Canvas Drawing:

Circle

Activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

MainActivity.java

```
package com.example.circle;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;

import android.app.Activity;
import android.os.Bundle;

public class MainActivity extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(new customview(this));
    }

}
```

CustomView.java

```
package com.example.circle;

import android.content.Context;
import android.graphics.Canvas;
import android.graphics.Color;
import android.graphics.Paint;
import android.view.View;

public class customview extends View {

    private Paint paint;

    public customview(Context context) {
        super(context);

        // create the Paint and set its color
        paint = new Paint();
        paint.setColor(Color.GRAY);
    }

    @Override
    protected void onDraw(Canvas canvas) {
        canvas.drawColor(Color.BLUE);
        canvas.drawCircle(200, 200, 100, paint);
    }

}
```

Oval

Activity_main.xml

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    >

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
       />
</RelativeLayout>
```

Mainactivity.java

```
package com.example.oval;
import android.app.Activity;
import android.os.Bundle;

public class MainActivity extends Activity {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(new  Myview(this));
    }
}
```

myview.java

```
package com.example.oval;
import android.content.Context;
import android.graphics.Canvas;
import android.graphics.Color;
import android.graphics.Paint;
import android.graphics.RectF;
import android.util.AttributeSet;
import android.view.View;

public class Myview extends View {

    public Myview(Context context) {
        super(context);
    }

    public Myview(Context context, AttributeSet attrs) {
        super(context, attrs);
```

```
    }

    public Myview(Context context, AttributeSet attrs, int defStyle) {
        super(context, attrs, defStyle);
    }

    @Override
    protected void onDraw(Canvas canvas) {
// TODO Auto-generated method stub
        super.onDraw(canvas);

        Paint paint = new Paint();
        paint.setStyle(Paint.Style.STROKE);
        paint.setColor(Color.GRAY);
     //   RectF oval1 = new RectF(0, 0, getWidth(), getHeight());
    //    canvas.drawOval(oval1, paint);

        paint.setStyle(Paint.Style.FILL);
   //     paint.setColor(Color.RED);
  //      RectF oval2 = new RectF(50, 50, 150, 150);
//        canvas.drawOval(oval2, paint);
        paint.setColor(Color.BLUE);
        RectF oval3 = new RectF(250, 50, 350, 400);
        canvas.drawOval(oval3, paint);
    }
}
```

Triangle

Activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:gravity="center"
    android:layout_marginTop="30dp"
    tools:context=".MainActivity">
    <ImageView
        android:layout_width="100dp"
        android:layout_height="100dp"
        android:background="@drawable/background"/>
</LinearLayout>
```

Background.xml

```
<vector xmlns:android="http://schemas.android.com/apk/res/android"
    android:height="100dp"
    android:width="100dp"
    android:viewportWidth="24"
    android:viewportHeight="24"
    android:layout_height="match_parent">
    <path android:fillColor="#000" android:pathData="M1,21H23L12,2" />
</vector>
```

Mainactivity.java

```
package com.example.trianle;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
}
```

Rectangle

Activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >

    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="Text" />

</LinearLayout>
```

Mainactivity.java

```
package com.example.rectangle;

import android.app.Activity;
import android.os.Bundle;

public class MainActivity extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(new Secondactivity(this));
    }

}
```

secondactivity.java

```
import android.content.Context;
import android.graphics.Canvas;
import android.graphics.Color;
import android.graphics.Paint;
import android.graphics.Rect;
import android.view.View;

public class Secondactivity extends View {

    private Rect rectangle;
    private Paint paint;

    public Secondactivity(Context context) {
        super(context);
        int x = 150;
        int y = 150;
```

```
        int sideLength = 250;
        int sl=350;

        // create a rectangle that we'll draw later
        rectangle = new Rect(x, y, sideLength, sl);

        // create the Paint and set its color
        paint = new Paint();
        paint.setColor(Color.RED);
    }

    @Override
    protected void onDraw(Canvas canvas) {
        canvas.drawColor(Color.GRAY);
        canvas.drawRect(rectangle, paint);
    }

}
```

Line

Activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:id="@+id/relativeLayout"
    tools:context=".MainActivity">
    <ImageView
        android:id="@+id/imageView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerInParent="true"/>
    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Draw Line"
        android:layout_alignParentBottom="true"
        android:layout_centerHorizontal="true"
        android:layout_marginBottom="70dp"/>
</RelativeLayout>
```

Mainactivity.java

```
package com.example.linedrawing;
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.content.Context;
import android.content.res.Resources;
import android.graphics.Color;
import android.graphics.Paint;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.RelativeLayout;
import android.graphics.Canvas;
import android.graphics.Bitmap;
public class MainActivity extends AppCompatActivity {
    Context context;
    Resources resources;
    RelativeLayout relativeLayout;
    Button button;
    ImageView imageView;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        context = getApplicationContext();
        resources = getResources();
        relativeLayout = findViewById(R.id.relativeLayout);
        button = findViewById(R.id.button);
```

```
        imageView = findViewById(R.id.imageView);
        button.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Bitmap bitmap = Bitmap.createBitmap(10, 200,
Bitmap.Config.ARGB_8888);
                Canvas canvas = new Canvas(bitmap);
                canvas.drawColor(Color.BLUE);
                Paint paint = new Paint();
                paint.setColor(Color.GREEN);
                paint.setStyle(Paint.Style.STROKE);
                paint.setStrokeWidth(10);
                paint.setAntiAlias(true);
                int offset = 50;
                canvas.drawLine(
                        offset, canvas.getHeight() / 2, canvas.getWidth() -
offset, canvas.getHeight() / 2, paint);
                imageView.setImageBitmap(bitmap);
            }
        });
    }
}
```

