> assets, string Body = "")
{
    var engineer = (from d in Global.Library.Settings.ServiceEngineers where d.Name ==
assets[0].ServiceEngineer select d).FirstOrDefault();
    var statics = (from d in Global.Library.Settings.StaticEmails select d.Email).ToList();

    var emaillist = new List<string>();
    //emaillist.Add(shipper.Email);
    emaillist.Add(engineer.Email);
    emaillist.AddRange(statics);

    new System.Threading.Thread(() =>
    {

        Body = Body.Replace("<name>", assets[0].ServiceEngineer);
        var serviceToolString = "";
        foreach(var item in assets)
        {
            serviceToolString += "(" + item.AssetName + ")";

```



```

    }

    Body = Body.Replace("<serviceTool>", serviceToolString);
    Body = Body.Replace("<serviceOrder>", assets[0].OrderNumber.ToString());
    Body = Body.Replace("<dateAssigned>", assets[0].DateShipped.ToString());
    Body = Body.Replace("<NL>", "<br />");
    MailMessage msg = new MailMessage();
    msg.Subject = "Asset Alert:: " + assets[0].AssetName + " :: " +
DateTime.Now.ToString();
    msg.IsBodyHtml = true;
    msg.BodyEncoding = Encoding.ASCII;
    msg.Body = Body;
    msg.From = new MailAddress("provider.service.secure@gmail.com");
    foreach (var email in emailist)
    {
        msg.To.Add(email);
    }
    string username = "provider.service.secure@gmail.com"; //email address or domain
user for exchange authentication
    string password = "@Service1"; //password
    SmtpClient mClient = new SmtpClient();
    mClient.Host = "smtp.gmail.com";
    mClient.Port = 587;
    mClient.UseDefaultCredentials = false;
    mClient.Credentials = new NetworkCredential(username, password);
    mClient.DeliveryMethod = SmtpDeliveryMethod.Network;
    mClient.Timeout = 100000;
    mClient.EnableSsl = true;
    try
    {
        mClient.Send(msg);
    }
    catch
    {

    }
    }).Start();
}
public static void SendNotificationSystemNotice(Notification.NotificationSystem.Notice notice)
{
    if (notice.NoticeType== Notification.NotificationSystem.NoticeType.Checkout)
    {

```

```

if (Global.Library.Settings.TESTMODE) return;
if (notice is EmailNotice )
{
    var en = notice as EmailNotice;
    string Body = en.Body;
    new System.Threading.Thread(() =>
    {

        Body = Body.Replace("<name>", en.EmailAddress.Name);
        var serviceToolString = " < br />";
        foreach (var item in en.Assets)
        {
            serviceToolString += "(" + item + ")";
        }

        Body = Body.Replace("<serviceTool>", serviceToolString);
        Body = Body.Replace("<serviceOrder>", en.NoticeControlNumber);
        Body = Body.Replace("<dateAssigned>", en.Created.ToShortDateString());
        Body = Body.Replace("<NL>", "<br />");
        MailMessage msg = new MailMessage();
        msg.Subject = "Return Request:: Job# " + en.NoticeControlNumber + " :: " +
DateTime.Now.ToString();
        msg.IsBodyHtml = true;
        msg.BodyEncoding = Encoding.ASCII;
        msg.Body = Body;
        msg.From = new MailAddress(Global.Library.Settings.SenderEmail);
        foreach (var email in en.Emails)
        {
            msg.To.Add(email.Email);
        }
        string username = Global.Library.Settings.SenderEmail; //email address or domain
user for exchange authentication
        string password = Global.Library.Settings.SenderPassword; //password
        SmtpClient mClient = new SmtpClient();
        mClient.Host = Global.Library.Settings.SenderHost;
        mClient.Port = Global.Library.Settings.SenderPort;
        mClient.UseDefaultCredentials = false;
        mClient.Credentials = new NetworkCredential(username, password);
        mClient.DeliveryMethod = SmtpDeliveryMethod.Network;
        mClient.Timeout = 100000;
        mClient.EnableSsl = true;
        try

```

```
        {
            mClient.Send(msg);
        }
        catch
        {
        }
    }).Start();
}

}

}
```

```
using Microsoft.AspNet.Identity.EntityFramework;
using Microsoft.AspNet.Identity.Owin;
using System;
using System.Collections.Concurrent;
using System.Collections.Generic;
using System.ComponentModel;
using System.IO;
using System.Linq;
using System.Net;
using System.Runtime.Serialization;
using System.Runtime.Serialization.Formatters.Binary;
using System.Text.RegularExpressions;
using System.Web;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Xml;
using System.Xml.Serialization;
using Web_App_Master;
using Web_App_Master.Models;
using Microsoft.AspNet.Identity;
using Microsoft.AspNet.Identity.Owin;
```

namespace Helpers

```
{
    public static class Extensions
    {
        public static bool IsAdmin(this HttpContext context)
        {
            if (context.User.IsInRole("Admins") || context.User.IsInRole("superadmin"))
            {
                return true;
            }
            else
            {
                return false;
            }
        }

        public static bool IsCustomer(this HttpContext context)
        {

```

```

        if (context.User.IsInRole("Customers"))
        {
            return true;
        }
        else
        {
            return false;
        }
    }

    public static bool IsUser(this HttpContext context)
    {
        if (context.User.IsInRole("Users"))
        {
            return true;
        }
        else
        {
            return false;
        }
    }

    public static Control RecursiveFindControl(this Control parent, string idToFind)
    {
        foreach (Control child in parent.Controls)
        {
            if (child.ID == idToFind)
            {
                return child;
            }
            else
            {
                Control control = RecursiveFindControl(child, idToFind);
                if (control != null)
                {
                    return control;
                }
            }
        }
        return null;
    }

    public static Control FindRoleControl(this LoginView lv, string controlName, string role, string

```

```

parentControl = null)
{
    int idx = 0;
    foreach (RoleGroup r in lv.RoleGroups)
    {
        if (r.Roles.Contains(role))
        {
            break;
        }
        ++idx;
    }
    Control c = null;
    ITemplate template = lv.RoleGroups[0].ContentTemplate;
    if (template != null)
    {
        Control container = new Control();
        template.InstantiateIn(container);

        if (parentControl != null)
        {
            var parent = container.RecursiveFindControl(parentControl);
            c = parent.RecursiveFindControl(controlName);
        }
        else
        {
            c = container.RecursiveFindControl(controlName);
        }
    }
    return c;
}

public static void Shake(this HtmlGenericControl control, string animation = " mif-ani-flash")
{
    var a = control.Attributes["class"];
    a += animation;
    control.Attributes["class"] = a;
}

public static void Quiet(this HtmlGenericControl control, string animation = " mif-ani-flash")
{
    var a = control.Attributes["class"];
    a = a.Replace(animation, "");
    control.Attributes["class"] = a;
}

```

```

}
public static IEnumerable<List<T>> SplitList<T>(this List<T> list, int nSize = 30)
{
    if (list.Count <= 30)
    { return new List<List<T>>() { list }; }
    try
    {
        var tmp = new T[list.Count];
        list.CopyTo(tmp);
        var temp = new List<T>(tmp);
        var ret = new List<List<T>>();
        int count = 0;
        do
        {
            var page = temp.Take(nSize).ToList();
            foreach (var item in page)
            {
                temp.Remove(item);
            }
            ret.Add(page);
            count += page.Count;
        } while (count < list.Count);
        return ret;
    }
    catch {
        return new List<List<T>>();
    }
}

public static SiteMaster SiteMaster(this Page page)
{
    return page.Master as SiteMaster;
}

public static void UpdateAll(this Page page)
{
    (page.Master as SiteMaster).UpdateAllPanels();
}

public static List<Asset> ImportXmlLibraryFile(string filename)
{
    var doc = new XmlDocument();
    using (StreamReader reader = new StreamReader(filename))

```

```
{
    doc.LoadXml(reader.ReadToEnd());
}
```

```
Global.Library.Assets = new List<Asset>();
```

```
XmlNodeList elemList = doc.GetElementsByTagName("Asset");
foreach (XmlElement asset in elemList)
{
    Asset a = new Asset();
    try
    {
        a.AssetName = asset.GetAttribute("Name").Sanitize();
    }
    catch { }
    try
    {
        a.AssetNumber = asset.SelectSingleNode("AssetNumber").InnerText.Sanitize();
    }
    catch { }
    try
    {
        a.DateRecieved =
DateTime.Parse(asset.SelectSingleNode("DateRecieved").InnerText.Sanitize());
    }
    catch { }
    try
    {
        a.DateShipped =
DateTime.Parse(asset.SelectSingleNode("DateShipped").InnerText.Sanitize());
    }
    catch { }
    try
    {
        a.Description = asset.SelectSingleNode("Description").InnerText.Sanitize();
    }
    catch { }
    try
    {
        a.OrderNumber = asset.SelectSingleNode("OrderNumber").InnerText.Sanitize();
    }
}
```



```

    }
    catch { }
    try
    {
        a.PackingSlip = asset.SelectSingleNode("PackingSlip").InnerText.Sanitize();
    }
    catch { }
    try
    {
        a.PersonShipping = asset.SelectSingleNode("PersonShipping").InnerText.Sanitize();
    }
    catch { }
    try
    {
        a.ReturnReport = asset.SelectSingleNode("ReturnReport").InnerText.Sanitize();
    }
    catch { }
    try
    {
        a.ServiceEngineer = asset.SelectSingleNode("ServiceEngineer").InnerText.Sanitize();
    }
    catch { }
    try
    {
        a.ShipTo = asset.SelectSingleNode("ShipTo").InnerText.Sanitize();
    }
    catch { }
    try
    {
        a.UpsLabel = asset.SelectSingleNode("UpsLabel").InnerText.Sanitize();
    }
    catch { }
    try
    {
        a.weight =
Convert.ToDecimal(asset.SelectSingleNode("Weight").InnerText.Sanitize());
    }
    catch { }
    try
    {
        a.IsOut = Convert.ToBoolean(asset.SelectSingleNode("IsOut").InnerText.Sanitize());
    }

```

```

        catch { }
    try
    {
        a.OnHold =
Convert.ToBoolean(asset.SelectSingleNode("OnHold").InnerText.Sanitize());
    }
    catch { }
    try
    {
        a.IsDamaged =
Convert.ToBoolean(asset.SelectSingleNode("IsDamaged").InnerText.Sanitize());
    }
    catch { }

    try
    {
        XmlNodeList imglist = asset.GetElementsByTagName("Image");

        foreach (XmlElement el in imglist)
        {
            a.Images += "" + el.InnerText + ",";
        }
    }
    catch { }

    Global.Library.Assets.Add(a);

}

return Global.Library.Assets;
}
public static bool CheckForInternetConnection()
{
    try
    {
        using (var client = new WebClient())
        {
            using (var stream = client.OpenRead("http://www.google.com"))
            {
                return true;
            }
        }
    }
}

```

```

    }
}
catch
{
    return false;
}
}

```

```

public static string Sanitize(this string str)

```

```

{
    if (str == null)
        str = "";
    return str.Replace("\'", "'").Replace("''", "");
}

```

```

public static string SanitizeHTML(this string str)

```

```

{
    if (str == null)
        str = "";
    return str.Replace("<", "").Replace(">", "").Replace("*", "").Replace(":", "").Replace("|", "");
}

```

```

public static Asset FindAssetByNumber(this List<Asset> assets, string assetNumber)

```

```

{
    return (from x in assets
            where x.AssetNumber == assetNumber
            select x).FirstOrDefault();
}

```

```

public static Asset FindAssetByName(this List<Asset> assets, string assetName)

```

```

{
    return (from x in assets
            where x.AssetName == assetName
            select x).FirstOrDefault();
}

```

```

public static List<Asset> FindAssetsByNumber(this List<Asset> assets, string assetNumber)

```

```

{
    return (from x in assets
            where x.AssetNumber == assetNumber
            select x).ToList();
}

```

```

public static RoleBindinglist ToRoleBindingList(this IEnumerable<IdentityRole> list)

```

```

{

```

```

        return new RoleBindinglist(list.ToList());
    }
    public static UserBindinglist ToUserBindingList(this IEnumerable<ApplicationUser> list)
    {
        return new UserBindinglist(list.ToList());
    }
    public static AssetNoticeBindinglist ToNoticeBindingList(this IEnumerable<AssetNotification>
list)
    {
        return new AssetNoticeBindinglist(list.ToList());
    }
    public static Asset ToAsset(this XElement asset)
    {
        Asset a = new Asset();
        try
        {
            a.AssetName = asset.GetAttribute("Name").Sanitize();
        }
        catch { }
        try
        {
            a.AssetNumber = asset.SelectSingleNode("AssetNumber").InnerText.Sanitize();
        }
        catch { }
        try
        {
            a.DateRecieved =
DateTime.Parse(asset.SelectSingleNode("DateRecieved").InnerText.Sanitize());
        }
        catch { }
        try
        {
            a.DateShipped =
DateTime.Parse(asset.SelectSingleNode("DateShipped").InnerText.Sanitize());
        }
        catch { }
        try
        {
            a.Description = asset.SelectSingleNode("Description").InnerText.Sanitize();
        }
        catch { }
    }

```

```
try
{
    a.OrderNumber = asset.SelectSingleNode("OrderNumber").InnerText.Sanitize();
}
catch { }
try
{
    a.PackingSlip = asset.SelectSingleNode("PackingSlip").InnerText.Sanitize();
}
catch { }
try
{
    a.PersonShipping = asset.SelectSingleNode("PersonShipping").InnerText.Sanitize();
}
catch { }
try
{
    a.ReturnReport = asset.SelectSingleNode("ReturnReport").InnerText.Sanitize();
}
catch { }
try
{
    a.ServiceEngineer = asset.SelectSingleNode("ServiceEngineer").InnerText.Sanitize();
}
catch { }
try
{
    a.ShipTo = asset.SelectSingleNode("ShipTo").InnerText.Sanitize();
}
catch { }
try
{
    a.UpsLabel = asset.SelectSingleNode("UpsLabel").InnerText.Sanitize();
}
catch { }
try
{
    a.weight = Convert.ToDecimal(asset.SelectSingleNode("Weight").InnerText.Sanitize());
}
catch { }
try
{
```

```

        a.IsOut = Convert.ToBoolean(asset.SelectSingleNode("IsOut").InnerText.Sanitize());
    }
    catch { }
    try
    {
        a.OnHold =
Convert.ToBoolean(asset.SelectSingleNode("OnHold").InnerText.Sanitize());
    }
    catch { }
    try
    {
        a.IsDamaged =
Convert.ToBoolean(asset.SelectSingleNode("IsDamaged").InnerText.Sanitize());
    }
    catch { }
    try
    {
        XmlNodeList imglist = asset.GetElementsByTagName("Image");

        foreach (XmlElement el in imglist)
        {
            a.Images += "" + el.InnerText + ",";
        }
    }
    catch { }
    return a;
}
public static AssetNotification ToNotification(this XmlElement asset)
{
    AssetNotification a = new AssetNotification();
    try
    {
        a.AssetId = asset.GetAttribute("AssetId").Sanitize();
        a.AssetName = asset.GetAttribute("AssetName").Sanitize();
        a.IsNotified = Convert.ToBoolean(asset.GetAttribute("IsNotified").Sanitize());
        a.LastNotified = DateTime.Parse(asset.GetAttribute("LastNotified").Sanitize());
        a.Time = DateTime.Parse(asset.GetAttribute("Time").Sanitize());
        var n30 = asset.GetAttribute("Is30day").Sanitize();
        var n15 = asset.GetAttribute("Is15Day").Sanitize();
        a.Is30Day = Convert.ToBoolean(n30);
        a.Is15Day = Convert.ToBoolean(n15);
        XmlNodeList elemList = asset.GetElementsByTagName("Email");
        foreach (XmlElement elem in elemList)

```

```

        {
            EmailAddress em = new EmailAddress();
            em.Email = asset.GetAttribute("Email").Sanitize();
            em.Name = asset.GetAttribute("Name").Sanitize();
            a.Emails.Add(em);
        }
    }
    catch { }

```

```

        return a;
    }
    public static CustomerBindinglist ToCustomerBindingList(this IEnumerable<Customer> list)
    {
        return new CustomerBindinglist(list.ToList());
    }

```

```

    public static List<Asset> RemoveDuplicateAssets(this List<Asset> list)
    {
        list = list.GroupBy(x => x.AssetId).Select(x => x.First()).ToList();
        return list;
    }

```

```

//public static IEnumerable<IGrouping<TKey, TSource>> RemoveDuplicates<TSource,
//TKey>(this IEnumerable<TSource> source, Func<TSource, TKey> keySelector)
//{
//    return new IEnumerable<IGrouping<TKey, TSource>>();
//}

```

```

//LOGGING
    public static bool AddToLog(this HttpContext context, string msg)
    {
        try
        {
            var manager =
context.GetOwinContext().GetUserManager<ApplicationUserManager>();
            // Require the user to have a confirmed email before they can log on.
            var user = manager.FindByName(context.User.Identity.Name);
            var UserData = context.Session["SessionUserData"] as Data.UserData;
            UserData.Guid = user.Id;

```

```

UserData.Email = user.Email;
UserData.Name = user.UserName;
UserData.Log.Add(new Data.LogEntry(msg));
try
{
    var db = Web_App_Master.Pull.Setting(user.Id);
    if (db != null)
    {
        Data.UserData tt;
        context.Session["PersistingUserData"] = tt = new
Data.UserData().DeserializeFromXmlString<Data.UserData>(db.XmlData) as Data.UserData;
        tt.Log.Add(new Data.LogEntry(msg));

        Helpers.SettingsDBData d = new Helpers.SettingsDBData();
        d.Appname = user.Id;
        d.XmlData = UserData.SerializeToXmlString(tt);
        Push.Setting(d);
    }
    else
    {
        var nn = new Data.UserData();
        nn.Guid = user.Id;
        nn.Email = user.Email;
        nn.Name = user.UserName;
        nn.Log.Add(new Data.LogEntry(msg));
        context.Session["PersistingUserData"] = nn;
        Helpers.SettingsDBData d = new Helpers.SettingsDBData();
        d.Appname = user.Id;
        d.XmlData = UserData.SerializeToXmlString(nn);
        Push.Setting(d);
    }
}
catch
{
    try
    {
        var u = UserData.Clone() as Data.UserData;
        u.Log.Add(new Data.LogEntry(msg));
        context.Session["PersistingUserData"] = u;
    }
}

```