Shapes using bitmapfactory methods

Activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

MainActivity.java

```
package com.example.bitmapfactory;

import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.util.Base64;
import android.util.DisplayMetrics;
import android.view.ViewGroup;
import android.widget.ImageView;
import android.widget.LinearLayout;

public class MainActivity extends AppCompatActivity {


    ImageView ivOne, ivTwo, ivThree;
    BitmapFactory.Options bmfOptions;
    Bitmap bm;


    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        /*
            create a linear layout with vertical orientation
```

```
         */
        LinearLayout ll = new LinearLayout(this);
        ll.setOrientation(LinearLayout.VERTICAL);

        /*create three imageViews*/
        ivOne = new ImageView(this);
        ivTwo = new ImageView(this);
        ivThree = new ImageView(this);

        /*
            set scale type of the imageViews to center
               No scaling is performed on the bitmap
               bitmap is just centered
         */
        ivOne.setScaleType(ImageView.ScaleType.CENTER);
        ivTwo.setScaleType(ImageView.ScaleType.CENTER);
        ivThree.setScaleType(ImageView.ScaleType.CENTER);




        /*Set the layoutparam of the imageViews
         *   width : match parent
         *   height: wrap content
         *   weight : 1
         */
        ll.addView(ivOne, new
LinearLayout.LayoutParams(ViewGroup.LayoutParams.MATCH_PARENT,
ViewGroup.LayoutParams.MATCH_PARENT, 1));
        ll.addView(ivTwo, new
LinearLayout.LayoutParams(ViewGroup.LayoutParams.MATCH_PARENT,
ViewGroup.LayoutParams.MATCH_PARENT, 1));
        ll.addView(ivThree, new
LinearLayout.LayoutParams(ViewGroup.LayoutParams.MATCH_PARENT,
ViewGroup.LayoutParams.MATCH_PARENT, 1));


        //decode a base 64 encoded image into bytes
        // the original image is 500px by 500px
        byte[] bytes =
Base64.decode("iVBORw0KGgoAAAANSUhEUgAAAfQAAAH0CAYAAAG80e8cAAAABGdBTUEAALGPC/xhBQA
AQABJREFUeAHsnQd8FcXWwM+FhBQIECD0jhQRUHov0rEAoiKgKCBIQH3W96kP21MR27Oh9ACCUkKHhE6AE
EISSgKETiCkAiGNkEbaN7Pxxptwy+7ebbN7hl+4e3dnzpzzP3vubJk9a1q4cGEJGLS4UUlt9fX0NaX4lQ1r
9t9FovFG9j55Hz1cg0LN7N+jQ/mFu7UcffgCpd+5AREQEt85//Xp4+1//qtCCwa90nBdaHnm4ndAmmqwvK
uajz19g0M0PqizK+AfFsLkGjWfTb85rbWjPcyc2zjMEiH7pJWh96VI5UfeqVoXahw6VW6elL04bn9+9O2d
PaytWVcvOBvN2t+PHrdRQd5Xo3b6goKDMMD4mUAjb/f35VHVYZ0+XaQ7r8Kkg2vjiPn34yC9XZ8T330NIU
FC5dWK+eMRnQrDPs2Kalmsjynjzrlx0Es8v3T/4gGdN+asJNj59wACntXIGnqXHjwYfcUoXwcZ75uY61aG
Yxjk5ORDw+OwHdvWiZ3/m1q394icxYkGw8aJ6sdJoxZIlVtY+uOrooWA40WwyVI++9eDGv9c0mh/yABibl
S02CDI+4rXXLJo6tzhp6VJeAvoO4hdmrpvf5SXPspIg4x+NjLRsq9jygJRN9vsymaB3/77261jZKsh4K+0
VW2UPwIDbG0XpIcj40126iOpEq40EGd9j8WLJ7FgzY4ZkssQKEmS82E6stZsq4Y+nNfl81gk2PsfDg49cy