```

        Helpers.SettingsDBData db = new Helpers.SettingsDBData();
        db.Appname = user.Id;
        db.XmlData = UserData.SerializeToXmlString(u);
        Push.Setting(db);
    }
    catch { }
}
return true;
}
catch (Exception)
{

    return false;
}
}
public static void AddUserNotice(this Page page, string notice, string xml=null)
{
    var list = page.Session["Notifications"] as List<MenuNotice>;
    if (list==null)
    {
        list = new List<MenuNotice>();
    }
    MenuNotice n = new MenuNotice();
    n.Name = page.User.Identity.Name;
    n.Text = notice;
    if (xml !=null)
    {
        n.XmlData = xml;
    }
    list.Add(n);
    page.Session["Notifications"] = list;
}
}

}
namespace System.Collections.Generic
{
    public class SizedList<T> : ConcurrentQueue<T>
    {
        private readonly object syncObject = new object();

```

```

public int Size { get; private set; }

public SizedList(int size)
{
    Size = size;
}

public new void Enqueue(T obj)
{
    base.Enqueue(obj);
    lock (syncObject)
    {
        while (base.Count > Size)
        {
            T outObj;
            base.TryDequeue(out outObj);
        }
    }
}
}

```

```
using Helpers;
using Notification;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.IO;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Web;
using System.Xml;
using Web_App_Master.Account;
```

```
namespace Web_App_Master
```

```
{
    //MAKE ALL OF THESE ASYNC
    public class Push
    {
        public static bool Asset(Asset asset)
        {
            try
            {
                return AssetController.UpdateAsset(asset);
            }
            catch { return false; }
        }
        public static Task<bool> SettingAsync(SettingsDBData data)
        {
            return Task.Run(() =>
            {
                try
                {
                    return AssetController.PushSetting(data);
                }
                catch { return false; }
            });
        }
        public static Task<bool> GlobalsAsync()
        {
            return Task.Run(() =>
```

```

{
    try
    {
        return AssetController.PushSettings(Global.Library.Settings);

    }
    catch { return false; }
});
}
public static Task<bool> LibraryAsync()
{
    return Task.Run(() =>
    {
        try
        {
            SQL_Request req = new SQL_Request().OpenConnection();

            //request all assets
            req.GetAssets(false);

            //merge all assets

            //post merged assets as new master DB
            if (req.Tag != null)
            {
                var cloud = req.Tag as List<Asset>;
                //upload assets
                foreach (Asset a in Global.Library.Assets)
                {
                    try
                    {
                        var lookup = cloud.FindAssetByNumber(a.AssetNumber);
                        if (lookup == null)
                        {
                            req.AddAsset(a, false);

                        }
                        else
                        if (lookup.AssetNumber == a.AssetNumber)
                        {
                            req.UpdateAsset(a, false);
                        }
                    }
                    catch { }
                }
            }
        }
        catch { }
    });
}

```

```

        }
    }
    catch
    { //PopNotify("Error", "Error Pushing Library To SQL");
    }
}

}
req.CloseConnection();
return true;
}
catch { return false; }
});
}
public static Task<bool> NotificationAsync(NotificationSystem.Notice notice)
{
    return Task.Run(() =>
    {
        try
        {
            SettingsDBData db = new SettingsDBData();
            db.Appname = notice.Guid;
            db.XmlData = notice.SerializeToXmlString(notice);
            return AssetController.PushSetting(db);
        }
        catch { return false; }
    });
}
public static Task<bool> NotificationSystemAsync()
{
    return Task.Run(() =>
    {
        try
        {
            SettingsDBData db = new SettingsDBData();
            db.Appname = Global.NoticeSystem.Guid;
            db.XmlData = Global.NoticeSystem.SerializeToXmlString(Global.NoticeSystem);
            return AssetController.PushSetting(db);
        }
        catch { return false; }
    });
}

```

```
}
```

```
public static bool Setting(SettingsDBData data)
```

```
{
```

```
    try
```

```
    {
```

```
        return AssetController.PushSetting(data);
```

```
    }
```

```
    catch { return false; }
```

```
}
```

```
public static bool Globals()
```

```
{
```

```
    try
```

```
    {
```

```
        return AssetController.PushSettings(Global.Library.Settings);
```

```
    }
```

```
    catch { return false; }
```

```
}
```

```
public static bool Library()
```

```
{
```

```
    try
```

```
    {
```

```
        SQL_Request req = new SQL_Request().OpenConnection();
```

```
        //request all assets
```

```
        req.GetAssets(false);
```

```
        //merge all assets
```

```
        //post merged assets as new master DB
```

```
        if (req.Tag != null)
```

```
        {
```

```
            var cloud = req.Tag as List<Asset>;
```

```
            //upload assets
```

```
            foreach (Asset a in Global.Library.Assets)
```

```
            {
```

```
                try
```

```
                {
```

```
                    var lookup = cloud.FindAssetByNumber(a.AssetNumber);
```

```

        if (lookup == null)
        {
            req.AddAsset(a, false);

        }
        else
        if (lookup.AssetNumber == a.AssetNumber)
        {
            req.UpdateAsset(a, false);

        }
    }
    catch
    { //PopNotify("Error", "Error Pushing Library To SQL");
    }
}

}
req.CloseConnection();
return true;
}
catch
{
    return false;
}

}
public static bool Notification(NotificationSystem.Notice notice)
{
    try
    {
        SettingsDBData db = new SettingsDBData();
        db.Appname = notice.Guid;
        db.XmlData = notice.SerializeToXmlString(notice);
        return AssetController.PushSetting(db);
    }
    catch { return false; }
}
public static bool NotificationSystem()
{
    try

```

```

    {
        SettingsDBData db = new SettingsDBData();
        db.Appname = Global.NoticeSystem.Guid;
        db.XmlData = Global.NoticeSystem.SerializeToXmlString(Global.NoticeSystem);
        return AssetController.PushSetting(db);
    }
    catch {
        //Global.NoticeSystem.Notices = new Notification.NotificationSystem.NoticeBindinglist();
        return false; }

}

public static bool Transaction(PendingTransaction transaction)
{
    return AssetController.PushTransaction(transaction);
}

}

public class Pull
{
    public static Asset Asset(string num)
    {
        try
        {
            return AssetController.GetAsset(num);
        }
        catch { return null; }
    }

    public static List<Asset> Assets()
    {
        try
        {
            return AssetController.GetAllAssets().OrderBy(w => w.AssetNumber).ToList();
        }
        catch { return new List<Helpers.Asset>(); }
    }

    public static async Task<List<Asset>> AssetsAsync()
    {
        var a = await AssetController.GetAllAssetsAsync();
        return a;
    }

    public static SettingsDBData Setting(string guid)

```



```

{
    try
    {
        return AssetController.GetSetting(guid);
    }
    catch { return null; }
}
public static bool Globals()
{
    try
    {
        Global.Library.Settings = AssetController.GetSettings();
        return true;
    }
    catch
    {
        Global.Library.Settings = new Settings();
        return false;
    }
}
public static bool Library()
{
    try
    {
        Global.Library.Assets = AssetController.GetAllAssets();
        return true;
    }
    catch { Global.Library = new DataStore(); return false; }
}
public static bool NotificationSystem(string app_name= "AWP_Notification_System")
{
    try
    {
        var db = AssetController.GetSetting(app_name);
        Global.NoticeSystem = new
NotificationSystem().DeserializeFromXmlString<NotificationSystem>(db.XmlData);

        return true;
    }
    catch { Global.NoticeSystem = new NotificationSystem("AWP_Notification_System");
return false; }
}

```

```

public static NotificationSystem.Notice Notification(string guid)
{
    try
    {
        var db = AssetController.GetSetting(guid);
        var notice = new
NotificationSystem.Notice().DeserializeFromXmlString<NotificationSystem.Notice>(db.XmlData);
        return notice;
    }
    catch { return null; }
}
public static NotificationSystem.Notice Notification(NotificationSystem.Notice notice)
{
    try
    {

        var db = AssetController.GetSetting(notice.Guid);
        var n = new
NotificationSystem.Notice().DeserializeFromXmlString<NotificationSystem.Notice>(db.XmlData);
        return n;
    }
    catch { return null; }
}
public static List<PendingTransaction> Transactions(string AppName =
"AWP_STARRAG_US")
{
    return AssetController.GetAllTransactions(AppName);
}
public static PendingTransaction Transaction(string transactionID, string AppName =
"AWP_STARRAG_US")
{
    return AssetController.PullTransaction(transactionID, AppName);
}

}
public static class ExportLibrary
{
    // GO BACK AND GET THE REAL ONE R USE THE NEW XML ONE
    public static BackgroundWorker Worker;
    public static void Export(string filename="/library.xml")
    {

```

```

Worker = new BackgroundWorker();
Worker.DoWork += Worker_DoWork;
Worker.WorkerReportsProgress = true;
Worker.WorkerSupportsCancellation = true;
Global.Library.Name = filename;
Worker.RunWorkerAsync(Global.Library);

}
private static void Worker_DoWork(object sender, DoWorkEventArgs e)
{
    var worker = sender as BackgroundWorker;
    var assetslibrary = e.Argument as DataStore;
    var assets = assetslibrary.Assets;
    var length = assets.Count;
    XmlWriterSettings settings = new XmlWriterSettings();
    settings.Indent = true;

    StringBuilder builder = new StringBuilder();
    builder.Capacity = 5000;
    int i = 0;
    using (XmlWriter writer = XmlWriter.Create(builder, settings))
    {
        writer.WriteStartDocument();
        writer.WriteStartElement("AssetList");
        foreach (var s in assets)
        {
            if (e.Cancel == true)
            {
                return;
            }
            try
            {
                AddAssetToXml(writer, s);
            }
            catch { }
            ++i;
        }
        writer.WriteEndElement();

        writer.WriteEndDocument();
    }
    using (var sr = File.OpenWrite(assetslibrary.Name))

```

```

using (StreamWriter s = new StreamWriter(sr))
{
    builder.ToString();
    s.WriteLine(builder.ToString());
}
}

```

```

private static void AddAssetToXml(XmlWriter writer, Asset s)
{
    try
    {
        writer.WriteStartElement("Asset");
        try
        {
            writer.WriteAttributeString("Name", s.AssetName);
        }
        catch { }

        writer.WriteStartElement("AssetNumber");
        try
        {
            writer.WriteString(s.AssetName);
        }
        catch { }
        writer.WriteEndElement();

        writer.WriteStartElement("Description");

        try
        {
            if (s.Description.Contains("\n"))
            {
                s.Description = s.Description.Replace("\n", " ");
            }
            writer.WriteString(s.Description);
        }
        catch { }
        writer.WriteEndElement();

        writer.WriteStartElement("BarcodeImage");
        try
        {

```

```
        writer.WriteString(s.BarcodeImage);
    }
    catch { }
    writer.WriteEndElement();

    writer.WriteStartElement("Images");
    try
    {

        writer.WriteStartElement("Image");
        writer.WriteString(s.Images);
        writer.WriteEndElement();

    }
    catch (Exception ex) { }
    writer.WriteEndElement();

    writer.WriteStartElement("DateRecieved");
    try
    {
        writer.WriteString(s.DateRecieved.ToString());
    }
    catch { }
    writer.WriteEndElement();

    writer.WriteStartElement("DateShipped");
    try
    {
        writer.WriteString(s.DateShipped.ToString());
    }
    catch { }
    writer.WriteEndElement();

    writer.WriteStartElement("IsOut");
    try
    {
        writer.WriteString(s.IsOut.ToString());
    }
    catch { }
    writer.WriteEndElement();
```

```
writer.WriteStartElement("OrderNumber");  
try  
{  
    writer.WriteString(s.OrderNumber.ToString());  
}  
catch { }  
writer.WriteEndElement();
```

```
writer.WriteStartElement("PersonShipping");  
try  
{  
    writer.WriteString(s.PersonShipping);  
}  
catch { }  
writer.WriteEndElement();
```

```
writer.WriteStartElement("ServiceEngineer");  
try  
{  
    writer.WriteString(s.ServiceEngineer);  
}  
catch { }  
writer.WriteEndElement();
```

```
writer.WriteStartElement("ShipTo");  
try  
{  
    writer.WriteString(s.ShipTo);  
}  
catch { }  
writer.WriteEndElement();
```

```
writer.WriteStartElement("Weight");  
try  
{  
    writer.WriteString(s.weight.ToString());  
}  
catch { }  
writer.WriteEndElement();
```

```
writer.WriteStartElement("OnHold");  
try
```

```
{
    writer.WriteString(s.OnHold.ToString());
}
catch { }
writer.WriteEndElement();

writer.WriteStartElement("IsDamaged");
try
{
    writer.WriteString(s.IsDamaged.ToString());
}
catch { }
writer.WriteEndElement();

writer.WriteStartElement("ShippingInformation");

try
{
    writer.WriteStartElement("PackingSlip");
    writer.WriteString(s.PackingSlip);
    writer.WriteEndElement();
}
catch { }
try
{
    writer.WriteStartElement("UPSLabel");
    writer.WriteString(s.UpsLabel);
    writer.WriteEndElement();
}
catch { }
try
{
    writer.WriteStartElement("ReturnReport");
    writer.WriteString(s.ReturnReport);
    writer.WriteEndElement();
}
catch { }

writer.WriteEndElement();
```

```

        writer.WriteEndElement();
    }
    catch (Exception ex)
    {

    }
}

public static void AddCookie(string name, string value, int expireAfter=3, bool secure = true,
bool shareable=false)
{
    HttpCookie cookie = new HttpCookie(name);
    cookie.Secure = secure;
    cookie.Shareable = shareable;
    cookie.Expires = DateTime.Now.AddDays(expireAfter);
    cookie.Value = value;

    HttpContext.Current.Response.Cookies.Add(cookie);
}
public static HttpCookieCollection GetCookies()
{
    return HttpContext.Current.Request.Cookies;
}
public static HttpCookie GetCookie(string name)
{
    return HttpContext.Current.Request.Cookies[name];
}
}
}

```



```
using System;
using System.Collections.Generic;
using System.Drawing;
using System.Drawing.Imaging;
using System.IO;
using System.Linq;
using System.Web;
using Helpers;
using iTextSharp.text.pdf;
using Microsoft.AspNet.Identity.Owin;
using Microsoft.AspNet.Identity;
using System.Drawing.Drawing2D;
using Web_App_Master.Account;
using System.Text;
```

```
namespace Web_App_Master
```

```
{
    public class BarcodeHandler : IHttpHandler
    {
        //some stuff snipped
        public bool IsReusable
        {
            get { return false; }
        }
        public void SomeCheckImageHandler() { }

        public void ProcessRequest(HttpContext context)
        {
            string prodCode = context.Request.QueryString.Get("code");
            var w = context.Request.QueryString.Get("w");
            var h = context.Request.QueryString.Get("h");
            int width=320;
            int height=100;
            if (w!=null)
            {
                try { width = Convert.ToInt32(w); } catch { width = 320; }
            }
            if (h != null)
            {
                try { height = Convert.ToInt32(h); } catch { height = 100; }
            }
        }
    }
}
```

```

        var filename =
context.Request.Url.Segments.Last().Replace(".barcode","").Replace(".Barcode",
 "").Replace(".BARCODE", "");
        context.Response.ContentType = "image/png";
        if (prodCode.Length > 0)
        {
            if (prodCode.Length > Global.Library.Settings.AssetNumberLength) prodCode =
prodCode.Substring(0, Global.Library.Settings.AssetNumberLength);
            Barcode128 code128 = new Barcode128();
            code128.CodeType = Barcode.CODE128;
            code128.ChecksumText = true;
            code128.GenerateChecksum = true;
            code128.StartStopText = true;
            code128.Code = prodCode;
            System.Drawing.Bitmap bm = new
System.Drawing.Bitmap(code128.CreateDrawingImage(System.Drawing.Color.Black,
System.Drawing.Color.White));
            Bitmap resized = new Bitmap(bm, new Size(width,height));

            resized.Save(context.Response.OutputStream,
System.Drawing.Imaging.ImageFormat.Png);
        }

    }
}

public class ResizeHandler : IHttpHandler
{
    //some stuff snipped
    public bool IsReusable
    {
        get { return false; }
    }
    public void SomeCheckImageHandler() { }

    public void ProcessRequest(HttpContext context)
    {
        string file = "~/Images/transparent.png";
        //var file = Encoding.UTF8.GetString(
HttpServerUtility.UrlTokenDecode(context.Request["f"]));
        var height = Convert.ToInt32( context.Request["h"]);
        var width = Convert.ToInt32( context.Request["w"]);
        file = context.Request.Path.Replace(".r", "");

```

```

file = HttpContext.Current.Server.MapPath(file);
context.Response.ContentType = "image/png";
try
{
    Image img = null;
    try
    {
        img = Image.FromFile(file);
    }
    catch {
        try
        {
            img =
Image.FromFile(HttpContext.Current.Server.MapPath("/Images/transparent.png"));
        }
        catch
        {
            img = new Bitmap(10, 10);
        }
    }
    Image _img = new Bitmap(width, height);

    Graphics graphics = Graphics.FromImage(_img);

    //Resize picture according to size

    graphics.DrawImage(img, 0, 0, width, height);

    graphics.CompositingQuality =
System.Drawing.Drawing2D.CompositingQuality.HighSpeed;

    graphics.Dispose();

    img.Dispose();

    //Create output stream

    MemoryStream str = new MemoryStream();

```

```
_img = _img.GetThumbnailImage(width, height, null, IntPtr.Zero);

_img.Save(str, System.Drawing.Imaging.ImageFormat.Png);

_img.Dispose();

str.WriteTo(context.Response.OutputStream);

str.Dispose();

str.Close();

//Set response type

context.Response.ContentType = ".jpg";

context.Response.End();
```

```
    }
    catch { }
}
}
```

```
public class ImageHandler : IHttpHandler
{
    //some stuff snipped
    public bool IsReusable
    {
        get { return false; }
    }
    public void SomeCheckImageHandler() { }

    public void ProcessRequest(HttpContext context)
    {
```

```

var id = context.Request["Id"];
context.Response.ContentType = "image/png";
var idx = context.Request["idx"];
if (idx != null)
{
    var apth = context.Server.MapPath(id + "/");
    if (Directory.Exists(apth))
    {
        string[] filePaths = System.IO.Directory.GetFiles(apth);
        List<Image> files = new List<Image>();
        foreach (string filePath in filePaths)
        {
            string fileName = System.IO.Path.GetFileName(filePath);
            var ext = System.IO.Path.GetExtension(fileName).ToUpper();
            if (ext.Equals(".PNG"))
            {
                files.Add(Image.FromFile(filePath));
            }
        }
        files[Convert.ToInt32(idx)].Save(context.Response.OutputStream, ImageFormat.Png);
    }
}

}

}

public class PdfHandler : IHttpHandler
{
    //some stuff snipped
    public bool IsReusable
    {
        get { return false; }
    }
    public void SomeCheckImageHandler() { }

    public void ProcessRequest(HttpContext context)
    {
        var path = context.Request["path"];

        Byte[] buffer = File.ReadAllBytes(path);

```

```

        if (buffer != null)
        {
            context.Response.ContentType = "application/pdf";
            context.Response.AddHeader("content-length", buffer.Length.ToString());
            context.Response.BinaryWrite(buffer);
        }
    }
}

public class PdfLabelHandler : IHttpHandler
{
    //some stuff snipped
    public bool IsReusable
    {
        get { return false; }
    }

    public void ReturnBlank(HttpContext context)
    {
        var buffer = new Pdf.PdfTemplate().TemplateBytes;
        context.Response.ContentType = "application/pdf";
        context.Response.AddHeader("content-length", buffer.Length.ToString());
        context.Response.BinaryWrite(buffer);
    }

    public void ReturnPdf(HttpContext context, byte[] buffer)
    {
        context.Response.ContentType = "application/pdf";
        context.Response.AddHeader("content-length", buffer.Length.ToString());
        context.Response.BinaryWrite(buffer);
    }

    private void DrawRotatedTextAt(Graphics gr, float angle, string txt, int x, int y, Font the_font,
Brush the_brush)
    {
        // Save the graphics state.
        GraphicsState state = gr.Save();
        gr.ResetTransform();

        // Rotate.
        gr.RotateTransform(angle);

        // Translate to desired position. Be sure to append

```

```

// the rotation so it occurs after the rotation.
gr.TranslateTransform(x, y, MatrixOrder.Append);

// Draw the text at the origin.
gr.DrawString(txt, the_font, the_brush, 0, 0);

// Restore the graphics state.
gr.Restore(state);
}
public void ProcessRequest(HttpContext context)
{
    string id = Path.GetFileNameWithoutExtension(context.Request.Path);
    try
    {

        if (id != null)
        {
            if (id == "")
            {
                //context.Application["CurrentPrintList" + id] = null;
                ReturnBlank(context);
                return;
            }
            object obj = null;
            try
            {
                obj = context.Application["CurrentPrintList" + id];
            }
            catch { }

            if (obj != null)
            {
                if (obj is List<string>)
                {
                    var list = obj as List<string>;
                    if (list.Count > 0)
                    {
                        List<byte[]> Buffers = new List<byte[]>();
                        var pages = list.SplitList(30);

                        foreach (var page in pages)

```

```

{
    //create images
    List<Image> images = new List<Image>();
    foreach (var code in page)
    {
        try
        {
            var asset = Global.Library.Assets.FindAssetByNumber(code);
            Bitmap bmp = new Bitmap(263, 100);
            RectangleF rectf = new RectangleF(10, 40, 248, 60);
            Graphics g = Graphics.FromImage(bmp);

            g.SmoothingMode = SmoothingMode.AntiAlias;
            g.InterpolationMode = InterpolationMode.HighQualityBicubic;
            g.PixelOffsetMode = PixelOffsetMode.HighQuality;
            g.FillRectangle(Brushes.White, new System.Drawing.Rectangle(0, 0,
263, 100));

            //DrawRotatedTextAt(g, 90, asset.AssetName, 0, 0, new
System.Drawing.Font("Tahoma", 21, FontStyle.Bold), Brushes.Black);
            g.DrawString(asset.AssetName, new System.Drawing.Font("Tahoma",
11, FontStyle.Regular), Brushes.Black, rectf);
            g.DrawString(asset.AssetName, new
System.Drawing.Font("Tahoma", 23, FontStyle.Bold), Brushes.Black, 2,2);
            g.ResetTransform();
            g.Flush();
            images.Add(bmp);
        }
        catch {
            ReturnBlank(context);
            return;
        }
    }

    Pdf.Avery5160GraphicLabels label = new Pdf.Avery5160GraphicLabels();
    label.Images = images;
    using (MemoryStream ms = new MemoryStream())
    {
        var result = label.Create(ms);
        if (!result)
        {
            ReturnBlank(context);
            return;
        }
    }
}

```



```

        }
        //ms.Position = 0;
        byte[] buffer = ms.ToArray();
        Buffers.Add(buffer);

    }
}
if (Buffers.Count == 0)
{
    //context.Application["CurrentPrintList" + id] = null;
    ReturnBlank(context);
    return;
}
using (MemoryStream ms = new MemoryStream())
{
    Pdf.CombineMultiplePDFs(Buffers, ms);
    //ms.Position = 0;
    ReturnPdf(context, ms.ToArray());
    return;
}

}
else
{
    //context.Application["CurrentPrintList" + id] = null;
    ReturnBlank(context);
    return;
}
}
else
{
    // context.Application["CurrentPrintList" + id] = null;
    ReturnBlank(context);
    return;
}
}
else
{
    //context.Application["CurrentPrintList" + id] = null;
    ReturnBlank(context);
    return;
}
}

```

```

    }
    else
    {
        // context.Application["CurrentPrintList" + id] = null;
        ReturnBlank(context);
        return;
    }
}
catch
{
    // context.Application["CurrentPrintList" + id] = null;
    ReturnBlank(context);
    return;
}
}
}

public class PdfBarcodeHandler : IHttpHandler
{
    //some stuff snipped
    public bool IsReusable
    {
        get { return false; }
    }

    public void ReturnBlank(HttpContext context)
    {
        var buffer = new Pdf.PdfTemplate().TemplateBytes;
        context.Response.ContentType = "application/pdf";
        context.Response.AddHeader("content-length", buffer.Length.ToString());
        context.Response.BinaryWrite(buffer);
    }

    public void ReturnPdf(HttpContext context, byte[] buffer)
    {
        context.Response.ContentType = "application/pdf";
        context.Response.AddHeader("content-length", buffer.Length.ToString());
        context.Response.BinaryWrite(buffer);
    }

    public void ProcessRequest(HttpContext context)
    {
        string id = Path.GetFileNameWithoutExtension(context.Request.Path);
        try

```

```
{
```

```
if (id != null)
{
    if (id == "")
    {
        //context.Application["CurrentPrintList" + id] = null;
        ReturnBlank(context);
        return;
    }
    object obj = null;
    try
    {
        obj = context.Application["CurrentPrintList" + id];
    }
    catch { }

    if (obj != null)
    {
        if (obj is List<string>)
        {
            var list = obj as List<string>;
            if (list.Count > 0)
            {
                List<byte[]> Buffers = new List<byte[]>();
                var pages = list.SplitList(30);

                foreach (var page in pages)
                {
                    Pdf.Avery5160BarcodeLabels label = new Pdf.Avery5160BarcodeLabels();
                    label.Codes = page;
                    using (MemoryStream ms = new MemoryStream())
                    {
                        var result = label.Create(ms);
                        if (!result)
                        {
                            ReturnBlank(context);
                            return;
                        }
                    }
                    //ms.Position = 0;
                }
            }
        }
    }
}
```

```

        byte[] buffer = ms.ToArray();
        Buffers.Add(buffer);

    }
}
if (Buffers.Count == 0)
{
    //context.Application["CurrentPrintList" + id] = null;
    ReturnBlank(context);
    return;
}
using (MemoryStream ms = new MemoryStream())
{
    Pdf.CombineMultiplePDFs(Buffers, ms);
    //ms.Position = 0;
    ReturnPdf(context, ms.ToArray());
    return;
}

}
else
{
    //context.Application["CurrentPrintList" + id] = null;
    ReturnBlank(context);
    return;
}
}
else
{
    // context.Application["CurrentPrintList" + id] = null;
    ReturnBlank(context);
    return;
}
}
else
{
    // context.Application["CurrentPrintList" + id] = null;
    ReturnBlank(context);
    return;
}
}
else

```

```
{
    // context.Application["CurrentPrintList" + id] = null;
    ReturnBlank(context);
    return;
}
}
catch
{
    //context.Application["CurrentPrintList" + id] = null;
    ReturnBlank(context);
    return;
}

}

}
```

```

using iTextSharp.text.pdf;
using Microsoft.AspNet.Identity.EntityFramework;
using Microsoft.AspNet.Identity.Owin;
using ShippingAPI;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.IO;
using System.Linq;
using System.Net;
using System.Runtime.Serialization;
using System.Runtime.Serialization.Formatters.Binary;
using System.Text.RegularExpressions;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Xml;
using System.Xml.Serialization;
using Web_App_Master;
using Web_App_Master.Models;
using static iTextSharp.text.pdf.AcroFields;

```

namespace Helpers

```

{

    public class CustomerBindinglist : BindingList<Customer>
    {
        public CustomerBindinglist() { this.AllowEdit = AllowNew = AllowRemove = true; }
        public CustomerBindinglist(List<Customer> notices)
        {
            this.AllowEdit = AllowNew = AllowRemove = true;
            foreach (var notice in notices)
            {
                this.Add(notice);
            }
        }
    }

    public class AssetNoticeBindinglist : BindingList<AssetNotification>
    {
        public AssetNoticeBindinglist() { this.AllowEdit = AllowNew = AllowRemove = true; }
        public AssetNoticeBindinglist(List<AssetNotification> notices)
    }
}

```

```

{
    this.AllowEdit = AllowNew = AllowRemove = true;
    foreach (var notice in notices)
    {
        this.Add(notice);
    }
}
}

```

```

public class EmailBindinglist : BindingList<EmailAddress>
{
    public EmailBindinglist() { this.AllowEdit = AllowNew = AllowRemove = true; }
    public EmailBindinglist(List<EmailAddress> emails)
    {
        this.AllowEdit = AllowNew = AllowRemove = true;
        foreach (var email in emails)
        {
            this.Add(email);
        }
    }
}

```

```

public class UserBindinglist : BindingList<BindableUserData>
{
    public UserBindinglist() { }
    public UserBindinglist(List<ApplicationUser> users)
    {
        foreach(var user in users)
        {
            this.Add(new BindableUserData() { IdentityUserRole = new IdentityUserRole() { UserId =
user.Id, RoleId = user.Email } });
        }
    }
}

```

```

public class RoleBindinglist: BindingList<BindableRoleData>
{
    public RoleBindinglist()
    {
        this.AddingNew += RoleBindinglist_AddingNew;
    }
    public RoleBindinglist(List<IdentityRole> roles)

```

```

{
    this.AddingNew += RoleBindinglist_AddingNew;
    foreach(var role in roles)
    {
        this.Add(new BindableRoleData( role));
    }
}

private void RoleBindinglist_AddingNew(object sender, AddingNewEventArgs e)
{
    var data = e.NewObject as BindableRoleData;
    data.RoleIndex = (sender as RoleBindinglist).Count.ToString();
}
}

public class BindableRoleData
{
    public BindableRoleData() { }
    public BindableRoleData(IdentityRole role)
    {
        this.iRole = role;
    }
    public IdentityRole iRole { get; set; }
    public string RoleName { get
    {
        return iRole.Name;
    }
    }
    public string RoleIndex { get; set; }
    public List<BindableUserData> RoleUsers {
        get
        {
            List<BindableUserData> users = new List<BindableUserData>();

            foreach(var user in iRole.Users)
            {
                BindableUserData ud = new BindableUserData(user);
                users.Add(ud);
            }
            return users;
        }
    }
}

```



```

    }

    public class BindableUserData
    {
        public BindableUserData()
        {

        }

        public BindableUserData( IdentityUserRole iur)
        {
            this.IdentityUserRole = iur;
        }

        public IdentityUserRole IdentityUserRole { get; set; }
        public string UserId { get
            {
                var manager =
HttpContext.Current.GetOwinContext().GetUserManager<ApplicationUserManager>();
                return (from user in manager.Users where user.Id==this.IdentityUserRole.UserId select
user ).FirstOrDefault().Id;
            }
        }

        public string UserName
        {
            get
            {
                var manager =
HttpContext.Current.GetOwinContext().GetUserManager<ApplicationUserManager>();
                return (from user in manager.Users where user.Id == this.IdentityUserRole.UserId select
user).FirstOrDefault().Email;
            }
        }

        public string RoleId { get { return this.IdentityUserRole.RoleId; } }
    }

```

```

public class CheckOutData
{
    public UPSCode ParseUpsCodeString(string code)
    {
        try
        {

```

```

        var a = code.Split('-')[0];
        var ret = (UPScore)Enum.Parse(typeof(UPScore), a);
        return ret;
    }
    catch { return UPScore.Ground; }
}

public Address From = new Address();
public Address To = new Address();
public Address Shipper = new Address();
public UPScore Code = UPScore.Ground;
public Package Package = new Package();
public DateTime Pickup = DateTime.Now;
public DateTime Arrival = DateTime.Now.AddDays(3);
public string Reference1 = "";
public string Reference2 = "";
public List<Asset> CheckOutItems;
}

[Serializable]
public class SqlSetting: Serializers.XSerializer<SqlSetting>
{
    public SqlSetting()
    {
        this.XmlData = Type = "";
        this.Guid = System.Guid.NewGuid().ToString();
    }
    public SqlSetting(System.Guid guid)
    {
        this.XmlData = Type = "";
        this.Guid = guid.ToString();
    }
    [XmlElement]
    public string XmlData { get; set; }
    [XmlElement]
    public string Type { get; set; }
    [XmlElement]
    public string Guid { get; set; }
}

[Serializable]
public class MenuNotice : Serializers.XSerializer<MenuNotice>
{
    public MenuNotice()
    {

```

```

        this.XmlData = Text = Name = "";
        this.Guid = System.Guid.NewGuid().ToString();
    }
    public MenuNotice(string name)
    {
        Name = name;
        this.XmlData = Text = "";
        this.Guid = System.Guid.NewGuid().ToString();
    }
    public MenuNotice(string name, string text)
    {
        Name = name;
        Text = text;
        this.XmlData = "";
        this.Guid = System.Guid.NewGuid().ToString();
    }
    public MenuNotice(string name, string text, string xml)
    {
        Name = name;
        Text = text;
        this.XmlData = xml;
        this.Guid = System.Guid.NewGuid().ToString();
    }
    public MenuNotice(System.Guid guid)
    {
        this.XmlData = Text = Name = "";
        this.Guid = guid.ToString();
    }
    [XmlElement]
    public DateTime Created = DateTime.Now;
    [XmlElement]
    public string XmlData { get; set; }
    [XmlElement]
    public string Name { get; set; }
    [XmlElement]
    public string Guid { get; set; }
    [XmlElement]
    public string Text { get; set; }
}
[Serializable]
public class TransationDBData : SettingsDBData
{

```

```

public TransationDBData(PendingTransaction transaction)
{
    this.XmlData = transaction.SerializeToXmlString(transaction);
    this.TransactionID = transaction.TransactionID;
    this.Appname = "AWP_STARRAG_US";
}
public TransationDBData()
{
    Appname = XmlData = TransactionID = "";
}
[XmlElement]
public string TransactionID { get; set; }

}
[Serializable]
public class SettingsDBData:Serializers.XSerializer<SettingsDBData>
{
    public SettingsDBData()
    {
        Appname = XmlData = XmlData2 = XmlData3 = XmlData4 = XmlData5="";
    }
    [XmlElement]
    public string Appname { get; set; }
    [XmlElement]
    public string XmlData { get; set; }
    [XmlElement]
    public string XmlData2 { get; set; }
    [XmlElement]
    public string XmlData3 { get; set; }
    [XmlElement]
    public string XmlData4 { get; set; }
    [XmlElement]
    public string XmlData5 { get; set; }
}
[Serializable]//xml
public class CalibrationLibrary : Serializers.XSerializer<CalibrationLibrary>
{
    [XmlElement("Calibrations")]
    public List<CalibrationData> Calibrations = new List<CalibrationData>();
    [XmlIgnore]
    public object Tag;
}

```

```
}
```

```
[Serializable]//xml
```

```
public class CalibrationData : Serializers.XSerializer<CalibrationData>, ICloneable
```

```
{
```

```
    public object Clone()
```

```
    {
```

```
        return this.MemberwiseClone();
```

```
    }
```

```
    public CalibrationData()
```

```
    {
```

```
        LastCalibration = DateTime.Now;
```

```
        AssetNumber = "";
```

```
        m_SchedulePeriod = "12";
```

```
        CalibrationCompany = "";
```

```
        m_IsCalibrated = true;
```

```
        ImagePath = "";
```

```
        History = new List<string>();
```

```
        ScheduledCalibrationDate =
```

```
LastCalibration.AddMonths(Convert.ToInt32(m_SchedulePeriod));
```

```
    }
```

```
    public bool CheckCalibration()
```

```
    {
```

```
        int result = DateTime.Compare(ScheduledCalibrationDate, DateTime.Now);
```

```
        int result2 = DateTime.Compare(DateTime.Now, ScheduledCalibrationDate);
```

```
        DaysUntilNextCalibration = (ScheduledCalibrationDate - DateTime.Now).TotalDays;
```

```
        if (DaysUntilNextCalibration <= 0)
```

```
        {
```

```
            return false;
```

```
        }
```

```
        else
```

```
        {
```

```
            return true;
```

```
        }
```

```
    }
```

```
    [XmlElement("LastCalibration")]
```

```
    public DateTime LastCalibration { get; set; }
```

```
    [XmlElement("ScheduledCalibrationDate")]
```

```
    public DateTime ScheduledCalibrationDate { get; set; }
```

```

[XmlAttribute("AssetNumber")]
public string AssetNumber { get; set; }
[XmlElement]
public string m_guid = System.Guid.NewGuid().ToString();
[XmlElement]
public string Guid { get { return m_guid; } set { m_guid = value; } }
[XmlIgnore]
private string m_SchedulePeriod = "12";
[XmlElement("SchedulePeriod")]
public string SchedulePeriod
{
    get { return m_SchedulePeriod; }
    set
    {
        m_SchedulePeriod = value;
        ScheduledCalibrationDate =
LastCalibration.AddMonths(Convert.ToInt32(m_SchedulePeriod));
        CheckCalibration();
    }
}
[XmlElement("CalibrationCompany")]
public string CalibrationCompany { get; set; }
[XmlIgnore]
public bool m_IsCalibrated = true;
[XmlElement("IsCalibrated")]
public bool IsCalibrated
{
    get { return CheckCalibration(); }
    set { m_IsCalibrated = value; }
}
[XmlIgnore]
public string DaysLeft { get { return DaysUntilNextCalibration.ToString(); } }
[XmlIgnore]
public object Tag;
[XmlIgnore]
public object Tag2;
[XmlIgnore]
private double m_days;
[XmlElement("DaysUntilNextCalibration")]
public double DaysUntilNextCalibration
{
    get

```