ev8sWQZJzP+9T5A49/ebwDfzgUb7x0czFe2zXr0DK/H0ok2t1vb0OS7g5zBL37+XtlmCiB/5fSy70IXBBt
PO3Dm9PT4t9+W6SgEwOCrf5W1s1wY9uQoy6+ClkUZT3uoFBoqqCNaeff770O/wYPhWESY4LZyNBBtvKurq
6A9gO4tY154gbPhrdO/lNmyNWB72bLSC6KNNytKjaJ/V9q2Na8q+6SXsczby1ZWWPg6eW2FNcp9dfoylln
VDn/+aV4s+3QrW/pnYcQS8gNl+ue7mktOe16o8umm7Aea9Fn60gPrlFihqPEZGRlWbSqEIqvr5V6pqPHDN
8yyac+1a9dsbpNrg6LG2zNiwoE59jbLsk0x499c+oksBjgjVDHjw+GqQz0nLnnLYR0pKyhmPB+lY0y3+VS
TrI4ixgs5hi8qUu6XXxHjhbiq93LlxnzZjZ+/bIEQ2xWtK7vxq0uE31WZt0LcTQih5GQ3XqhCtP6Wwggxz
QS3kdV4IT90gjWXoIGsxjujnxLgZDPeb9UKZ2xXpK1sxi/O3+u0AYuX8buZKbYj2YwXq5Bl07+Sg5ZfJV+
WxXgl4lUKErIYL4ViZhlygpTc+GPh2rgsbYZn71Ny4986889laXsdC9kWsDtQSHXedSU3nnfPAip+Ef/gl
WEBzW1WldT4EUvJZWmGiqTGp8ODl6WlYtFv6WSpRJXJkcz49PT0MqFyLNyHQsnFukgl0dvbGyJmqHfrSYw
```

```
dknleTOdqtzG08SajPkJuog+7qb3rqdG/YXd3NFyN3U3NPtHjatJXo2+rHjc/Hv7xnP9wOj0xcgT3uX3bV
pj/6y+wZfNmiLnq+M6qGgbx7lPos9v/++F7oU00WR/Hcd67iE4qWo1xndhm1ww03C4eHW6U5KJE5MmT0N5
KnqzNs2bBxGnTNInNqV199bJl3NPQ1oym1o5buJDbfuA/pccDUhFYvtj5+3CihzMxz7PSGdRSFPo8rLOPi
YjyuBijqcFi21nC2rJmveVX0cuCDd8/x7npnc4aX/ut0sQB2556W7TRtKFgw/vv3etUh1I19g6Pd0qUoBg
P+vBD6HvggFMdmhsLifeV5DGwlnN2mZs+8OkR8AF079njgfX2Vggy3Nnd1FIRvoZbPthv2d7aspAfPMG7u
rUO5VzH1xi+9cy6at5wqqjHjv8z62v1U6jRVAgThnfv1RNKKll/ashrl7iDIyYMpx667+1BPx4onbt1fWA
dnxXMGO6WmmPVnlVL/Kyud7RSkOFHhwxxJE+R7cVVKpf10+h7cVPIBBk++Jtvyjp0ZoEOZWLvr9MfskGJ/
tA1dhWngmuGuFQyggynPR0ZPtwZm8vajtg4u2yZz0Je/WrlTkyqksdGxfyam/sSdABjbuTMgQz19hd+P0B
A8UlVJxUI9jg1nu9RlxmU+dPcjhpNy5vLPjFvUvxTlOFUS2qE/8yZvBQOIeFhNtqyQXiJejclRO3qlsrT5
ahTp+BhKxCsXXqqOJVTrTk0khheEYS97xUNp3XVMF70rm7POFvbFi2VLreNrT74rlfU8OVwyKpev/kttLp
ezpWKGm7LkFXFwbY2ybZeMcOtxbZsVvEQrJjhjnRRGowihm8J2ObIbsW3K2L4vOR1vAxbvnolr3pSVFLEc
L6KLsrbw7eq0/VkN7yvSqlVHJGR3fACgWlVlPqRk9VwNVKpOPK0ebushotNoxIh0V1Vs5HWPmU13FqHfNa
9EfUjn2pO1ZHN8AkKp00RSkE2w685mTJl1FJ5M4vKYnhhofNPEKXCPaFOFFRfFsP7rJDmubG7d+8KMkZIZ
VkMF6KAvbpD1/O7pmdPhq1tkhv+5fIfbPWlqfWSG76jqPTSsVRWvrPsc6lElZMjueHlpEvw5WjJJQmkPCh
CUsOVOs5+0AzhayQ1XHj3/FrIAVQyw+VOd8IPEf9akhkud6qTBX6L+FvFo6bid1J46KRIFck8roi2EnZiW
MO5x6iN+jJqCXcgpkQZdm9nyksSK4tOlxgoC+LQ6Sx4SWId0ekSA2VBHDqdBS9JrKNgp5szSlTUY8/u3RV
Xlft+XIFbg+U6xC82CfB2ei55+Sl1uGuVKtwnXX6sU0cIDwuD06ej4MMP/o9bv3jRQribmQm9enSHTh0e4
dbNJK/dm/rKy5wSz417BoqLi20qhBvkJ4Dn6fIz1lwPvCNdc5qjQqIJoNNFo2O3ITqdXd+J1hydLhoduw0
lyYwkl/n0KD8pKQlSUlKgZs2aULduXaCPnmJxjoAmnL72119h3OrVVi3xIWvpn7nkmxcsPmNat4b2a9ZYr
MFFewRUc3p2797g8vc80XH2NOSxrdWVK+WSTuWTjD7Vq1fn0VK5KhcvXoTb/ec49YC8VNoqOqbfu3ePcw5
9UN/scKkMsZTjRnJf0D4WLZJ2no1lH0KXqcNp+Wver0KbSl5fMadTJ7g+/rjkBtgTONXPr9wvgL26cm6zT
HHV5MfDcOfOHTm7cyhbdqfTN6I6k4LDoQU8KtD+/yQ7gBpl2YIHf23OPyzfJHY+NsrqdHr0XdirFx89ZK/
zPPmp/2OJ82lBhSian58PbT7bZ7XJweYTra5XYqWsB3IFPXsqYQPvPiYsXQrw2mu86/OpaPnTzae+uU7l7
Ptgr23StB4w4dsPzNUl/ZQt0lcqHFV8qWT278+3Kq96NF1V4mvS/ppRmXI5nBolm9Mn0qjSYHHPy5Ncq4l
z/y3JqVjC7D6SyHFkoKw/744619t2GqF/ffkTNPk1RLBpziS4E9qZbJEuVBG91H/xk3fg5lRhaZWVdDjlL
JvT15LZMloseR7WU8VKqWv9FRGCxF26JM+DmbaUkM3pUyQ+SrZlgND1NYKDhTaRvf7pxZtk78OyA9mcTjt
xDQ+37Ev15XUK7Ii2opb+hNO/q58Pe4BD/dXSPsv+QAcVVsjq9EqVKoELmTiphbKBvBTnFTLkbNqxBV5Y8
i/ZVDqzcGOZ7EZh33GOthyzp73uW7YubWDLsrpKLsh+9F65cmWoTKY/q3kplubYfelvqt/eJG8xIanv/Va
tgFdfnio569wODXifdo3d+H1Z/+fPn4f27duXfZdzQdZIt1Scgi85csRylezLf5HItkyqbJnTYHH+XpAiB
VBFI16ePq3iKl7flXI4VUb2SLe02N3dHYA4nxbL++mWdaRYNt9Pt8RvLU0hTX+kRm5lKWx0RoZikV5Ryar
HjnFRSCNx82Tnck/RmTNUjvmv4gSK+csWgK0UjZbRX1FHvX7X9MMOUsyRo6kpHWVpHFO5O8yZ9q5effyAX
Yr+vD/Qu4MV9Oi/cePG3J+DqjY3O3I4bbit6DgMjQiHnj20dVfQplFOblDt591JvXk1F/LT/ebpn3nJ1EM
l3TpdTCpWITsJy87XpdPpObjYNLQjlkxn2Z+8dNed0+m5Nz0HF1vSTdnAWhpAobbqzulSpB6mKRAzyTP2e
i26crqUY/Iwf1+9+ly+++lKE5Mj3bSUO5HSPOz1p4tIPxZ+DKROtW2G9tayz8yLuvnUhdPfOiPfo0LHSi7
D9l0BunE4NYR5pyvxE/xVwl/odK0QGLFUuXNqJXYupbgyG+n0bazpkK0UJ66ffkuduxuoqLJ2OmPS6RkZG
WDrTbR2bHV6030ohN+XKf+mW6cVryCASacP3zCrghnKff2jJBhiY2OV61CGnphzuhbG1vH7PpLBFcqJZMr
pby77RDkyDnrSws7nQEWbm5lx+paArRBectWmIWpseHHJO2p063SfzDh9XvJ6p42VWsAV001Y+dcfUouVX
R4TTtfyT+mCnN3MZbXWvNP7aPQN65bh2MvvRcuvml/WtNN/81tIzoyLNA+RKqjlX6OKADXr9JiYGFhVHFx
RX01//2YFG5MrNev0iUEfa9rB1pTbXBgOJ08p+wSqNT0crdOk01n6qawIeNZJ7b87XHMPO5w8eRIWdnm/I
kumvlMbunbtqlmdNed0LcPSrBcFKqbJn3eBNmB1gQS4BxgFtsHqjBMw7FvjGfebU+rjz7tT+NhsjE5n029
OaY1Odwofm43R6Wz6zSmt0elO4WOzMTqdTb85pTU63Sl8bDZGp7PpN6e0Rqc7hY/Nxuh0Nv3mlNbodKfws
dlYkNM7tH8Y6LvGKpZ/v/9exVUPfB81YvgD63CFOgQEOZ2+xrpr58c4TeNu3IAB/fpyy9dJKk5anhkzBui
OQf9o+ejD0veKDRsyGOLj48tt4yrgf6oQ4D2Jgjqy1UMPQczVq/DUE09Az149IS0tjVOavjH46NGj5MG+6
2VGzP3qK9gZGAhubu6QnJwMXchMklNkRgkW9Qkocmu14yPt4bnnn4fPPv+v+hajBqCI05GztggIGtO1pTp
qI5YAOl0sOYbbodMZdp5Y1dHpYskx3A6dzrDzxKqOThdLjuF26HSGnSdWdXS6WHIMt0OnM+w8sarzvvYut
gNn2qWnp0NKSgrcv38f6taty/05Iw/blhLQ1GXYlQsWwMQVK3j55vCoUdD3gw+A3vnDIoyA6k5PSEgAn2e
eEaZ1hdo733kHnpk0qcJa/GqLgKpOv9u/P7jl5dnSTfD6JH9/aNGiheB2SjRYvmgJTPN9TYmuHPahyoGc/
6pV3PvUpXQ4tbTh+PGw92Nt5qp56JM9QN+RroWieKTv+eQTGLR7t+y2W743XfbOeHQQ7PMs5DaqDiOi+B2
z8BApuoqikX5q2jRFHE5p5HfvLhqK1A3/mlf6jhmPxLtSixYlTzGnb583Dx45e1aUkmIb5fboIbappO2a/
Hi4TN7mv9TPcauI03dv3QojNm8uM1yphUolJXCUHNmrWVJTU8t1X+dt/3Lf1fiiiNMfnztXDdu4PruFhEA
4+VOrhEz5Sq2ubfYru9Njn37aZudKbXhMxWj3Dot7wMxlCxY9sE7JFbI6nT4Y0eDmTSXtsdnX8kXKgw4Nt
v4L0+azfTb1VGKDrE5PGz1aCRt49fGinx+velJWyn/ld5virD0pZLOyxBtkPU/X0mkT5XZh8WJ4rEsXiRG
WijsRcRyu7jkKDX89Ilj+3UfqQsrELjD+pUmK3EuQzemrly2D8QSylsrdGjXAZ/9+yVRa9+F30NAvXDJ5l
oIGpGyy/Crpsmw/70+tXi2polIIq56ZKYWYMhkTvvm/smUpFyr5/0tKcQ/Iks3pnjk5D3SmxxVSR2Tia72
g3+MDZUUlm9Nl1doJ4RHHjjnR2npT183vWt8gYu3Euf8W0UpYE1mcnp2t7ItvhZh8KTJSSHVedXv37wsJs
/vwqmuvktS/Grb6ksXpd+7csdWf6utrk5fzylEm/ZckZjCZRIv22CHP8YE1hWRxOp3XptVSOy5ONtUG3N4
oSnbcm/2gO3neX6kii9N9fHyU0l9wP3eaNhXcRkgDr13/EVIdSiqZ4KVPlb0pJIvT69SpI8hwJSun1awpa
3eduwl7d8utl7vJqo814bI4XcszVNt27myNg2rram8/p3jfsjhdcSsEdNijd28BtYVXvXz5sqBGrmnKX8+
Qzek5np6CjNdL5fDD1u+sack+2ZweMHmyluzkdKHX3uUubhe1cSvZnp2yOX3y9On2+lVlW+J338neb+0d5
```