```

{
    m_days = (ScheduledCalibrationDate - DateTime.Now).TotalDays;
    if (m_days <= 0)
    {
        IsCalibrated = false;
    }
    else
    {
        IsCalibrated = true;
    }

    return m_days;
}
set { m_days = value; }
}
[XmlElement("ImagePath")]
public string ImagePath { get; set; }
[XmlElement("History")]
public List<string> History { get; set; }

}

```

```

[Serializable]//xml
public class UPSaccount:Serializers.XSerializer<UPSaccount>
{
    public void Write(string path)
    {
        try
        {
            IFormatter formatter = new BinaryFormatter();
            using (Stream stream = new FileStream(path, FileMode.Create, FileAccess.Write,
FileShare.None))
            {
                formatter.Serialize(stream, this);
            }
        }
        catch { }
    }
    public static UPSaccount Read(string path)
    {
        UPSaccount obj = new UPSaccount();
        try

```

```

{
    IFormatter formatter = new BinaryFormatter();
    Stream stream = new FileStream(path,
                                   FileMode.Open,
                                   FileAccess.Read,
                                   FileShare.Read);
    obj = (UPSaccount)formatter.Deserialize(stream);
    stream.Close();
}
catch { }
return obj;
}
[XmlElement("P")]
public string P;
[XmlElement("A")]
public string A;
[XmlElement("I")]
public string I;
[XmlElement("N")]
public string N;
}

```

[Serializable]//xml

public class Email:Serializers.XSerializer<Email>

```

{
    [XmlElement("Address")]
    public EmailAddress Address;
    [XmlElement("Subject")]
    public string Subject;
    [XmlElement("Body")]
    public string Body;
    [XmlElement("Attachments")]
    public object Attachments;
    [XmlElement("Time")]
    public DateTime Time;
}

```

[Serializable]//xml

public class EmailAddress:Serializers.XSerializer<EmailAddress>

```

{
    public static EmailAddress Create(string email)
    {

```



```

EmailAddress e = new EmailAddress();
e.Email = email;
if (!e.Email.Contains("@"))
{
    e.Email = "servicetools.starrag@gmail.com";
}
try
{
    if (email.Contains("@"))
    {
        var temp = email.Split('@');
        e.Name = temp[0];
        if (temp[0].Contains("."))
        {
            e.Name = temp[0].Replace(".", " ");
        }
    }
}
catch
{
    e.Name = e.Email;
}

return e;
}
[XmlElement("Name")]
public string Name { get; set; }
[XmlElement("Email")]
public string Email { get; set; }
}

[Serializable]//xml
public class Customer:Serializers.XSerializer<Customer>
{
    public Customer()
    {
        CompanyName = "";
        Attn = "";
        Address = "";
        Address2 = "";
        Address3 = "";
    }
}

```

```

    City = "";
    State = "";
    NickName = "";
    ResInd = "";
    LocID = "";
    ConsInd = "";
    AccountNumber = "";
    AccPostalCd = "";
    USPSPOBoxIND = "";
    Postal = "";
    Country = "";
    PackageWeight = "";
    Phone = "";
    Fax = "";
    EmailAddress = new EmailAddress();
    OrderNumbers = new List<string>();
    OrderNumber = "";
    CurrentAssignedAssets = new List<string>();
    Tag = null;

}

public static Customer Create(string coName, string attn, string address, string phone, string
fax, EmailAddress email)
{
    Customer c = new Customer();
    c.CompanyName = coName;
    c.Attn = attn;
    c.Address = address;
    c.Phone = phone;
    c.Fax = fax;
    c.EmailAddress = email;
    return c;
}

[XmlElement("CompanyName")]
public string CompanyName { get; set; }
[XmlElement("Attn")]
public string Attn { get; set; }
[XmlElement("Address")]
public string Address { get; set; }
[XmlElement("Address2")]
public string Address2 { get; set; }
[XmlElement("Address3")]

```

```

public string Address3 { get; set; }
[XmlElement("City")]
public string City { get; set; }
[XmlElement("State")]
public string State { get; set; }
[XmlElement("NickName")]
public string NickName { get; set; }
[XmlElement("ResInd")]
public string ResInd { get; set; }
[XmlElement("LocID")]
public string LocID { get; set; }
[XmlElement("ConsInd")]
public string ConsInd { get; set; }
[XmlElement("AccountNumber")]
public string AccountNumber { get; set; }
[XmlElement("AccPostalCd")]
public string AccPostalCd { get; set; }
[XmlElement("USPSPOBoxIND")]
public string USPSPOBoxIND { get; set; }
[XmlElement("Postal")]
public string Postal { get; set; }
[XmlElement("Country")]
public string Country { get; set; }
[XmlElement("PackageWeight")]
public string PackageWeight { get; set; }
[XmlElement("Phone")]
public string Phone { get; set; }
[XmlElement("Fax")]
public string Fax { get; set; }
[XmlElement("EmailAddress")]
public EmailAddress EmailAddress { get; set; }
[XmlElement("Email")]
public string Email { get { return this.EmailAddress.Email; } }
[XmlElement("OrderNumbers")]
public List<string> OrderNumbers { get; set; }
[XmlElement("OrderNumber")]
public string OrderNumber { get; set; }
[XmlElement("CurrentAssignedAssets")]
public List<string> CurrentAssignedAssets { get; set; }
[XmlIgnore]
public object Tag { get; set; }
}

```

[Serializable]//xml

public class Engineer:Serializers.XSerializer<Engineer>

```
{
    public Engineer()
    {}
    public static Engineer Create(string name, string address, string phone, EmailAddress email)
    {
        Engineer e = new Engineer();
        e.Name = name;
        e.Address = address;
        e.Phone = phone;
        e.Email = email;
        return e;
    }
    [XmlElement("Name")]
    public string Name = "";
    [XmlElement("Address")]
    public string Address = "";
    [XmlElement("Phone")]
    public string Phone = "";
    [XmlElement("Email")]
    public EmailAddress Email = new EmailAddress();
    [XmlIgnore]
    public object Tag = null;
}
```

[Serializable]//xml

public class NotificationLibrary:Serializers.XSerializer<NotificationLibrary>

```
{
    public NotificationLibrary()
    {

    }
    [XmlElement("ThrityDayNotifications")]
    public List<AssetNotification> ThrityDayNotifications = new List<AssetNotification>();
    [XmlElement("FifteenDayNotifications")]
    public List<AssetNotification> FifteenDayNotifications = new List<AssetNotification>();
}
```

[Serializable]//xml

public class Backup:Serializers.XSerializer<Backup>

```

{
    public Backup()
    {
    }
    [XmlAttribute("LastBackup")]
    public DateTime LastBackup;
    [XmlElement("Tag")]
    public object Tag;
    [XmlElement("Tag2")]
    public object Tag2;
}

```

```

[Serializable]//xml
public class AssetNotification:Serializers.XSerializer<AssetNotification>
{
    public static AssetNotification Create(string assetNumber, List<EmailAddress> emails, bool
isNotified = false)
    {
        AssetNotification n = new AssetNotification();
        n.Emails = new List<EmailAddress>();
        n.Emails.AddRange(emails);
        n.AssetIdNumber = assetNumber;
        n.Time = DateTime.Now;
        n.LastNotified = DateTime.Now;
        n.IsNotified = isNotified;
        return n;
    }
    public AssetNotification()
    {
        AssetNumber = "";
        Emails = new List<EmailAddress>();
    }
    [XmlElement("Emails")]
    public List<EmailAddress> Emails { get; set; }
    [XmlElement("AssetNumber")]
    public string AssetNumber { get; set; }
    [XmlElement("Time")]
    public DateTime Time { get; set; }
    [XmlElement("LastNotified")]
    public DateTime LastNotified { get; set; }
    [XmlElement("IsNotified")]
    public bool IsNotified { get; set; }
}

```

```

[XmlElement("Is30Day")]
public bool Is30Day { get; set; }
[XmlElement("Is15Day")]
public bool Is15Day { get; set; }
[XmlIgnore]
public object Tag { get; set; }
[XmlElement("EmailsCSV")]
public string EmailsCSV { get {
    string ret = "";
    foreach(var email in Emails)
    {
        ret += email.Name + ", ";
    }
    return ret;
}
}
}

```

```

[Serializable]//xml
public class Settings: Serializers.XSerializer<Settings>, ICloneable
{
    public static Settings GetInstance()
    {
        return new Settings();
    }
    public Settings()
    {
    }
    public Settings Deserialize(string xml)
    {
        return this.DeserializeFromXmlString<Settings>(xml);
    }
    public string Serialize()
    {
        return this.SerializeToXmlString(this);
    }

    public object Clone()
    {
    }
}

```

```
    return this.MemberwiseClone();  
}
```

```
[XmlAttribute("LibraryPath")]  
public string LibraryPath;
```

```
[XmlAttribute("AssetsPath")]  
public string AssetsPath;
```

```
[XmlAttribute("TimeStamp")]  
public string TimeStamp = DateTime.Now.ToString();
```

```
[XmlElement("Notifications_30_Day")]  
public List<AssetNotification> Notifications_30_Day = new List<AssetNotification>();
```

```
[XmlElement("Notifications_15_Day")]  
public List<AssetNotification> Notifications_15_Day = new List<AssetNotification>();
```

```
[XmlElement("NotSentEmails")]  
public List<Email> NotSentEmails = new List<Email>();
```

```
[XmlElement("ServiceEngineers")]  
public List<EmailAddress> ServiceEngineers = new List<EmailAddress>();
```

```
[XmlElement("Customers")]  
public List<Customer> Customers;
```

```
[XmlElement("StaticEmails")]  
public List<EmailAddress> StaticEmails;
```

```
[XmlElement("ShippingPersons")]  
public List<EmailAddress> ShippingPersons;
```

```
[XmlElement("CheckOutMessage")]  
public string CheckOutMessage = "";
```

```
[XmlElement("CheckInMessage")]  
public string CheckInMessage = "";
```

```
[XmlElement("NotificationMessage")]  
public string NotificationMessage = "";
```

```
[XmlElement("ShipperNotification")]
public string ShipperNotification = "";
```

```
[XmlElement("UpsAccount")]
public UPSaccount UpsAccount = new UPSaccount();
```

```
[XmlElement]
public bool TESTMODE = false;
```

```
[XmlElement]
public int AssetNumberLength = 4;
```

```
[XmlElement]
public string SenderEmail = "provider.service.secure@gmail.com";
```

```
[XmlElement]
public string SenderPassword = "@Service1";
```

```
[XmlElement]
public string SenderHost = "smtp.gmail.com";
```

```
[XmlElement]
public int SenderPort = 587;
```

```
[XmlElement]
public string CalibrationMessage = @"Hello <name>,<NL><NL>
    Calibrated Tool, <serviceTool>, is out of calibration <NL>
    Please send tool for calibration.<NL><NL>
    Kind Regards,<NL><NL>
    Support Team
    ";
```

```
}
```

```
[Serializable]//xml
```

```
public class Asset : Serializers.XSerializer<Asset>
```

```
{
    public Asset Deserialize(string xml)
    {
        return this.DeserializeFromXmlString<Asset>(xml);
    }
    public string Serialize()
    {
        return this.SerializeToXmlString(this);
    }
}
```

```
public static Asset Create(string assetName = "", string assetNumber = "")
{
```



```

    Asset ass = new Asset().Init();
    ass.AssetName = assetName;
    ass.AssetNumber = assetNumber;
    return ass;
}
public object Clone()
{
    return this.MemberwiseClone();
}
public Asset()
{
    Init();
}
private Asset Init()
{
    Images = "";
    IsOut = false;
    weight = 0;
    History = new AssetHistory();
    AssetNumber = "0000";
    OrderNumber = "0";
    AssetName = PackingSlip = UpsLabel = ReturnReport = ShipTo = ServiceEngineer =
PersonShipping = Barcode = Description = BarcodeImage = CalibrationCompany =
CalibrationPeriod = "";
    IsOut = IsDamaged = OnHold = false;
    CalibrationHistory = new CalibrationLibrary();
    return this;
}
[XmlElement]
public string FirstImage{ get
{
    if (this.Images=="")
    {
        return "/Images/transparent.png";
    }
    var img = Images.Split(',')[0].Replace(",,", "<@#$>").Replace(",", "").Replace("<@#$>",
",").Replace("Images", "").Replace("\\", "");
    img = "/Account/Images/" + img;
    return img;
}
}

```

```

[XmlElement]
public string Images { get; set; }
[XmlElement]
public string AssetName { get; set; }
[XmlElement]
public string OrderNumber { get; set; }
[XmlElement]
public string ShipTo { get; set; }
[XmlElement]
public DateTime DateShipped { get; set; }
[XmlElement]
public DateTime LastCalibrated { get; set; }
[XmlElement]
public DateTime DateRecieved { get; set; }
[XmlElement]
public string DateRecievedString { get { return DateRecieved.ToString(); } }
[XmlElement]
public string LastCalibratedString { get { return LastCalibrated.ToString(); } }
[XmlElement]
public string DateShippedString { get { return DateShipped.ToString(); } }
[XmlIgnore]
public string DateShippedTicks { get { return DateShipped.Ticks.ToString(); } }
[XmlElement]
public string ServiceEngineer { get; set; }
[XmlElement]
public string PersonShipping { get; set; }
[XmlElement]
public string Barcode { get; set; }
[XmlElement]
public string AssetNumber { get; set; } // "0000";
[XmlElement]
public string Description { get; set; }
[XmlElement("BarcodeImage")]
public string BarcodeImage { get; set; }
[XmlElement("IsOut")]
public bool IsOut { get; set; }
[XmlElement("IsDamaged")]
public bool IsDamaged { get; set; }
[XmlElement("IsCalibrated")]
public bool IsCalibrated { get; set; }
[XmlElement("CalibrationCompany")]
public string CalibrationCompany { get; set; }

```

```

[XmlElement("CalibrationPeriod")]
public string CalibrationPeriod { get; set; }
[XmlElement("CalibrationHistory")]
public CalibrationLibrary CalibrationHistory { get; set; }
[XmlElement("OnHold")]
public bool OnHold { get; set; }
[XmlElement("weight")]
public decimal weight { get; set; }
[XmlElement("PackingSlip")]
public string PackingSlip { get; set; }
[XmlElement("UpsLabel")]
public string UpsLabel { get; set; }
[XmlElement("ReturnReport")]
public string ReturnReport { get; set; }
[XmlElement("History")]
public AssetHistory History { get; set; }
[XmlElement]
public bool IsHistoryItem { get; set; }
[XmlIgnore]
public string Color
{
    get
    {
        if (IsOut)
            return "bg-sg-amber";
        if (IsDamaged)
            return "bg-sg-red";
        if (OnHold)
            return "bg-sg-violet";

        return "bg-sg-blue";
    }
}

}

[Serializable]
public class AssetHistory:Serializers.XSerializer<AssetHistory>
{
    public AssetHistory Deserialize(string xml)
    {

```

```

        return this.DeserializeFromXmlString<AssetHistory>(xml);
    }
    public string Serialize()
    {
        return this.SerializeToXmlString(this);
    }
    public AssetHistory()
    {
        History = new List<Asset>();
    }
    [XmlElement]
    public List<Asset> History { get; set; }
}
[Serializable]//xml
public class DataStore:Serializers.XSerializer<DataStore>, ICloneable
{
    [XmlElement]
    public Settings Settings = new Settings();
    [XmlElement]
    public List<Asset> Assets = new List<Asset>();
    [XmlElement]
    public Notification.NotificationSystem NotificationSystem = new
Notification.NotificationSystem();
    [XmlElement]
    public string Name;
    [XmlElement]
    public string Filename;
    [XmlIgnore]
    public object Tag;
    public DataStore()
    {
        Settings = new Settings();
    }

    public object Clone()
    {
        return this.MemberwiseClone();
    }
}
[Serializable]
public class EmailNotice : Notification.NotificationSystem.Notice
{

```

```

    public static EmailNotice Create(List<string> assets, List<EmailAddress> emails,string body,
string subject)
    {
        EmailNotice n = new EmailNotice();
        n.Emails = new List<EmailAddress>();
        n.Emails.AddRange(emails);
        n.Assets = assets;
        n.Body = body;
        n.Subject = subject;
        return n;
    }
    public EmailNotice()
    {
        Body = ""; Subject = "";
        Emails = new List<EmailAddress>();
        Assets = new List<string>();
    }
    [XmlElement("Emails")]
    public List<EmailAddress> Emails { get; set; }
    [XmlElement]
    public List<string> Assets = new List<string>();
    [XmlElement]
    public string Body { get; set; }
    [XmlElement]
    public string Subject { get; set; }
    [XmlIgnore]
    public object Tag { get; set; }
    [XmlElement("EmailsCSV")]
    public string EmailsCSV
    {
        get
        {
            string ret = "";
            foreach (var email in Emails)
            {
                ret += email.Name + ", ";
            }
            return ret;
        }
    }
}

```

[Serializable]

```
public class PendingTransaction : Serializers.XSerializer<PendingTransaction>
{
    public static PendingTransaction Create()
    {
        PendingTransaction n = new PendingTransaction();

        return n;
    }
    public PendingTransaction()
    {
        TransactionID = Guid.NewGuid().ToString();
        Name = HttpContext.Current.User.Identity.Name;
        var list = from a in Global.Library.Settings.ServiceEngineers where a.Email ==
HttpContext.Current.User.Identity.Name select a;
        if (list.Count()==0)
        {
            Email = new EmailAddress() { Name = HttpContext.Current.User.Identity.Name, Email =
HttpContext.Current.User.Identity.Name };
        }else
        {
            Email = list.First();
        }
        OrderNumber = "";
        Address = new Customer();
        Assets = new List<string>();
        ConfirmationNumber = "";
        Date = DateTime.Now;
    }
    public PendingTransaction( List<Asset> assets)
    {
        TransactionID = Guid.NewGuid().ToString();
        ConfirmationNumber = "";
        OrderNumber = "";
        Name = HttpContext.Current.User.Identity.Name;
        var list = from a in Global.Library.Settings.ServiceEngineers where a.Email ==
HttpContext.Current.User.Identity.Name select a;
        if (list.Count() == 0)
        {
            Email = new EmailAddress() { Name = HttpContext.Current.User.Identity.Name, Email =
HttpContext.Current.User.Identity.Name };
        }
    }
}
```

```

else
{
    Email = list.First();
}
Address = new Customer();
this.Assets = (from a in assets select a.AssetNumber).ToList();
Date = DateTime.Now;
}
[XmlElement]
public EmailAddress Email { get; set; }
[XmlElement]
public DateTime Date { get; set; }

//THIS IS NOT SERIALIZING
[XmlElement]
public List<string> Assets = new List<string>();

[XmlElement]
public string Name { get; set; }
[XmlElement]
public string TransactionID { get; set; }
[XmlElement]
public string ConfirmationNumber { get; set; }
[XmlElement]
public string OrderNumber { get; set; }
[XmlElement]
public Customer Address { get; set; }
[XmlIgnore]
public object Tag { get; set; }

}
}

```

```
using System;
using System.Security.Claims;
using System.Threading.Tasks;
using Microsoft.AspNet.Identity;
using Microsoft.AspNet.Identity.EntityFramework;
using Microsoft.AspNet.Identity.Owin;
using Microsoft.Owin;
using Microsoft.Owin.Security;
using Web_App_Master.Models;
using System.Configuration;
```

```
namespace Web_App_Master
```

```
{
    public class EmailService : IIdentityMessageService
    {
        public Task SendAsync(IdentityMessage message)
        {
            // Plug in your email service here to send an email.
            return EmailHelper.SendAsync(message);
        }
    }
}
```

```
public class SmsService : IIdentityMessageService
{
    public Task SendAsync(IdentityMessage message)
    {
        // Plug in your SMS service here to send a text message.
        return Task.FromResult(0);
    }
}
```

// Configure the application user manager used in this application. UserManager is defined in ASP.NET Identity and is used by the application.