632UVylstX1aqyUzenUmLRatdSwyWafct1WtezQNT3X8iu3fOXzYTAo0R9sXXGjefiULLI6vdb27UraYre
vv159ldu+YMlCu/Wk3tjl+h/w6uu+ZWKp4xPeKM20aV4ZHnzUvKjIp6xOd3Nzg+QGDRQxxFEn03xLwa80B
TuqKsn29IEtuciuVq3aA/ImffYuNAr7Z6jxOJf8QB05V8jqdKp4cw1Ee9RPP3EMb/49X++H5fPlZAr5K6f
DmI3f2+2jVatW3E6R17A61A6wfhxgV4ATG2V3OtXt4Jw5TqjoXNMT/fpBT/JHi+/2L7hP/6JQ7lPq/y5cu
MA5ctiTo3iLHn56BSR9NIx3fSkqyjZdqqJy9AkXpR94KCZ3vTwiIspUsXwv+/Lec6BDhw5l24y0oEikU6C
jP/oILjzyiKJsLR2+2G9Jub7fOGb/57dcZZ19UczplNtjK1fC4ZEjFUFY8alVv+KD5frNgfvlvhvpi6JOp
2CHf/klbHvzTdkYHx4xAio6PM9G4oOFS5V/AEI2wwUIVmxMt6aTUpkopi35N0SbEqypABEz1lpdr+eVike
6JczqR45AypYtlqtELdOcMzS6baUeseVw2llSUpKoPllupKrTKbjGjRtzDqNOWzt1Km+WNLtU4aFDXFt7S
YZW+6+xK3NWwJd2t+txo6o/746ASpFHzvI0zVZ/RvuJVz3SbTmCrvf29oY2bdpw59M0eaBcZYnfUrlEa1K
upp3uLLEPl/FLhrCsOMjZrphqr2unB5VE83ZGbu6Dt0R5N2asom6dvu+gsKdT3/jzM8ZcJ15d3Tr946vCk
hCchXjxFBlrqVunl4hwxJoN60S0Yq+JLp3+P5H3y3/O2MaeB0VorEunr5fpfrkIvppsojunn43mf8RuzSN
zls+ztlpX63Tn9DdCnbtPvq/ojK4cbM0Y3Tk91+T8ffIDh/R9sUZXTl8g0f3xOVeWWQsQ3azTldPJvBxJH
FMMYk74JOlaESG6cXpysrRzx39c/psiDlCjE904fdaOLyTlt65I2adOJFXegTDdOD3JJH0CIa28c8WBDwV
v1oXTK05vFkzBRoPZId/a2ML2al04veL0ZqlckiPB6Z9Uukgph3mny30ffOFSbSU1lsL5zDt99upPpeBgU
8YKOGRzG6sbmHf6ORvz2aV0iPlpVyllqimLaac7mt4sFdhZfz/tKpU8teUw7fT5mTsU4ZdoSlekH6U6Ydr
pSkGi/ehpmjSzTuc7vVmqHUNP06SZdbqQ6c1SOd7W069SyVdKDpNOFzq9WSqYb67WxzRpJp0udHqzVE4/D
XFSiVJVDpNOV/Nu99qN6r8V2dk9hjmn/+D3q7M2O9X+p/StTrXXQmPmnO5ffEwL3JjWgSmnn40+qwnYnyz
/RhN6iFWCKae/EfqDWDslbben6LSk8pQWxpTTpZjeLBXgg8GHpBKluBxmnC7V9GapCH94sXwyQqnkKiGHG
adLNb1ZKqglJjVPHJ2zggmnazXt18/Lf3eOvkqtmXC6VtN+rSnS/kt6rO1XTDg9WYbpzdZgiFlH032zVjT
vdK3fx3495J83NLDifM07Xev3se9BHiu+LtNT006Xe3pzGQUnFxYxNk1a005/nZE0X8sZmyataadHM5Tm6
/bt207+XijXXLNO/8ufrTzsvlv/q5zXnOxJs07/JVM7L/LjwzjBlManmibqaNLpJSVsXuJcuoKNtCWadPp
/ln+tiYgQqsTSwgNCm6hSX5NOP1DsXC44VUj+3en9+85nt5Jbf805fV+QsOzNcgMSKv/NP7Q/TVpzTv84R
lj2ZqFOkbt+JMTK3YXT8l2cliCxgAVd3pdYIoqrSEDTL+6pqCx+l4aA5n7epTELpdgjgE63R0en29DpOnW
sPbPQ6fbo6HSbaeHChSW+vr46NQ/NQgJIgBLAX3fcD5CAAQhgoBvAyWgiEsBAx30ACRiAAAa6AZyMJiIBD
HTcB5CAAQhgoBvAyWgiEsBAx30ACRiAAAa6AZyMJiIBDHTcB5CAAQhgoBvAyWgiEsBAx30ACRiAgKYCPSo
qCu7evWsA7GgiElCWgGyBvnrVH/Boxw7Qv28f+PSTjzmrOnV4BDq0f7js77tvvy2zdv/+ffDSpIkw7+u5E
BgQwNV5btwz3OfjAweWtSkuLoaI8HDu+5DHB0HHR9pzMo4dCy2rM+WVl7l1tK8ujz3KrS/rCBeQgAEJyBb
o337zDUyd9io8P348bN60iUN7Jvoc9xl9/gLMfv0N2LZ1SxnyoUOHQbuHH+a+P/nUU9C0WTPo0LET0LopK
be5T7qRBvnBg0FcvckvvwLmhGOvTZ8Ojw8eDD4+deHHn36GFcuXc3XWrlsPG/7un1uB/yEBAxKQ5THVwoI
CcHF15XAWFRVB5cqVgaZwdnd3B5PJxAVnYWEhuP5dx5I7HbErVSr9/SkgcirWMcuOj4uDRo0bQ2ZmJtSsW