```
public class ApplicationUserManager : UserManager<ApplicationUser>
{
    public ApplicationUserManager(IUserStore<ApplicationUser> store)
        : base(store)
    {
    }

    public static ApplicationUserManager
```



```

Create(IdentityFactoryOptions<ApplicationUserManager> options, IOwinContext context)
{
    var manager = new ApplicationUserManager(new
UserStore<ApplicationUser>(context.Get<ApplicationDbContext>()));
    // Configure validation logic for usernames
    manager.UserValidator = new UserValidator<ApplicationUser>(manager)
    {
        AllowOnlyAlphanumericUserNames = false,
        RequireUniqueEmail = true
    };

    // Configure validation logic for passwords
    manager.PasswordValidator = new PasswordValidator
    {
        RequiredLength = 4,
        RequireNonLetterOrDigit = false,
        RequireDigit = false,
        RequireLowercase = false,
        RequireUppercase = false
    };

    // Register two factor authentication providers. This application uses Phone and Emails as
a step of receiving a code for verifying the user
    // You can write your own provider and plug it in here.
    manager.RegisterTwoFactorProvider("Phone Code", new
PhoneNumberTokenProvider<ApplicationUser>
    {
        MessageFormat = "Your security code is {0}"
    });
    manager.RegisterTwoFactorProvider("Email Code", new
EmailTokenProvider<ApplicationUser>
    {
        Subject = "Security Code",
        BodyFormat = "Your security code is {0}"
    });

    // Configure user lockout defaults
    manager.UserLockoutEnabledByDefault = true;
    manager.DefaultAccountLockoutTimeSpan = TimeSpan.FromMinutes(5);
    manager.MaxFailedAccessAttemptsBeforeLockout = 5;

    manager.EmailService = new EmailService();

```

```

        manager.SmsService = new SmsService();
        var dataProtectionProvider = options.DataProtectionProvider;
        if (dataProtectionProvider != null)
        {
            manager.UserTokenProvider = new
DataProtectorTokenProvider<ApplicationUser>(dataProtectionProvider.Create("ASP.NET
Identity"));
        }
        return manager;
    }
}

```

```

public class ApplicationSignInManager : SignInManager<ApplicationUser, string>
{
    public ApplicationSignInManager(ApplicationUserManager userManager,
IAuthenticationManager authenticationManager) :
        base(userManager, authenticationManager) { }

    public override Task<ClaimsIdentity> CreateUserIdentityAsync(ApplicationUser user)
    {
        return user.GenerateUserIdentityAsync((ApplicationUserManager)userManager);
    }

    public static ApplicationSignInManager
Create(IdentityFactoryOptions<ApplicationSignInManager> options, IOwinContext context)
    {
        return new
ApplicationSignInManager(context.GetUserManager<ApplicationUserManager>(),
context.Authentication);
    }
}

public class ApplicationRoleManager : RoleManager<IdentityRole>
{
    public ApplicationRoleManager(IRoleStore<IdentityRole, string> roleStore)
        : base(roleStore)
    {
    }
}

```

```

    public static ApplicationRoleManager
Create(IdentityFactoryOptions<ApplicationRoleManager> options, IOwinContext context)
    {
        var appRoleManager = new ApplicationRoleManager(new

```

```
RoleStore<IdentityRole>(context.Get<ApplicationDbContext>()));
```

```
    return appRoleManager;
```

```
}
```

```
}
```

```
}
```

```

using Helpers;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Linq;
using System.Web;
using System.Xml.Serialization;

namespace Notification
{
    /// <summary>
    /// Self Contained Scheduled Notificaton System...Just Add Notice
    /// </summary>
    [XmlInclude(typeof(EmailNotice))]
    [Serializable]
    [XmlRoot]
    public class NotificationSystem:Serializers.XSerializer<NotificationSystem>, ICloneable
    {
        [XmlElement]
        public string m_guid = new System.Guid().ToString();
        [XmlElement]
        public string Guid { get { return m_guid; } set { m_guid = value; } }

        public delegate Notice NoticeActionDelegate(Notice n);
        [XmlIgnore]
        private int m_capacity = -1;
        /// <summary>
        /// Self Removing Que...If Notices reaches Capacity the LastItem is removed
        /// </summary>
        [XmlElement]
        public int Capacity { get { return m_capacity; } set { m_capacity = value; } }

        private static Notice DefaultAction(Notice n) { return n; }
        public NotificationSystem()
        {
            Guid = System.Guid.NewGuid().ToString();
            Notices = new NoticeBindinglist();
            Notices.ListChanged += Notices_ListChanged;
            NoticeTimer = new System.Timers.Timer();
            NoticeTimer.Elapsed += NoticeTimer_Elapsed;
            NoticeTimer.Interval = m_interval; // Every 5 Minutes by default
            NoticeTimer.Enabled = true;

```

```

}
public NotificationSystem(string app_name)
{

    Guid = app_name;
    Notices = new NoticeBindinglist();
    Notices.ListChanged += Notices_ListChanged;
    NoticeTimer = new System.Timers.Timer();
    NoticeTimer.Elapsed += NoticeTimer_Elapsed;
    NoticeTimer.Interval = m_interval; // Every 5 Minutes by default
    NoticeTimer.Enabled = true;
}

[Serializable]
public enum NoticeType
{
    None,User,Customer,Administration,Log,Application,Asset,AssetList,Calibration,Notice,Checkout,
    Checkin,Damaged,OnHold
}

[XmlElement]
public NoticeBindinglist Notices { get; set; }
[XmlElement]
public double Interval { get { return m_interval; } set { if (value >= 60000) { m_interval = value;
NoticeTimer.Interval = value; } } }
[XmlIgnore]
private System.Timers.Timer NoticeTimer;
[XmlElement]
private double m_interval = (1000 * 60) * 5;
[XmlElement]
private int Count { get { return Notices.Count; } }

private void NoticeTimer_Elapsed(object sender, System.Timers.ElapsedEventArgs e)
{
    HandleTimerTick(new TimerTickEvent(this));
    foreach (var notice in Notices)
    {
        try
        {
            if (notice.IsExpirable && notice.IsExpired)
            {
                HandleExpiredNotice(new NotificationEvent(notice));
            }
        }
        catch { }
    }
}

```

```

        break;
    }
    if (notice.IsScheduledTime && notice.IsTimed)
    {

        HandleOnScheduledTime(new NotificationEvent(notice));
    }
    if (!notice.IsTimed)
    {
        HandleNonTimedNotice(new NotificationEvent(notice));
    }
}catch
{

}

}
}
private void Notices_ListChanged(object sender, ListChangedEventArgs e)
{
    if (e.ListChangedType == ListChangedType.ItemDeleted)
    {
        HandleRemovedNotice(new NotificationEvent(Notices[e.NewIndex]));
    }
    if (e.ListChangedType == ListChangedType.ItemAdded)
    {
        HandleAddedNotice(new NotificationEvent(Notices[e.NewIndex]));
    }
    HandleChangedNotice(new NotificationEvent(Notices[e.NewIndex], e));
}

public void Add(Notice notice)
{
    if (Capacity == -1) { Notices.Add(notice); return; }
    else
    {
        if (Notices.Count==Capacity)
        {
            Notices.RemoveAt(Notices.Count - 1);
        }
        Notices.Add(notice);
    }
}

```

```

    }
    public bool Remove(Notice notice)
    {
        return Notices.Remove(notice);
    }
    public bool Contains(Notice notice)
    {
        return Notices.Contains(notice);
    }
    public bool Contains(string guid)
    {
        try
        {
            return Convert.ToBoolean((from n in Notices where n.Guid == guid select
n).ToList().Count);
        }
        catch { return false; }
    }

    }
    public Notice Get(string guid)
    {
        return (from n in Notices where n.Guid == guid select n).ToList().FirstOrDefault();
    }
    public void IndexOf(Notice notice)
    {
        Notices.IndexOf(notice);
    }
    public void Start()
    {
        NoticeTimer.Enabled = true;
    }
    public void Stop()
    {
        NoticeTimer.Enabled = false;
    }
    public string Save()
    {
        try
        {
            return SerializeToString(this);
        }
        catch(Exception ex) { HandleException(new ExcpetionEvent(ex)); return null; }
    }

```

```

}
public static NotificationSystem Load(string xml)
{
    try
    {
        return new NotificationSystem().DeserializeFromXmlString<NotificationSystem>(xml);
    }
    catch (Exception ex) { return null; }
}

}
#region Sub-Classes
[Serializable]

public class NoticeBindinglist : BindingList<Notice>
{
    public NoticeBindinglist() { this.AllowEdit = AllowNew = AllowRemove = true; }
    public NoticeBindinglist(List<Notice> notices)
    {
        this.AllowEdit = AllowNew = AllowRemove = true;
        foreach (var notice in notices)
        {
            this.Add(notice);
        }
    }
}

[Serializable]
public class Notice:Serializers.XSerializer<Notice>
{
    [XmlIgnore]
    public NoticeActionDelegate NoticeAction = DefaultAction;
    /// <summary>
    /// Create a Notification
    /// </summary>
    public Notice()
    {
        this.IsExpirable = false;
        this.Expires = DateTime.Now.AddMonths(12);
        this.Guid = System.Guid.NewGuid().ToString();
        this.NoticeControlNumber = "";
        this.Data = "";
        this.EmailAddress = new EmailAddress();
        this.EmailAddress.Name = "null.null";
    }
}

```



```

        this.EmailAddress.Email = "null@null.null";
        Created = DateTime.Now;
        Scheduled = Created.AddMonths(1);
        IsTimed = true;
        NoticeType = NoticeType.None;
    }
    /// <summary>
    /// Create a Notification
    /// </summary>
    /// <param name="sheduledTime">Expiration Date</param>
    /// <param name="noticeAction">ACtion</param>
    public Notice(DateTime sheduledTime, NoticeActionDelegate noticeAction=null)
    {
        this.IsExpirable = false;
        this.Expires = DateTime.Now.AddMonths(12);
        if (noticeAction!=null)NoticeAction = noticeAction;
        Guid = System.Guid.NewGuid().ToString();
        NoticeControlNumber = "";

        Data = "";
        this.EmailAddress = new EmailAddress();
        this.EmailAddress.Name = "null.null";
        this.EmailAddress.Email = "null@null.null";
        Created = DateTime.Now;
        Scheduled = sheduledTime;
        IsTimed = true;
        NoticeType = NoticeType.None;
    }
    public Notice(DateTime sheduledTime, NoticeType type, NoticeActionDelegate
noticeAction = null)
    {
        this.IsExpirable = false;
        this.Expires = DateTime.Now.AddMonths(12);
        if (noticeAction != null) NoticeAction = noticeAction;
        Guid = System.Guid.NewGuid().ToString();
        NoticeControlNumber = ""; Data = "";
        this.EmailAddress = new EmailAddress();
        this.EmailAddress.Name = "null.null";
        this.EmailAddress.Email = "null@null.null";
        Created = DateTime.Now;
        Scheduled = sheduledTime;
        IsTimed = true;

```

```

        NoticeType = type;
    }
    public Notice(DateTime sheduledTime, NoticeType type, EmailAddress email,
NoticeActionDelegate noticeAction = null)
    {
        this.IsExpirable = false;
        this.Expires = DateTime.Now.AddMonths(12);
        if (noticeAction != null) NoticeAction = noticeAction;
        Guid = System.Guid.NewGuid().ToString();
        NoticeControlNumber = ""; Data = "";
        this.EmailAddress = email;
        Created = DateTime.Now;
        Scheduled = sheduledTime;
        IsTimed = true;
        NoticeType = type;
    }
    public Notice(DateTime sheduledTime, NoticeType type, EmailAddress email, string
controlNumber, NoticeActionDelegate noticeAction = null)
    {
        this.IsExpirable = false;
        this.Expires = DateTime.Now.AddMonths(12);
        if (noticeAction != null) NoticeAction = noticeAction;
        Guid = System.Guid.NewGuid().ToString();
        NoticeControlNumber = controlNumber;
        Data = "";
        this.EmailAddress = email;
        Created = DateTime.Now;
        Scheduled = sheduledTime;
        IsTimed = true;
        NoticeType = type;
    }
    public Notice(DateTime sheduledTime, NoticeType type, EmailAddress email, string
controlNumber, string data, NoticeActionDelegate noticeAction = null)
    {
        this.IsExpirable = false;
        this.Expires = DateTime.Now.AddMonths(12);
        if (noticeAction != null) NoticeAction = noticeAction;
        Guid = System.Guid.NewGuid().ToString();
        NoticeControlNumber = controlNumber;

        Data = data;
        this.EmailAddress = email;

```

```

        Created = DateTime.Now;
        Scheduled = sheduledTime;
        IsTimed = true;
        NoticeType = type;
    }
    public static Notice Load(string guid)
    {
        return Web_App_Master.Pull.Notification(guid);
    }
    public void Save()
    {
        Web_App_Master.Push.Notification(this);
    }
    [XmlElement]
    public string NoticeControlNumber { get; set; }
    [XmlElement]
    public NoticeType NoticeType { get; set; }
    [XmlElement]
    public string Guid { get; set; }
    [XmlElement]
    public string Data { get; set; }
    [XmlElement]
    public string Name { get { return this.EmailAddress.Name; } set { this.EmailAddress.Name
= value; } }
    [XmlElement]
    public EmailAddress EmailAddress { get; set; }
    [XmlElement]
    public string Email { get { return this.EmailAddress.Email; } set { this.EmailAddress.Email =
value; } }
    [XmlElement]
    public DateTime Created { get; set; }
    [XmlIgnore]
    public double MinutesUntil
    {
        get
        {
            return Math.Round((Scheduled - DateTime.Now).TotalMinutes);
        }
    }

    }
    [XmlIgnore]
    public double HoursUntil

```

```

{
    get
    {
        return Math.Round((Scheduled - DateTime.Now).TotalHours);
    }
}
[XmlIgnore]
public double DaysUntil
{
    get
    {
        var days = (Scheduled- DateTime.Now).TotalDays;
        return Math.Round(days);
    }
}
[XmlElement]
public DateTime Scheduled { get; set; }
[XmlElement]
public DateTime Expires { get; set; }
[XmlIgnore]
public bool IsScheduledTime
{
    get
    {
        {
            if (MinutesUntil>0)
            {
                return false;
            }
            else
            {
                return true;
            }
        }
    }
}
[XmlIgnore]
public bool IsExpired
{
    get
    {
        if ((Expires-Created).TotalSeconds > 0)

```

```

        {
            return false;
        }
        else
        {
            return true;
        }
    }
}
[XmlElement]
public bool IsTimed { get; set; }
[XmlElement]
public bool IsExpirable { get; set; }
}
#endregion
#region Events
public class TimerTickEvent : EventArgs
{
    public TimerTickEvent(object obj)
    {
        this.Notice = obj;
    }
    public object Notice { get; set; }
}
public class NotificationEvent : EventArgs
{
    public NotificationEvent(Notice notice = null, ListChangedEventArgs
changeArguments=null)
    {
        this.Notice = notice;
        this.ChangeArguments = changeArguments;
    }
    public Notice Notice { get; set; }
    public ListChangedEventArgs ChangeArguments { get; set; }
}
public class ExcpetionEvent : EventArgs
{
    public ExcpetionEvent(Exception ex = null)
    {
        this.Exception = ex;
    }
    public Exception Exception { get; set; }
}

```

```
}
```

```
public event EventHandler<ExceptionEvent> OnException;  
protected virtual void HandleException(ExceptionEvent e)  
{  
    try  
    {  
        EventHandler<ExceptionEvent> handler = OnException;  
        if (handler != null)  
        {  
            handler(this, e);  
        }  
    }  
    catch { }  
}
```

```
public event EventHandler<NotificationEvent> OnNonTimedNoticeTick;  
protected virtual void HandleNonTimedNotice(NotificationEvent e)  
{  
    try  
    {  
        EventHandler<NotificationEvent> handler = OnNonTimedNoticeTick;  
        if (handler != null)  
        {  
            handler(this, e);  
        }  
    }  
    catch { }  
}
```

```
public event EventHandler<NotificationEvent> OnScheduledTime;  
protected virtual void HandleOnScheduledTime(NotificationEvent e)  
{  
    try  
    {  
        EventHandler<NotificationEvent> handler = OnScheduledTime;  
        if (handler != null)  
        {  
            handler(this, e);  
        }  
    }  
    catch { }  
}
```

```

}
public event EventHandler<NotificationEvent> OnNoticeExpired;
protected virtual void HandleExpiredNotice(NotificationEvent e)
{
    try
    {
        EventHandler<NotificationEvent> handler = OnNoticeExpired;
        if (handler != null)
        {
            handler(this, e);
        }
    }
    catch { }
}
public event EventHandler<NotificationEvent> OnNoticeChanged;
protected virtual void HandleChangedNotice(NotificationEvent e)
{
    try
    {
        EventHandler<NotificationEvent> handler = OnNoticeChanged;
        if (handler != null)
        {
            handler(this, e);
        }
    }
    catch { }
}
public event EventHandler<NotificationEvent> OnNoticeRemoved;
protected virtual void HandleRemovedNotice(NotificationEvent e)
{
    try
    {
        EventHandler<NotificationEvent> handler = OnNoticeRemoved;
        if (handler != null)
        {
            handler(this, e);
        }
    }
    catch { }
}
public event EventHandler<NotificationEvent> OnNoticeAdded;
protected virtual void HandleAddedNotice(NotificationEvent e)

```