ZMb2U9GRoGbmxs3gp84Fcn1d+bMaejU6VHLLnAZCRiOgCyBbjiKaDAS0DgB2Q7dNW43qocEDEUAA91Q7kZ
jjUoAA92onke7DUUAA91Q7kZjjUoAA92onke7DUUAA91Q7kZjjUoAA92onke7DUUAA91Q7kZjjUoAA92on
ke7DUUAA91Q7kZjjUpAcy9U1aIj0tPTISwoCJJv34ZmCQnQ9OJFaEI+TWRevr1yl8zBv9G0KcQ98ghkVqs
G3Xr3ho4dO9prgtuQgCwEMNAtsN68eRN2btsGfUhQt7h2rWyLJ1kaXPaN/0L1jAzoSP/OnCl1tHQp5Fs0L
yYP+xwYPRq8unWDgcOHW2zBRSQgLQFDP9SSmpoKB1avhjF//SUtVSek3WjRAqLHjoVxkyY5IcXYTW+TIy/
6+HK9evWMDcLCesMF+uVLlyDzp5+g08mTFhi0uVhMHuld/+qrMGXmTG0qqFGtto36F5hKAEbv/lWjGiqvl
iEuxtFf97/mz4f87t2h2UsvMRHkdFeoRPSeuGwZp3fSyJFw5vhx5fcQxnr0+30ReJ9IhJonE4EuYykloOt
Az87OhrDXX4f7PXrAc6tWMe3z2uQ0o+3s2VzQr125kmlb5FL+SNAhaP35vjLxdPnIwUNl3428oMtAp1ltj
r79NrgMGgSdIyJ0599xv//OBfyWtWt1Z5tYg2g2osIpD47ghS8vBLrN6EV3gf7HggVQ2KsXdDt6VPe+feL
HHyGDJM6kdwuMXra98w1Uzn0woCvnFcK2t+cZHQ/oJtDpFfQ7Q4bAhBUrDOVUj5wc8H76aVixcKGh7LY0d
sXipVB3fZTlqnLLdf1Pw3JSx8hFF4EesHEjVCMXq7wMnBN+Esl6e4pcoTdauXz5MrT6eLdDsx8idWhdoxb
mA33FokUwzCI/vFEdSe1+hEzMyRwwAO7du2cYDNde/I63rULq8hbKSEWmA33F4sUwyc+PEdTKqOlO0mrnj
RkDNJ223suf3/0GnrHpvM2kdWkbIxZmJ8xsX7cORvzvf0b0GS+bbzRvDm02bOBVV+uV8vLy4Py5cxB9Mgp
c7twDrwOXocYZ5y5AZnZqAFlDWkNhnWrQoetj0J48j0DfO6DXwuRc97S0NAxyB3tks9hYWE1uw00m8wi0W
rKysuD82WjuzyMpC7yCLoPXxRSb6ja3uUX4hhpnksmPRTLX8B7sBFs3YbPa+UDW460hu6EXPEJeMdae/FW
vXl14hyq3YDLQo77/HvqqDI6F7seTiTXR9L11HTpoVt2M2UugVaJ2X6xJf3jMPz6ZDY+BKYTNabXMnaNHR
0dD3/37Nbvjak2xlD//1JpKZfp4eXnBiKgVcPXLEWXrtLpAdRxxegVQnVkszAX6JQxyQftZnwMH4NatW4L
aKF15mu9r0DV2FeQ0ral01w77yyU6Ud2ojiwX5gK9z27H90xZdogcuh/euVMOsZLKrFq1Kow86QcxX42UV
K4zwq4SXUYQnahurBemAj0lJQVqkRlwWIQRqETeOMtKmTpzBnSP+xOyW9RSTeXs5t6cDtOILnopzAW6XsA
raUdtxubCe3h4wKiIpXBt7iglMXF9xZA+Rx1fBlQHPRWmAp2+Wx2LcQhMeW069Ij/C+61ri270dmt63B9T
SV96rEwFeh16tTRow9ktym1fn3Z+5CrAzqJ5YnQJRDz2TC5uoBrnw6DUaGLdT1hhqlA9/HxgbTa8v+6y7Z
HqSS4uEYNlXqWrttaOy9IJ6yCpFqB5yus0d9XpgKd4g8lT6lhEUZg4BNPCGugsdo0U5D38QTZtKJpp+gsP
T0X5gK97dChevaH5LaFkmf0Wc+GGnb4iORcKgoMDw6puEpX35kLdDqd8ygGO++d0Ickw2S9xF2Lld2EBAX
6kN0IOx0wF+jUlsf+/W87JuEmMwH/KVM0Pc/drKejz9oB8p9D19ohfx+O7JRzO5OBXqtWLdjz3ntycmFeN
n1MVctPrvEFTM+d6Tk035I6qCU0Dv8emkT8AGmPt+LbDGpGJsFdHWcocuFNQmMVR0+YACvIjK9JJO85lvI
EsshjlC11kiE27FAwuJU374FveeQR0srzJsKQJ8o/HNPCvzT7TNCuPVD40Tpwd/CUXPjhEBj2NNsXLh+A8
/cKJkd0szFTyRtM1hgwT5rZfmufeWRGlzt5f5yLyz+/4SOXzoDFy5ZYq675dUmx8VZ1LHatBDfmPQUDUjb
B8NMrHwhyy0aDR42A4eQpOVqXtqFtrZXE6zesrdbFOusWM2TaVF9f2PfBBwxpLJ+q5zp1ghrBwVCNvLnVX
D5c9hWkwT3wKzkIYRHh5tXMfHpvjy6n6xUycab/7Y0wKGkDTJ4+tdw2Pl9oG9qWBj2VZVlq6/g8nflAp45