```

{
    try
    {
        EventHandler<NotificationEvent> handler = OnNoticeAdded;
        if (handler != null)
        {
            handler(this, e);
        }
    }
    catch { }
}

public event EventHandler<TimerTickEvent> OnTimerTick;
protected virtual void HandleTimerTick(TimerTickEvent e)
{
    try
    {
        EventHandler<TimerTickEvent> handler = OnTimerTick;
        if (handler != null)
        {
            handler(this, e);
        }
    }
    catch { }
}

public object Clone()
{
    return this.MemberwiseClone();
}

#endregion
}
}

```



```

using Ghostscript.NET;
using iTextSharp.text;
using iTextSharp.text.pdf;
//using NReco.PdfRenderer;
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Web;
using Web_App_Master.Account.Templates;
using static iTextSharp.text.pdf.AcroFields;

public class Pdf
{
    /*
    public static System.Drawing.Image Tolmage(string source, string dest,int page=1)
    {
        var pdfTolmg = new PdfTolmageConverter();
        pdfTolmg.ScaleTo = 800;
        var memStream = new MemoryStream();
        pdfTolmg.GenerateImage(source, page, ImageFormat.Png, memStream);
        System.Drawing.Bitmap bm = new System.Drawing.Bitmap(memStream);

        using (FileStream fs = new FileStream(dest, FileMode.Create))
        {
            memStream.Position = 0;
            memStream.CopyTo(fs);
        }
        return bm;
    }*/
    public delegate PdfStamper StampAction(PdfStamper stamper);
    public enum Alignment
    {
        Left, Center, Right
    }
    public static TextField CreateMultiLineField(float x, float y, float width, float height, string name, PdfStamper stamper, Alignment alignment = Alignment.Left)
    {
        TextField Tracking = new TextField(stamper.Writer, new iTextSharp.text.Rectangle(x, y, x + width, y + height), name);
        Tracking.Options = TextField.MULTILINE;
    }

```

```

    try
    {
        Tracking.Alignment = (int)alignment;
    }
    catch { }
    //
    return Tracking;
}

public static TextField CreateField(float x, float y, float width, float height, string name,
PdfStamper stamper)
{
    TextField Tracking = new TextField(stamper.Writer, new iTextSharp.text.Rectangle(x, y, x +
width, y + height), name);
    return Tracking;
}

public static bool CreatePdf(List<TextField> fields, string filename, string template)
{
    try
    {
        var dest = filename;
        PdfReader reader = new PdfReader(template);
        PdfStamper stamper = new PdfStamper(reader, new FileStream(dest, FileMode.Create));
        foreach (var field in fields)
        {
            var a = field.GetTextField();

            stamper.AddAnnotation(a, 1);
        }

        stamper.Close();
        return true;
    }
    catch { return false; }
}

public static bool CreatePdf(List<TextField> fields, Stream output, Stream template)
{
    try
    {

        PdfReader reader = new PdfReader(template);
        PdfStamper stamper = new PdfStamper(reader, output);
        foreach (var field in fields)

```

```

    {
        var a = field.GetTextField();

        stamper.AddAnnotation(a, 1);
    }

    stamper.Close();
    return true;
}
catch { return false; }
}

public static bool CreatePdf(List<TextField> fields, Stream output, byte[] template)
{
    try
    {

        PdfReader reader = new PdfReader(template);
        PdfStamper stamper = new PdfStamper(reader, output);
        foreach (var field in fields)
        {
            var a = field.GetTextField();

            stamper.AddAnnotation(a, 1);
        }

        stamper.Close();
        return true;
    }
    catch { return false; }
}

public static bool StampPdf(StampAction stampAction, Stream output, Stream template)
{
    try
    {
        using (PdfReader reader = new PdfReader(template))
        using (PdfStamper stamper = new PdfStamper(reader, output))
        {
            stampAction.Invoke(stamper);
        }
        return true;
    }
    catch { return false; }
}

```

```

}
public static bool StampPdf(StampAction stampAction, Stream output, string template)
{
    try
    {
        using (PdfReader reader = new PdfReader(template))
        using (PdfStamper stamper = new PdfStamper(reader, output))
        {
            stampAction.Invoke(stamper);
        }
        return true;
    }
    catch { return false; }
}

```

```

public static bool StampPdf(StampAction stampAction, Stream output, byte[] template)
{
    try
    {
        PdfReader reader = new PdfReader(template);
        PdfStamper stamper = new PdfStamper(reader, output);
        {
            stampAction.Invoke(stamper);
        }
        reader.Close();
        return true;
    }
    catch { return false; }
}

```

```

public static bool StampPdf(StampAction stampAction, string filename, byte[] template)
{
    try
    {
        using (PdfReader reader = new PdfReader(template))
        using (FileStream fs = new FileStream(filename, FileMode.Create))
        using (PdfStamper stamper = new PdfStamper(reader, fs))
        {
            stampAction.Invoke(stamper);
        }
    }
}

```

```

        return true;
    }
    catch { return false; }
}

public static bool FillInForm(string source, string dest, Dictionary<string, string> Keys)
{
    try
    {
        using (PdfReader reader = new PdfReader(source))
        using (PdfStamper stamper = new PdfStamper(reader, new FileStream(dest,
FileMode.Create)))
        {
            AcroFields form = stamper.AcroFields;
            foreach (var i in Keys)
            {
                try
                {
                    Item item = form.GetFieldItem(i.Key);
                    form.SetField(i.Key, i.Value);
                }
                catch { }
            }

        }
        return true;
    }
    catch { return false; }
}

public static bool FillInForm(Stream source, Stream dest, Dictionary<string, string> Keys)
{
    try
    {

        using (PdfReader reader = new PdfReader(source))
        using (PdfStamper stamper = new PdfStamper(reader, dest))
        {
            AcroFields form = stamper.AcroFields;
            foreach (var i in Keys)
            {
                Item item = form.GetFieldItem(i.Key);
                form.SetField(i.Key, i.Value);
            }
        }
    }
}

```

```

    }
    return true;
}
catch { return false; }
}
public static bool FillInForm(byte[] source, Stream dest, Dictionary<string, string> Keys)
{
    try
    {
        PdfReader reader = new PdfReader(source);
        PdfStamper stamper = new PdfStamper(reader, dest);

        AcroFields form = stamper.AcroFields;
        foreach (var i in Keys)
        {
            Item item = form.GetFieldItem(i.Key);
            form.SetField(i.Key, i.Value);
        }

        stamper.Close();
        return true;
    }
    catch { return false; }
}

public static iTextSharp.text.Image createBarcode(PdfContentByte cb, String code, float mh = 1,
float mw = 1, bool generateChecksum = true)
{
    Barcode128 pf = new Barcode128();
    pf.GenerateChecksum = (generateChecksum);
    pf.Code = (code);
    //BarcodePDF417 pf = new BarcodePDF417();
    // pf.SetText("BarcodePDF417 barcode");
    iTextSharp.text.Rectangle size = pf.BarcodeSize;
    iTextSharp.text.pdf.PdfTemplate template = cb.CreateTemplate(mw * size.Width, mh *
size.Height);
    pf.PlaceBarcode(template, BaseColor.BLACK, BaseColor.RED);
    return iTextSharp.text.Image.GetInstance(template);
}

public static iTextSharp.text.Image createVectorBarcode(PdfStamper stamper, String code,
float mh = 1, float mw = 1, bool generateChecksum = true, int page = 1)
{
    Barcode128 pf = new Barcode128();

```

```

        pf.GenerateChecksum = (generateChecksum);
        pf.Code = (code);
        //BarcodePDF417 pf = new BarcodePDF417();
        // pf.SetText("BarcodePDF417 barcode");
        iTextSharp.text.Rectangle size = pf.BarcodeSize;
        iTextSharp.text.pdf.PdfTemplate template =
stamper.GetOverContent(page).CreateTemplate(mw * size.Width, mh * size.Height);
        pf.PlaceBarcode(template, BaseColor.BLACK, BaseColor.RED);
        return iTextSharp.text.Image.GetInstance(template);
    }
    public static iTextSharp.text.Image createBarcode(PdfStamper stamper, String code, float mh =
1, float mw = 1, bool generateCheckSum = true, int page = 1)
    {
        Barcode128 code128 = new Barcode128();
        code128.GenerateChecksum = (generateChecksum);
        code128.Code = (code);
        return code128.CreateImageWithBarcode(stamper.GetOverContent(page),
BaseColor.BLACK, BaseColor.RED);
    }

```

```

public class PackingSlip
{
    public StampAction StampAction = DefaultPackingSlipForm;
    public PackingSlip(PdfTemplate template)
    {
        Keys = DefaultPackingSlipKeys;
        Template = template;
    }
    public PdfTemplate Template=null;
    public Dictionary<string, string> Keys;
    public static readonly Dictionary<string, string> DefaultPackingSlipKeys = new
Dictionary<string, string>() {
        {"ShipDate", ""},
        {"Tracking", ""},
        {"Signature", ""},
        {"Attn", ""},
        {"Po", ""},
        {"To", ""},
        {"From", ""},
        {"Weight", ""},

```

{"Ordernumber", ""},

{"Desc1", ""},

{"Desc2", ""},

{"Desc3", ""},

{"Desc4", ""},

{"Desc5", ""},

{"Desc6", ""},

{"Desc7", ""},

{"Desc8", ""},

{"Desc9", ""},

{"Desc10", ""},

{"Desc11", ""},

{"Desc12", ""},

{"Desc13", ""},

{"PartNo1", ""},

{"PartNo2", ""},

{"PartNo3", ""},

{"PartNo4", ""},

{"PartNo5", ""},

{"PartNo6", ""},

{"PartNo7", ""},

{"PartNo8", ""},

{"PartNo9", ""},

{"PartNo10", ""},

{"PartNo11", ""},

{"PartNo12", ""},

{"PartNo13", ""},

{"Qty1", ""},

{"Qty2", ""},

{"Qty3", ""},

{"Qty4", ""},

{"Qty5", ""},

{"Qty6", ""},

{"Qty7", ""},

{"Qty8", ""},

{"Qty9", ""},

{"Qty10", ""},

{"Qty11", ""},

{"Qty12", ""},


```
{"Qty13", ""}  
};
```

```
public static PdfStamper DefaultPackingSlipForm(PdfStamper stamper)
```

```
{
```

```
    try
```

```
    {
```

```
        List<TextField> fields = new List<TextField>();
```

```
        //bottom left margin x,y
```

```
        var x = 38;
```

```
        var y = 55;
```

```
        var w = 260;
```

```
        var h = 45;
```

```
        var name = "ShipDate";
```

```
        fields.Add(Pdf.CreateField(x, y, w, h, name, stamper));
```

```
        //field
```

```
        x = x + 0
```

```
        ;
```

```
        y = y + 54
```

```
        ;
```

```
        w = 260;
```

```
        h = 40;
```

```
        name = "Tracking";
```

```
        fields.Add(CreateField(x, y, w, h, name, stamper));
```

```
        //field
```

```
        x = x + 0
```

```
        ;
```

```
        y = y + 45
```

```
        ;
```

```
        w = 260;
```

```
        h = 40;
```

```
        name = "Signature";
```

```
        fields.Add(CreateField(x, y, w, h, name, stamper));
```

```
        var PartNoStartY = y;
```

```
        //field //*****LOWER X TABLE ANGE FOR ITEMS
```

```
        x = x + 0
```

```
        ;
```

```
        y = y + 45
```

```
    ;  
    w = 260;  
    h = 18;  
    name = "Desc13";  
    fields.Add(CreateField(x, y, w, h, name, stamper));
```

```
//field  
x = x + 0  
    ;  
y = y + 23  
    ;  
w = 260;  
h = 18;  
name = "Desc12";  
fields.Add(CreateField(x, y, w, h, name, stamper));
```

```
//field  
x = x + 0  
    ;  
y = y + 23  
    ;  
w = 260;  
h = 18;  
name = "Desc11";  
fields.Add(CreateField(x, y, w, h, name, stamper));
```

```
//field  
x = x + 0  
    ;  
y = y + 23  
    ;  
w = 260;  
h = 18;  
name = "Desc10";  
fields.Add(CreateField(x, y, w, h, name, stamper));
```

```
//field  
x = x + 0  
    ;  
y = y + 23  
    ;  
w = 260;
```

```
h = 18;  
name = "Desc9";  
fields.Add(CreateField(x, y, w, h, name, stamper));
```

```
//field  
x = x + 0  
;  
y = y + 23  
;  
w = 260;  
h = 18;  
name = "Desc8";  
fields.Add(CreateField(x, y, w, h, name, stamper));
```

```
//field  
x = x + 0  
;  
y = y + 22  
;  
w = 260;  
h = 18;  
name = "Desc7";  
fields.Add(CreateField(x, y, w, h, name, stamper));
```

```
//field  
x = x + 0  
;  
y = y + 22  
;  
w = 260;  
h = 18;  
name = "Desc6";  
fields.Add(CreateField(x, y, w, h, name, stamper));
```

```
//field  
x = x + 0  
;  
y = y + 22  
;  
w = 260;  
h = 18;  
name = "Desc5";
```

```
fields.Add(CreateField(x, y, w, h, name, stamper));
```

```
//field
```

```
x = x + 0
```

```
;
```

```
y = y + 24
```

```
;
```

```
w = 260;
```

```
h = 18;
```

```
name = "Desc4";
```

```
fields.Add(CreateField(x, y, w, h, name, stamper));
```

```
//field
```

```
x = x + 0
```

```
;
```

```
y = y + 22
```

```
;
```

```
w = 260;
```

```
h = 18;
```

```
name = "Desc3";
```

```
fields.Add(CreateField(x, y, w, h, name, stamper));
```

```
//field
```

```
x = x + 0
```

```
;
```

```
y = y + 24
```

```
;
```

```
w = 260;
```

```
h = 18;
```

```
name = "Desc2";
```

```
fields.Add(CreateField(x, y, w, h, name, stamper));
```

```
//field
```

```
x = x + 0
```

```
;
```

```
y = y + 23
```

```
;
```

```
w = 260;
```

```
h = 18;
```

```
name = "Desc1";
```

```
fields.Add(CreateField(x, y, w, h, name, stamper));
```

```
//PART NO COLUMN
```

```
y = PartNoStartY;
```

```
x = x + 262;
```

```
w = 260 / 2;
```

```
h = 18;
```

```
x = x + 0
```

```
;
```

```
y = y + 45;
```

```
name = "PartNo13";
```

```
fields.Add(CreateField(x, y, w, h, name, stamper));
```

```
//field
```

```
x = x + 0
```

```
;
```

```
y = y + 23
```

```
;
```

```
name = "PartNo12";
```

```
fields.Add(CreateField(x, y, w, h, name, stamper));
```

```
//field
```

```
x = x + 0
```

```
;
```

```
y = y + 23
```

```
;
```

```
name = "PartNo11";
```

```
fields.Add(CreateField(x, y, w, h, name, stamper));
```

```
//field
```

```
x = x + 0
```

```
;
```

```
y = y + 23
```

```
;
```

```
name = "PartNo10";
```

```
fields.Add(CreateField(x, y, w, h, name, stamper));
```

```
//field
```

```
x = x + 0
```

```
;
```

```
y = y + 23
```

```
    ;
    name = "PartNo9";
    fields.Add(CreateField(x, y, w, h, name, stamper));

    //field
    x = x + 0
    ;
    y = y + 23
    ;
    name = "PartNo8";
    fields.Add(CreateField(x, y, w, h, name, stamper));

    //field
    x = x + 0
    ;
    y = y + 22
    ;
    name = "PartNo7";
    fields.Add(CreateField(x, y, w, h, name, stamper));

    //field
    x = x + 0
    ;
    y = y + 22
    ;
    name = "PartNo6";
    fields.Add(CreateField(x, y, w, h, name, stamper));

    //field
    x = x + 0
    ;
    y = y + 22
    ;
    name = "PartNo5";
    fields.Add(CreateField(x, y, w, h, name, stamper));

    //field
    x = x + 0
    ;
    y = y + 24
    ;
    name = "PartNo4";
```

```
fields.Add(CreateField(x, y, w, h, name, stamper));
```

```
//field
```

```
x = x + 0
```

```
;
```

```
y = y + 22
```

```
;
```

```
name = "PartNo3";
```

```
fields.Add(CreateField(x, y, w, h, name, stamper));
```

```
//field
```

```
x = x + 0
```

```
;
```

```
y = y + 24
```

```
;
```

```
name = "PartNo2";
```

```
fields.Add(CreateField(x, y, w, h, name, stamper));
```

```
//field
```

```
x = x + 0
```

```
;
```

```
y = y + 23
```

```
;
```

```
name = "PartNo1";
```

```
fields.Add(CreateField(x, y, w, h, name, stamper));
```

```
//Qty COLUMN
```

```
y = PartNoStartY;
```

```
x = x + 130;
```

```
w = 260 / 2;
```

```
h = 18;
```

```
x = x + 0
```

```
;
```

```
y = y + 45;
```

```
name = "Qty13";
```

```
fields.Add(CreateField(x, y, w, h, name, stamper));
```

```
//field
x = x + 0
;
y = y + 23
;
name = "Qty12";
fields.Add(CreateField(x, y, w, h, name, stamper));
```

```
//field
x = x + 0
;
y = y + 23
;
name = "Qty11";
fields.Add(CreateField(x, y, w, h, name, stamper));
```

```
//field
x = x + 0
;
y = y + 23
;
name = "Qty10";
fields.Add(CreateField(x, y, w, h, name, stamper));
```

```
//field
x = x + 0
;
y = y + 23
;
name = "Qty9";
fields.Add(CreateField(x, y, w, h, name, stamper));
```

```
//field
x = x + 0
;
y = y + 23
;
name = "Qty8";
fields.Add(CreateField(x, y, w, h, name, stamper));
```

```
//field
x = x + 0
```



```
    ;
    y = y + 22
    ;
    name = "Qty7";
    fields.Add(CreateField(x, y, w, h, name, stamper));

//field
x = x + 0
;
y = y + 22
;
name = "Qty6";
fields.Add(CreateField(x, y, w, h, name, stamper));

//field
x = x + 0
;
y = y + 22
;
name = "Qty5";
fields.Add(CreateField(x, y, w, h, name, stamper));

//field
x = x + 0
;
y = y + 24
;
name = "Qty4";
fields.Add(CreateField(x, y, w, h, name, stamper));

//field
x = x + 0
;
y = y + 22
;
name = "Qty3";
fields.Add(CreateField(x, y, w, h, name, stamper));

//field
x = x + 0
;
y = y + 24
```

```
    ;  
    name = "Qty2";  
    fields.Add(CreateField(x, y, w, h, name, stamper));
```

```
//field  
x = x + 0  
    ;  
y = y + 23  
    ;  
name = "Qty1";  
fields.Add(CreateField(x, y, w, h, name, stamper));
```