66rnn4B55qo0eshq1rCVHNl0q5M9btHQxBJWcK0Py7un5kEPSQ7NSMsmpWc2oZEiY1Rs6X1vJBeerb/iCV
FOhqSwa8F2u/0H66AM1opKA9qnHwmzOOFvOoC9wMFJu9xxyqJ7v7w/1K0xz3XdwP8y5+mDizA7QGJbP+N4
WPlyvUwK6GNEtffMKeT+ZS1gYnOir/2RTO999F7zJoXrFIKej0idXl1tiKVuOhgT4zc+4L3soA2GwBd0FO
vVf5cqVoe/PP0PhoUMQ1bOn7ly6mSR8dCNvVn1m4kSrtvmu/xSKgbw32EZZVRwMf/mvtbEVV+uRgC4D3ew
omhmk52+/QRXyosWNL79sXs3kZyp5mCeCvKiCBvhEMhHGVvnB71eIMd22tbls/S+Z2yEmJqbsOy7om4CuA
93sOnrx5sU33+SC5AZJlnima1fzJk1/FhO9106fzundkFxs7D94sF19/VatAP/iY3brWG6cHTQP6Lvjsei
fgCEC3dKNbdq2he7kNU50ZKRX6rdNmmS5WfXlGy1aQOA773D6eZAjkSlkrgCfcpZkx12cv5dP1bI66ZANc
1bMK/uOC/oloLur7s64ir5+eCeZbNLn4EFoocBhbTG5lnCAvD7Jq0sXGDhihGjV6euXhq6YDjmQL0rGNBg
EvjP4/aCI6gAbqU4AA52HC9LT0yEsKAiSb9+GZomJ0PTiRWgcFweVHBz2ZpKED3FkzvmN9u3hLrle0K1PH
+jYsSOPHoVVeWPpJ+RNI1eFNapQ++uHpsPQx4dUWItf9ULgn3mSerFIBju8vb1h1LPPCpZcl7Sgf90Et+T
fYOHSRU4HOe1tztVl0K1zV6hZU3u51fnTwJq2CBjuHN0WCBbXbwnYCivgsCSq00tyvv6fSSILhWiPAAa69
nzCS6Pk5GSYl7yeV12+la6R23I/Lp/PtzrWY4gABjpDzrJUddaOLyy/Sra8rigU6G06LPoigIHOoD+/9Ps
BkkwZsmlOb9OdI+8jx6IfAhjojPlysd8S2FF8UnatZx/9DuhtOyz6IICBzpAfwyLCwK/4oCIa55ruw3srv
1SkL+xEfgIY6PIzlqSH3NxceO/0b5LI4ivkWMlloLfvsLBPAAOdER++vvpTKIAixbWlt++2Be5QvF/sUFo
CGOjS8pRF2vxlCyDalCCLbD5C5yatAXo7Dwu7BDDQNe67P/3XwOqSI6prOWsHnq+r7gQnFMBAdwKe3E2vX
bsGv2Zq47A5yZQOc5f/KLfJKF8mAhjoMoGVQuys/V9LIUYyGduKjsMSv6WSyUNByhHAQFeOtaCePlw2F9J
N2YLaKFF5WXEQRByPUKIr7ENCAhjoEsKUSlRpmuby+cylki2FnHeifoW8vDwpRKEMhQhgoCsEmm83NE3zc
jjEt7oq9ehtvtdXfapK39ipOAIY6OK4ydIqIyMDPraSi12WzpwUetYUD7/74WQaJzEq1hwDXTHUjjuiz4P
T58JZKX8UH4Y1G9axoq6h9cRA14j7aZpm+jw4a+XnjG1w/fp11tQ2nL4Y6Bpwud+q5YLSNGtA5XIqaO02Y
Dnl8AtHAANd5R2hNE3zPpW1cK57+rbWj5d/45wQbC0rAQx0WfHaF06f934jVB8vPNxbdJrZd7Db95I+tmK
gq+jHt1d8DvS5b70U+g72oMPKPC+vF2ZK2YGBrhTpCv0s4NI06+/dZ/+5vFS37xiv4EKmvmKgq+CuLQHbY
KVEaZpVUN9ul/QtrrPWf2a3Dm5UngAGusLMS9M06/ve81XTLfhpubLZcBR2I3PdYaAr7LJZAcZ4rntt0VF
Y8edKhelid7YIYKDbIiPDei5NM6TLIFmbIhfm7oHz589rUzmDaYWBrpDDF5PnuJVI06yQOby7eZ2kjS4qU
j7XHW8FDVIRA10BRx8Lp2magxToSXtdZJNXOb+34gvtKWYwjTDQZXY4l6b5jLEvTIVi2miZ9zLH4jHQHTN
yqsbrf34GhSqkaXZKaRka07TRO3YFyiAZRfIhgIHOh5LIOlyaZogX2Vp/zb5M+BNu3bqlP8MYsAgDXSYna
SVNs0zmiRY7a9sXottiQ/EEMNDFs7PZMiYmRjNpmm0qqdKGBFMazFv+k0q9G7dbDHSJfV9SUgKzg+ZJLFV
f4rYURcCS5cv0ZZTGrcFAl9hBH/l9TabEaC9Ns8RmOi1uWdEBOH7ihNNyUAA/Ahjo/DjxqqX1NM28jFCw0