```
//Attn  
//field  
w = 260;  
h = 40;  
x = 38  
    ;  
y = y + 42  
    ;  
name = "Attn";  
fields.Add(CreateField(x, y, w, h, name, stamper));
```

```
//PO  
//field  
x = x + 262  
    ;  
y = y + 0  
    ;  
name = "Po";  
fields.Add(CreateField(x, y, w, h, name, stamper));
```

```
//to  
//field  
w = 260;  
h = 190;  
x = 38  
    ;  
y = y + 48  
    ;  
name = "To";
```

```
fields.Add(CreateMultiLineField(x, y, w, h, name, stamper));
```

```
//from
```

```
//field
```

```
x = x + 262
```

```
;
```

```
y = y + 0
```

```
;
```

```
name = "From";
```

```
fields.Add(CreateMultiLineField(x, y, w, h, name, stamper));
```

```
//weight
```

```
//field
```

```
w = 260;
```

```
h = 40;
```

```
x = 38
```

```
;
```

```
y = y + 42 + 155
```

```
;
```

```
name = "Weight";
```

```
fields.Add(CreateField(x, y, w, h, name, stamper));
```

```
//order number
```

```
//field
```

```
x = x + 262
```

```
;
```

```
y = y + 0
```

```
;
```

```
name = "Ordernumber";
```

```
var ordernum = CreateField(x, y, w, h, name, stamper);
```

```
ordernum.Alignment = PdfAppearance.ALIGN_RIGHT;
```

```
fields.Add(ordernum);
```

```
//*****CHANGE ME*****
```

```
// add the field here, the second param is the page you want it on  
foreach (var field in fields)
```

```
{
```

```
    var a = field.GetTextField();
```

```
    stamper.AddAnnotation(a, 1);
```

```

    }

    }
    catch { }
    return stamper;

}
public bool Create(Stream output)
{
    return Pdf.StampPdf(this.StampAction, output, Template.TemplateBytes);
}
public bool Fill(Stream output, Dictionary<string, string> keys)
{
    return Pdf.FillInForm(Template.TemplateBytes, output, keys);
}
public bool CreateAndFill(string file, Dictionary<string, string> keys, StampAction action =
null)
{

    var temp = HttpContext.Current.Server.MapPath(
"/Account/PackingLists/"+Guid.NewGuid() + ".pdf");
    try
    {
        bool result = false;
        Pdf.PackingSlip ps = new Pdf.PackingSlip(Template);
        using (MemoryStream ms = new MemoryStream())
        {

            if (action == null)
                result = StampPdf(this.StampAction, temp, Template.TemplateBytes);
            else
                result = StampPdf(action, temp, Template.TemplateBytes);

            if (result)
            {
                //fill data
                using (MemoryStream fms = new MemoryStream())
                {

                    result = Pdf.FillInForm(temp,file, keys);

```

```

        }

    }
}
File.Delete(temp);
return result;
}
catch { return false; }
}
public bool CreateAndFill(Stream output, Dictionary<string, string> keys, StampAction
action=null)
{
    try
    {
        bool result = false;
        PackingSlip ps = new Pdf.PackingSlip(Template);
        if (action==null)
            result = StampPdf(this.StampAction, output, Template.TemplateBytes);
        else
            result = StampPdf(action, output, Template.TemplateBytes);
        if (result)
        {
            //fill data
            result = ps.Fill(output, keys);
            if (result)
            {
                return true;

            }
            else
            {
                return false;
            }
        }
    }

    return true;
}
catch { return false; }
}
}

```

```

public class Avery5160TextLabels
{
    public Avery5160TextLabels()
    {
        Keys = DefaultAveryTextKeys;
        Template = new Avery5160TextLabelTemplate();
    }
    public Avery5160TextLabels(PdfTemplate template)
    {
        Keys = DefaultAveryTextKeys;
        Template = template;
    }

    public static PdfStamper DefaultAvery5160textLabels(PdfStamper stamper)
    {
        List<TextField> fields = new List<TextField>();
        var x = 17;
        var y = 38;
        var w = 183;
        var h = 68;
        var name = "Col1Row10";
        fields.Add(CreateMultiLineField(x, y, w, h, name, stamper));
        //Field
        x = x + 0
        ;
        y = y + 72
        ; w = 183; h = 68;
        name = "Col1Row9";
        fields.Add(CreateMultiLineField(x, y, w, h, name, stamper));
        //Field
        x = x + 0
        ;
        y = y + 72
        ; w = 183; h = 68;
        name = "Col1Row8";
        fields.Add(CreateMultiLineField(x, y, w, h, name, stamper));
        //Field
        x = x + 0
        ;
        y = y + 72
        ; w = 183; h = 68;
        name = "Col1Row7";
    }
}

```

```

fields.Add(CreateMultiLineField(x, y, w, h, name, stamper));
//Field
x = x + 0
;
y = y + 72
; w = 183; h = 68;
name = "Col1Row6";
fields.Add(CreateMultiLineField(x, y, w, h, name, stamper));
//Field
x = x + 0
;
y = y + 72
; w = 183; h = 68;
name = "Col1Row5";
fields.Add(CreateMultiLineField(x, y, w, h, name, stamper));
//Field
x = x + 0
;
y = y + 72
; w = 183; h = 68;
name = "Col1Row4";
fields.Add(CreateMultiLineField(x, y, w, h, name, stamper));
//Field
x = x + 0
;
y = y + 72
; w = 183; h = 68;
name = "Col1Row3";
fields.Add(CreateMultiLineField(x, y, w, h, name, stamper));
//Field
x = x + 0
;
y = y + 72
; w = 183; h = 68;
name = "Col1Row2";
fields.Add(CreateMultiLineField(x, y, w, h, name, stamper));
//Field
x = x + 0
;
y = y + 72
; w = 183; h = 68;
name = "Col1Row1";

```

```
fields.Add(CreateMultiLineField(x, y, w, h, name, stamper));
```

```
//Next Column
```

```
//Field
```

```
x = x + 198
```

```
;
```

```
y = 38
```

```
; w = 183; h = 68;
```

```
name = "Col2Row10";
```

```
fields.Add(CreateMultiLineField(x, y, w, h, name, stamper));
```

```
//Field
```

```
x = x + 0
```

```
;
```

```
y = y + 72
```

```
; w = 183; h = 68;
```

```
name = "Col2Row9";
```

```
fields.Add(CreateMultiLineField(x, y, w, h, name, stamper));
```

```
//Field
```

```
x = x + 0
```

```
;
```

```
y = y + 72
```

```
; w = 183; h = 68;
```

```
name = "Col2Row8";
```

```
fields.Add(CreateMultiLineField(x, y, w, h, name, stamper));
```

```
//Field
```

```
x = x + 0
```

```
;
```

```
y = y + 72
```

```
; w = 183; h = 68;
```

```
name = "Col2Row7";
```

```
fields.Add(CreateMultiLineField(x, y, w, h, name, stamper));
```

```
//Field
```

```
x = x + 0
```

```
;
```

```
y = y + 72
```

```
; w = 183; h = 68;
```

```
name = "Col2Row6";
```

```
fields.Add(CreateMultiLineField(x, y, w, h, name, stamper));
```

```
//Field
```

```
x = x + 0
```

```
;
```



```

y = y + 72
    ; w = 183; h = 68;
name = "Col2Row5";
fields.Add(CreateMultiLineField(x, y, w, h, name, stamper));
//Field
x = x + 0
    ;
y = y + 72
    ; w = 183; h = 68;
name = "Col2Row4";
fields.Add(CreateMultiLineField(x, y, w, h, name, stamper));
//Field
x = x + 0
    ;
y = y + 72
    ; w = 183; h = 68;
name = "Col2Row3";
fields.Add(CreateMultiLineField(x, y, w, h, name, stamper));
//Field
x = x + 0
    ;
y = y + 72
    ; w = 183; h = 68;
name = "Col2Row2";
fields.Add(CreateMultiLineField(x, y, w, h, name, stamper));
//Field
x = x + 0
    ;
y = y + 72
    ; w = 183; h = 68;
name = "Col2Row1";
fields.Add(CreateMultiLineField(x, y, w, h, name, stamper));

//Next Column
//Next Column

//Field
x = x + 198
    ;
y = 38
    ; w = 183; h = 68;
name = "Col3Row10";

```

```

fields.Add(CreateMultiLineField(x, y, w, h, name, stamper));
//Field
x = x + 0
;
y = y + 72
; w = 183; h = 68;
name = "Col3Row9";
fields.Add(CreateMultiLineField(x, y, w, h, name, stamper));
//Field
x = x + 0
;
y = y + 72
; w = 183; h = 68;
name = "Col3Row8";
fields.Add(CreateMultiLineField(x, y, w, h, name, stamper));
//Field
x = x + 0
;
y = y + 72
; w = 183; h = 68;
name = "Col3Row7";
fields.Add(CreateMultiLineField(x, y, w, h, name, stamper));
//Field
x = x + 0
;
y = y + 72
; w = 183; h = 68;
name = "Col3Row6";
fields.Add(CreateMultiLineField(x, y, w, h, name, stamper));
//Field
x = x + 0
;
y = y + 72
; w = 183; h = 68;
name = "Col3Row5";
fields.Add(CreateMultiLineField(x, y, w, h, name, stamper));
//Field
x = x + 0
;
y = y + 72
; w = 183; h = 68;
name = "Col3Row4";

```

```

fields.Add(CreateMultiLineField(x, y, w, h, name, stamper));
//Field
x = x + 0
;
y = y + 72
; w = 183; h = 68;
name = "Col3Row3";
fields.Add(CreateMultiLineField(x, y, w, h, name, stamper));
//Field
x = x + 0
;
y = y + 72
; w = 183; h = 68;
name = "Col3Row2";
fields.Add(CreateMultiLineField(x, y, w, h, name, stamper));
//Field
x = x + 0
;
y = y + 72
; w = 183; h = 68;
name = "Col3Row1";
fields.Add(CreateMultiLineField(x, y, w, h, name, stamper));

```

```

// add the field here, the second param is the page you want it on
foreach (var field in fields)

```

```

{
    //field.Text = ("default text");
    var a = field.GetTextField();

    stamper.AddAnnotation(a, 1);
}

```

```

return stamper;

```

```

}

```

```

public PdfTemplate Template = null;

```

```

public Dictionary<string, string> Keys;

```

```

public StampAction StampAction = DefaultAvery5160textLabels;

```

```

public static readonly Dictionary<string, string> DefaultAveryTextKeys = new

```

```

Dictionary<string, string>() {
    {"Col1Row1", ""},
    {"Col1Row2", ""},

```

```

{"Col1Row3", ""},
{"Col1Row4", ""},
{"Col1Row5", ""},
{"Col1Row6", ""},
{"Col1Row7", ""},
{"Col1Row8", ""},
{"Col1Row9", ""},
{"Col1Row10", ""},
{"Col2Row1", ""},
{"Col2Row2", ""},
{"Col2Row3", ""},
{"Col2Row4", ""},
{"Col2Row5", ""},
{"Col2Row6", ""},
{"Col2Row7", ""},
{"Col2Row8", ""},
{"Col2Row9", ""},
{"Col2Row10", ""},
{"Col3Row1", ""},
{"Col3Row2", ""},
{"Col3Row3", ""},
{"Col3Row4", ""},
{"Col3Row5", ""},
{"Col3Row6", ""},
{"Col3Row7", ""},
{"Col3Row8", ""},
{"Col3Row9", ""},
{"Col3Row10", ""},
};

```

```

public bool Create(Stream output)

```

```

{
    return Pdf.StampPdf(this.StampAction, output, Template.TemplateBytes);
}

```

```

public bool Fill(Stream output, Dictionary<string, string> keys)

```

```

{
    return Pdf.FillInForm(Template.TemplateBytes, output, keys);
}

```

```

public bool CreateAndFill(string file, Dictionary<string, string> keys, StampAction action =
null)

```

```

{

```

```

    var temp = HttpContext.Current.Server.MapPath("/Account/Avery/" + Guid.NewGuid() +

```

```

".pdf");
    try
    {
        bool result = false;
        Pdf.PackingSlip ps = new Pdf.PackingSlip(Template);
        using (MemoryStream ms = new MemoryStream())
        {

            if (action == null)
                result = StampPdf(this.StampAction, temp, Template.TemplateBytes);
            else
                result = StampPdf(action, temp, Template.TemplateBytes);

            if (result)
            {
                //fill data
                using (MemoryStream fms = new MemoryStream())
                {

                    result = Pdf.FillInForm(temp, file, keys);

                }

            }
        }
        File.Delete(temp);
        return result;
    }
    catch { return false; }
}

public bool CreateAndFill(Stream output, Dictionary<string, string> keys, StampAction action
= null)
{
    try
    {
        bool result = false;
        PackingSlip ps = new Pdf.PackingSlip(Template);
        if (action == null)
            result = StampPdf(this.StampAction, output, Template.TemplateBytes);
        else
            result = StampPdf(action, output, Template.TemplateBytes);
    }
}

```

```

        if (result)
        {
            //fill data
            result = ps.Fill(output, keys);
            if (result)
            {
                return true;
            }
            else
            {
                return false;
            }
        }

        return true;
    }
    catch { return false; }
}

```

```

}
public class Avery5160ImageLabels
{
    public Avery5160ImageLabels()
    {

        Template = new Avery5160TextLabelTemplate();
    }
    public Avery5160ImageLabels(PdfTemplate template)
    {

        Template = template;
    }

    public PdfTemplate Template = null;
    public List<string> ImagePaths;
    public bool Create(Stream output)
    {

        try
        {

```

```

PdfReader reader = new PdfReader(Template.TemplateBytes);
PdfStamper stamper = new PdfStamper(reader, output);
var x = 17;
var y = 38;
var w = 183;
var h = 68;
int count = 0;
foreach (var image in ImagePaths)
{
    if (count == 10 || count == 20) //Reset Column
    {
        x = x + 198;
        y = 38;
    }
    stamper = Stammers.ImageStamper(stamper, image, x, y, w, h);
    //Set Next Row
    y = y + 72;
    count++;
}

stamper.Close();
return true;
}
catch { return false; }

}
public bool Create(string file)
{
    try
    {
        PdfReader reader = new PdfReader(Template.TemplateBytes);
        PdfStamper stamper = new PdfStamper(reader, File.Open(file, FileMode.Create));
        var x = 17;
        var y = 38;
        var w = 183;
        var h = 68;
        int count = 0;
        foreach (var image in ImagePaths)
        {
            if (count == 10 || count == 20) //Reset Column
            {

```

```

        x = x + 198;
        y = 38;
    }
    stamper = Stammers.ImageStamper(stamper, image, x, y, w, h);
    //Set Next Row
    y = y + 72;
    count++;
}

    stamper.Close();
    return true;
}
catch { return false; }

}

```

```

}
public class Avery5160BarcodeLabels
{
    public Avery5160BarcodeLabels()
    {

        Template = new Avery5160TextLabelTemplate();
    }
    public Avery5160BarcodeLabels(PdfTemplate template)
    {

        Template = template;
    }

    public PdfTemplate Template = null;
    public List<string> Codes;
    public bool Create(Stream output)
    {
        try
        {
            PdfReader reader = new PdfReader(Template.TemplateBytes);
            PdfStamper stamper = new PdfStamper(reader, output);
            var x = 17;
            var y = 38;
            var w = 183;

```



```

var h = 68;
int count = 0;
List<iTextSharp.text.Image> images = new List<iTextSharp.text.Image>();
foreach (var code in Codes)
{
    // images.Add(createBarcode(stamper, code));
    images.Add(createVectorBarcode(stamper, code, 3, 3)); //set scale
}

foreach (var image in images)
{
    if (count == 10 || count == 20) //Reset Column
    {
        x = x + 198;
        y = 38;
    }
    stamper = Stammers.ImageStamper(stamper, image, x, y, w, h, 3, 3); //match scale
    //Set Next Row
    y = y + 72;
    count++;
}

stamper.Close();
return true;
}
catch { return false; }
}
public bool Create(string file)
{
    try
    {
        PdfReader reader = new PdfReader(Template.TemplateBytes);
        PdfStamper stamper = new PdfStamper(reader, File.Open(file, FileMode.Create));
        var x = 17;
        var y = 38;
        var w = 183;
        var h = 68;
        int count = 0;
        List<iTextSharp.text.Image> images = new List<iTextSharp.text.Image>();
        foreach (var code in Codes)
        {
            // images.Add(createBarcode(stamper, code));

```

```

        images.Add(createVectorBarcode(stamper, code, 3, 3)); //set scale
    }

    foreach (var image in images)
    {
        if (count == 10 || count == 20) //Reset Column
        {
            x = x + 198;
            y = 38;
        }
        stamper = Stampers.ImageStamper(stamper, image, x, y, w, h, 3, 3); //match scale
        //Set Next Row
        y = y + 72;
        count++;
    }

    stamper.Close();
    return true;
}
catch { return false; }

}

```

```

}
public class Avery5160GraphicLabels
{
    public Avery5160GraphicLabels()
    {

        Template = new Avery5160TextLabelTemplate();
    }
    public Avery5160GraphicLabels(PdfTemplate template)
    {

        Template = template;
    }

    public PdfTemplate Template = null;
    public List<System.Drawing.Image> Images;
    public bool Create(Stream output)
    {

```

```

try
{
    PdfReader reader = new PdfReader(Template.TemplateBytes);
    PdfStamper stamper = new PdfStamper(reader, output);
    var x = 17;
    var y = 38;
    var w = 183;
    var h = 68;
    int count = 0;
    foreach (var image in Images)
    {
        if (count == 10 || count == 20) //Reset Column
        {
            x = x + 198;
            y = 38;
        }
        stamper = Stammers.ImageStamper(stamper, image, x, y, w, h);
        //Set Next Row
        y = y + 72;
        count++;
    }

    stamper.Close();
    return true;
}
catch { return false; }

}

public bool Create(string file)
{

    try
    {
        PdfReader reader = new PdfReader(Template.TemplateBytes);
        PdfStamper stamper = new PdfStamper(reader, File.Open(file, FileMode.Create));
        var x = 17;
        var y = 38;
        var w = 183;
        var h = 68;
        int count = 0;
        foreach (var image in Images)

```

```

    {
        if (count == 10 || count == 20) //Reset Column
        {
            x = x + 198;
            y = 38;
        }
        stamper = Stampers.ImageStamper(stamper, image, x, y, w, h);
        //Set Next Row
        y = y + 72;
        count++;
    }

    stamper.Close();
    return true;
}
catch { return false; }

}

```

```

}

```

```

public class Stampers
{
    public static PdfStamper ImageStamper(PdfStamper stamper, System.Drawing.Image img,
float x, float y, float w, float h, int scalex = 1, int scaley = 1, int page = 1)
    {

        var pdfContentByte = stamper.GetOverContent(page);
        try
        {
            iTextSharp.text.Image image = iTextSharp.text.Image.GetInstance(img,
System.Drawing.Imaging.ImageFormat.Bmp);
            image.ScaleAbsolute(w * scalex, h * scaley);
            image.SetAbsolutePosition(x, y);
            pdfContentByte.AddImage(image);
        }
        catch { }
        return stamper;
    }
}

```

```

public static PdfStamper ImageStamper(PdfStamper stamper, string img, float x, float y, float

```

```

w, float h, int scalex=1, int scaley=1 , int page = 1)
{
    using (Stream inputStream = new FileStream(img, FileMode.Open,
        FileAccess.Read, FileShare.Read))
    {
        var pdfContentByte = stamper.GetOverContent(page);
        iTextSharp.text.Image image = iTextSharp.text.Image.GetInstance(inputStream);
        image.ScaleAbsolute(w*scalex, h*scaley);
        image.SetAbsolutePosition(x, y);
        pdfContentByte.AddImage(image);
    }
    return stamper;
}

public static PdfStamper ImageStamper(PdfStamper stamper, iTextSharp.text.Image image,
float x, float y, float w, float h, int scalex = 1, int scaley = 1, int page = 1)
{
    var pdfContentByte = stamper.GetOverContent(page);
    image.ScaleAbsolute(w * scalex, h * scaley);
    image.SetAbsolutePosition(x, y);
    pdfContentByte.AddImage(image);
    return stamper;
}

}

public class PdfTemplate
{
    public string m_template =
"JVBERi0xLjUNCiW1tbW1DQoxIDAgb2JqDQo8PC9UeXBIL0NhdGFsb2cvUGFnZXMGMiAwIFlvT
GFuZyhlbi1VUykgL1N0cnVjdFRyZWVSb290IDcgMCBSL01hcmtJbmZvPDwvTWfya2VklHRydW
U+Pj4+DQplbmRvYmoNCjlgMCBvYmoNCjw8L1R5cGUvUGFnZXMvQ291bnQgMS9LaWRzWyAz
IDAgUI0gPj4NCmVuZG9iag0KMyAwIG9iag0KPDwvVHlwZS9QYWdlL1BhcmVudCAyIDAgUi9SZX
NvdXJjZXM8PC9Gb250PDwvRjEgNSAwIFI+Pi9Qcm9jU2V0Wy9QREYvVGv4dC9JbWFnZUIvS
W1hZ2VDL0ltYWdlISV0gPj4vTWVkaWFCb3hbIDA gMCA1OTUuMzlgODQxLjkyXSAvQ29udGVud
HMgNCAwIFlvR3JvdXA8PC9UeXBIL0dyb3VwL1MvVHJhbnNwYXJlbnN5L0NTL0RldmliZVJHqj4
+L1RhYnMvUy9TdHJ1Y3RQYXJlbnRzIDA+Pg0KZW5kb2JqDQo0IDAgb2JqDQo8PC9MZW5ndG
ggMTAxPj4NCnN0cmVhbQ0KIC9QIDw8L01DSUQgMC9MYW5nIChkZS1DSCK+PiBCREMGQIQ
NCi9GMSA5Ljk2IFRmDQoxIDA gMCAxIDM2IDc5Ni42OCBUbQ0KMCBnDQowIEcNCi9QIDw8L01DSUQ
DQpFVA0KIEVNQyANCmVuZHN0cmVhbQ0KZW5kb2JqDQo1IDAgb2JqDQo8PC9UeXBIL0Zvb
nQvU3VidHlwZS9UcnVlVHlwZS9OYW1lL0YxL0Jhc2VGB250L0FyaWFsTVQvRW5jb2Rpbm
cvV2luQW5zaUVuY29kaW5nL0ZvbNREZXNjcmlwdG9yIDYgMCBSL0ZpcnN0Q2hhciAzMi9MYXN0Q2hh
ciAzMi9XaWR0aHMgMTUgMCBSPj4NCmVuZG9iag0KNiAwIG9iag0KPDwvVHlwZS9Gb250RGV
zY3JpcHRvci9Gb250TmFtZS9BcmllbE1UL0ZsYWdzIDMyL0l0YWxpY0FuZ2xIDAvQXNjZW50ID

```

kwNS9EZxNjZW50IC0yMTAvQ2FwSGVpZ2h0IDcyOC9BdmdXaWR0aCA0NDEvTWF4V2lkdGgg
MjY2NS9Gb250V2VpZ2h0IDQwMC9YSGVpZ2h0IDI1MC9MZWFkaW5nIDMzL1N0ZW1WIDQ0L0
ZvbnRCQm94WyAtNjY1IC0yMTAgMjAwMCA3MjhdID4+DQplbmRvYmoNCjEzIDAgb2JqDQo8PC
9UeXBIL09ialN0bS9OIDYvRmlyc3QgMzYvRmlsdGVyL0ZsYXRIRGVjb2RIL0xlbmd0aCAyODQ+P
g0Kc3RyZWFTDQp4nG1RwYrCMBC9C/7D/MEk3YoVRJBv2UUsP3sQTzEdrYttonEFPTvN2kr7
WEvybyZ995MjpwDA85gwWFpbwYB8GAFK/C9BXAPfD+A9Rojx2EQY4IRpq87YWJ0m5l9TQ0e
z8AugFEBH46z2cxnnWT5Vght/hNx1zu+wCCYMFJNFctIMFY1ncTdTuWsrBHJrmgHdAlnwnqTST
GkpznSC/hgfLBOUhnC0B17mY8gtdSremJCmcEvEjnpPnaad/wt60pSUgo3n0tspXUQplJywNpUv8
IGHfpR+nZV6oY7lbWNnanLPEoi06/iJDKtJvzItOcE7ypRq2KSSOoqpwm372NphRYNHqqi1TS8N
Wybx9l9qTfudtz0fPYHJuqasw0KZW5kc3RyZWFTDQplbmRvYmoNCjE0IDAgb2JqDQo8PC9Qcm9
kdWNlciJ+/wBNAGkAYwByAG8AcwBvAGYAdACuACAAVwBvAHIAZAAGADIAMAAxADMpIC9Dc
mVhdG9yKP7/AE0AaQBjAHIAbwBzAG8AZgB0AK4AIABXAG8AcgBkACAAMgAwADEAMykgL0N
yZWFOaW9uRGF0ZShEOjIwMTgwMTE1MjEwMTM4LTA4JzAwJykgL01vZERhdGUoRDoyMDE4
MDExNTIxMDEzOC0wOCcwMCCpID4+DQplbmRvYmoNCjE1IDAgb2JqDQpbIDI3OF0gDQplbmR
vYmoNCjE2IDAgb2JqDQo8PC9UeXBIL1hSZWYvU2l6ZSAxNi9XWyAxIDQgMI0gL1Jvb3QgMSAw
IFlvSW5mbyAxNCAwIFlvSURbPDBERDNGMTgzNzJBQTU2NDZFBQkM0N0MxRDQyOUMwRDJ
GPjwwREQzRjE4MzcyQUE1NjQxQUJDNDdDMUQ0MjIDMEQyRj5dIC9GaWx0ZXlvRmxhdGVEZ
WNvZGUvTGvUz3RoIDY3Pj4NCnN0cmVhbQ0KeJxjYACC//8ZgaQgAwOlqoFQW8AU4wIwxWQ
Doe4AMVCOI4EJQjFDKBYIxiQihoePygRqY94H1sdqBKTyUBgYA5kMGgw0KZW5kc3RyZWFTDQ
plbmRvYmoNCnhyZWYNCjAgMTcNCjAwMDAwMDAwMDcgNjU1MzUgZg0KMDAwMDAwMDAx
NyAwMDAwMCBuDQowMDAwMDAwMTI0IDAwMDAwIG4NCjAwMDAwMDAxODAgMDAwMDAg
bg0KMDAwMDAwMDQxNiAwMDAwMCBuDQowMDAwMDAwNTcyIDAwMDAwIG4NCjAwMDAw
MDA3MzIgMDAwMDAgbg0KMDAwMDAwMDAwOCA2NTUzNSBmDQowMDAwMDAwMDA5IDY
1NTM1IGYNCjAwMDAwMDAwMTAgNjU1MzUgZg0KMDAwMDAwMDAxMSA2NTUzNSBmDQow
MDAwMDAwMDEyIDY1NTM1IGYNCjAwMDAwMDAwMTMgNjU1MzUgZg0KMDAwMDAwMDAw
MCA2NTUzNSBmDQowMDAwMDAxMzQyIDAwMDAwIG4NCjAwMDAwMDE1NDYgMDAwMDAg
bg0KMDAwMDAwMTU3MyAwMDAwMCBuDQp0cmFpbGVyDQo8PC9TaXplIDE3L1Jvb3QgMSA
wIFlvSW5mbyAxNCAwIFlvSURbPDBERDNGMTgzNzJBQTU2NDZFBQkM0N0MxRDQyOUMwRD
JGPjwwREQzRjE4MzcyQUE1NjQxQUJDNDdDMUQ0MjIDMEQyRj5dID4+DQpzdGFydHhyZWYN
CjE4NDANCiUIRU9GDQp4cmVmDQowIDANCnRyYWIsZXINCjw8L1NpemUgMTcvUm9vdCAxID
AgUi9JbmZvIDE0IDAgUi9JRFs8MEREM0YxODM3MkFBNTY0MUFCQzQ3QzFENDI5QzBEMkY
+PDBERDNGMTgzNzJBQTU2NDZFBQkM0N0MxRDQyOUMwRDJGPI0gL1ByZXlYgMTg0MC9YU
mVmU3RtIDE1NzM+Pg0Kc3RhcnR4cmVmDQoyMzM2DQoIJuVPRg==";

```
public byte[] TemplateBytes
{
    get
    {
        return Convert.FromBase64String(m_template);
    }
}

public string TemplateBase64 { get { return m_template; } set { m_template = value; } }
}
```

```

public static void ToImage(string source)
{
    string filenoext = Path.GetFileNameWithoutExtension(source);
    string directory = source.Replace(Path.GetFileName(source), "");
    var dest = directory + filenoext + ".png";
    var bb = 0;
    GhostscriptPngDevice img = new
GhostscriptPngDevice(GhostscriptPngDeviceType.Png16m);
    img.GraphicsAlphaBits = GhostscriptImageDeviceAlphaBits.V_4;
    img.TextAlphaBits = GhostscriptImageDeviceAlphaBits.V_4;
    img.ResolutionXY = new GhostscriptImageDeviceResolution(800, 600);
    img.InputFiles.Add(source);
    img.Pdf.FirstPage = 1;
    img.Pdf.LastPage = 1;
    img.PostScript = String.Empty;
    img.OutputPath = dest;
    img.Process();
}

public static void CombineMultiplePDFs(string[] fileNames, string outFile)
{
    // step 1: creation of a document-object
    Document document = new Document();

    // step 2: we create a writer that listens to the document
    PdfCopy writer = new PdfCopy(document, new FileStream(outFile, FileMode.Create));
    if (writer == null)
    {
        return;
    }

    // step 3: we open the document
    document.Open();

    foreach (string fileName in fileNames)
    {
        // we create a reader for a certain document
        PdfReader reader = new PdfReader(fileName);
        reader.ConsolidateNamedDestinations();

        // step 4: we add content
    }
}

```

```

    for (int i = 1; i <= reader.NumberOfPages; i++)
    {
        PdfImportedPage page = writer.GetImportedPage(reader, i);
        writer.AddPage(page);
    }

    PRAcroForm form = reader.AcroForm;
    if (form != null)
    {
        writer.CopyDocumentFields(reader);
    }

    reader.Close();
}

// step 5: we close the document and writer
writer.Close();
document.Close();
}

public static void CombineMultiplePDFs(List<byte[]> bytelist, string outFile)
{
    // step 1: creation of a document-object
    Document document = new Document();

    // step 2: we create a writer that listens to the document
    PdfCopy writer = new PdfCopy(document, new FileStream(outFile, FileMode.Create));
    if (writer == null)
    {
        return;
    }

    // step 3: we open the document
    document.Open();

    foreach (var bytes in bytelist)
    {
        // we create a reader for a certain document
        PdfReader reader = new PdfReader(bytes);
        reader.ConsolidateNamedDestinations();

        // step 4: we add content
        for (int i = 1; i <= reader.NumberOfPages; i++)

```



```

    {
        PdfImportedPage page = writer.GetImportedPage(reader, i);
        writer.AddPage(page);
    }

    PRAcroForm form = reader.AcroForm;
    if (form != null)
    {
        writer.CopyDocumentFields(reader);
    }

    reader.Close();
}

// step 5: we close the document and writer
writer.Close();
document.Close();
}
public static void CombineMultiplePDFs(List<byte[]> bytelist, Stream output)
{
    // step 1: creation of a document-object
    Document document = new Document();

    // step 2: we create a writer that listens to the document
    PdfCopy writer = new PdfCopy(document, output);
    if (writer == null)
    {
        return;
    }

    // step 3: we open the document
    document.Open();

    foreach (var bytes in bytelist)
    {
        // we create a reader for a certain document
        PdfReader reader = new PdfReader(bytes);
        reader.ConsolidateNamedDestinations();

        // step 4: we add content
        for (int i = 1; i <= reader.NumberOfPages; i++)
        {

```

```

        PdfImportedPage page = writer.GetImportedPage(reader, i);
        writer.AddPage(page);
    }

    PRAcroForm form = reader.AcroForm;
    if (form != null)
    {
        writer.CopyDocumentFields(reader);
    }

    reader.Close();
}

// step 5: we close the document and writer
writer.Close();
document.Close();
}

public static bool ImageToPdf(string source, string dest)
{

    Document doc = new Document(PageSize.A4, 10f, 10f, 10f, 10f);

    PdfWriter writer = PdfWriter.GetInstance(doc, new FileStream(source, FileMode.Create));

    doc.Open();

    try

    {

        Paragraph paragraph = new Paragraph("Getting Started iTextSharp.");

        string imageURL = HttpContext.Current.Server.MapPath(".") + "/image2.jpg";

        iTextSharp.text.Image jpg = iTextSharp.text.Image.GetInstance(imageURL);

        //Resize image depend upon your need

        jpg.ScaleToFit(140f, 120f);

        //Give space before image

```

```
jpg.SpacingBefore = 10f;
```

```
//Give some space after the image
```

```
jpg.SpacingAfter = 1f;
```

```
jpg.Alignment = Element.ALIGN_LEFT;
```

```
doc.Add(paragraph);
```

```
doc.Add(jpg);
```

```
}
```

```
catch (Exception ex)
```

```
{ return false; }
```

```
finally
```

```
{
```

```
    doc.Close();
```

```
}
```

```
return true;
```

```
}
```

```
public static System.Drawing.Size ResizeKeepAspect(System.Drawing.Size src, decimal  
maxWidth, decimal maxHeight)
```

```
{
```

```
    decimal rnd = Math.Min(maxWidth / (decimal)src.Width, maxHeight / (decimal)src.Height);
```

```
    return new System.Drawing.Size((int)Math.Round(src.Width * rnd),
```

```
(int)Math.Round(src.Height * rnd));
```

```
}
```

```
public static bool ImageToPdf(System.Drawing.Image image, string dest)
```

```
{
```

```

try
{

    using (Document document = new Document(PageSize.A4))
    {
        try
        {

            var pagesize = document.PageSize;
            var width = pagesize.Width;
            var height = pagesize.Height;
            //image = ResizeImage(image, width, 600, InterpolationMode.HighQualityBicubic,
true);
            var newsize = ResizeKeepAspect(image.Size, Convert.ToDecimal(width),
Convert.ToDecimal(width / 2));
            PdfWriter.GetInstance(document, new FileStream(dest, FileMode.Create));
            document.Open();
            iTextSharp.text.Image pdfImage = iTextSharp.text.Image.GetInstance(image,
System.Drawing.Imaging.ImageFormat.Jpeg);
            pdfImage.ScaleAbsolute(new iTextSharp.text.Rectangle(newsize.Width,
newsize.Height));
            document.Add(pdfImage);
            document.Close();
        }
        catch
        {
            document.Close();
            return false;
        }
    }
    return true;
}
catch { return false; }
}

public static bool Merge(List<String> InFiles, String OutFile, bool flatten=false)
{
    try
    {
        using (FileStream stream = new FileStream(OutFile, FileMode.Create))
        using (Document doc = new Document())
        using (PdfCopy pdf = new PdfCopy(doc, stream))
        {

```

```

doc.Open();

PdfReader reader = null;
PdfImportedPage page = null;

//fixed typo
InFiles.ForEach(file =>
{
    if (flatten)
    {
        Flatten(file, file);
    }

    reader = new PdfReader(file);

    for (int i = 0; i < reader.NumberOfPages; i++)
    {
        page = pdf.GetImportedPage(reader, i + 1);
        pdf.AddPage(page);
    }

    pdf.FreeReader(reader);
    reader.Close();
});
}
return true;
}
catch { return false; }
}
public static bool Flatten(string filename, string dest)
{
    try
    {
        PdfReader reader = new PdfReader(filename);
        using (MemoryStream ms = new MemoryStream())
        {
            PdfStamper stamper = new PdfStamper(reader, ms);
            //Method 1
            stamper.FormFlattening = true;
            //Method 1
            // string[] fields = stamper.AcroFields.Fields.Select(x => x.Key).ToArray();
            // for (int key = 0; key <= fields.Count() - 1; key++)

```

```

        // {
        //     stamper.AcroFields.SetFieldProperty(fields(key), "setfflags",
PdfFormField.FF_READ_ONLY, null);
        // }
        stamper.Writer.CloseStream = false;
        stamper.Close();
        reader.Close();
        using (FileStream fs = new FileStream(dest, FileMode.Create))
        {
            ms.Position = 0;
            ms.CopyTo(fs);
        }

    }

    return true;
}
catch { return false; }
}

```

```

}

```

```

public static class PdfExtensions
{
    public static PdfStamper StampImage(this PdfStamper stamper, string img, float x, float y, float
w, float h, int page = 1)
    {
        using (Stream inputImageStream = new FileStream(img, FileMode.Open, FileAccess.Read,
FileShare.Read))
        {
            var pdfContentByte = stamper.GetOverContent(page);
            iTextSharp.text.Image image = iTextSharp.text.Image.GetInstance(inputImageStream);
            image.ScaleAbsolute(w, h);
            image.SetAbsolutePosition(x, y);
            pdfContentByte.AddImage(image);
        }
        return stamper;
    }
}

```