tunfob8/HwFezRuVxjoEvl+b5D20zRLZKpkYgpMRfDGH/ikm2RA7QjCQLcDh++m9PR0+CTGj291rGdB4LT
pBvy+DNNGWyCRZREDXQKsszZ8zlSaZglMllTEHyWHYd0mf0llorDyBDDQy/MQ/O2H5WymaRZsqMwNfkzbA
jdu3JC5F+OKx0B3wvd+q1aAf9ExJyRgU0sCs/bMtfyKyxISwEAXCfNs9FlYnL9XZGtsZo3AHVMWfOKHaaO
tsXF2HQa6CIKlaZp/ENESmzgisKcY00Y7YiRmOwa6CGrvrPyvrtI0i0AgaxOaNvrwkWBZ+zCacAx0gR5fS
NI0h5dcFdgKqwsl8MHFRZCVlSW0Gda3QQAD3QYYa6tpmmb6XDUW+QnQtNG+a/Ed7FKRxkDnSTIpKQnmJes
7TTNPFIpVu2K6CT8v/12x/vTcEQY6T+8aJU0zTxyKVVtTFAIr16xSrD+9doSBzsOzXy3/HySbMnjUxCpyE
FiQvQsuXbokh2jDyMRAd+DqxcuWwPYifJzSASbZN886PA+Ki4tl70evHbjo1TCp7Jo5/TWYCa9JJQ7lIAF
VCOCIrgp27BQJKEsAA11Z3tgbElCFAAa6KtixUySgLAEMdGV5Y29IQBUCGOiqYMdOkYCyBEwLFy4sUbZL7
A0JIAGlCZhIHnIMdKWpY39IQEECixYtAjx0VxA4doUE1CKAga4WeewXCShIAANdQdjYFRJQiwAGulrksV8
koCABDHQFYWNXSEAtAhjoapHHfpGAggQw0BWEjV0hAbUIYKCrRR77RQIKEsBAVxA2doUE1CKAga4WeewXC
ShIAANdQdjYFRJQiwAGulrksV8koCABDHQFYWNXSEAtAhjoapHHfpGAggQw0BWEjV0hAbUIYKCrRR77RQI
KEsBAVxA2doUE1CKAga4WeewXCShIAANdQdjYFRJQiwAGulrksV8koCABDHQFYWNXSEAtAhjoapHHfpGAg
gQw0BWEjV0hAbUIaCrQV6/6Qy0O2C8S0DUB2QL9nbfegi6PPQo9u3fjBZC+5H5nYCCvuvYqvTptKixc8Lu
9KrgNCRiOgItcFu/btxeOHA2F/n37cF0UFBRA50c7QbuHH4aLFy5w66LPl37SL506PMKto/91aP9w2TJdc
HFxgcLCQqhfvz7sDzrIbafLN2/eBP+NG6F9+0eg4yPtwcPTE3Kys8Fv+Qr4/bffygL+rbffgRmvvVZOJn5
BAkYiIMuIHhsbyzF89+23uc/IyEhwdXWF6TNe44LcMsDNsC3XmZfNn088+SQsIK+VoYFNy/zffodf5v/GL
R8JDuY+6ZulQsgPCy20fzqq9+jRg/sR+OXnn7j1+B8SMCoBWUb02b4zoVq1atC3X1+IOh0Fn348B3YE7gS
TyQReXl6CWNM2nTp1ggYNGpa1e/ON16F79x5l36PPngXPqlVh8KBB8My4cdC8eXNu2+SXX4GGDRty28oq4
wISMCAByQP9sU4ducNsynLipBfh559+guvXr8OY0U9DQnw85Ofnw66dpefigwb0h0PBRzjsjw8cyH0++cQ
ouJeVxS1HREQAHamXLF4MtTZs4NZ9M+9r7jPrXmkdv2XLuH7oIXsOZMOO7dth5MhR0JH8ONAfBA8PD6hSp
QocPRbGtcP/kIARCUj+NlUamHQUvn//Phdg9CJbpUqlZwj0PJtuq1y5Mse6kJy3u5BDelrMy7R+cVFR2fo
ismyWaW5H69MfjxYtWnCnAovJYX3E8Qg4GnoM5nz0EQQGBkDUmbOQnp7O6VCVjPZYkIBRCdC3qUoe6GrAp
D8gzz/7LFy5cpkbwYNDjnKfauiCfSIBrRGggS75obsaRtKr8lu2bVOja+wTCTBBoPSYmglVUUkkgATEEsB
AF0sO2yEBhghgoDPkLFQVCYglgIEulhy2QwIMEcBAZ8hZqCoSEEsAA10sOWyHBBgigIHOkLNQVSQglgAGu
lhy2A4JMEQAA50hZ6GqSEAsAQx0seSwHRJgiAAGOkPOQlWRgFgCGOhiyWE7JMAQAQx0hpyFqiIBsQQw0MW
Sw3ZIgCECGOgMOQtVRQJiCWCgiyWH7ZAAQwQw0BlyFqqKBMQSwEAXSw7bIQGGCGCgM+QsVBUJiCWAgS6WH
LZDAgwR0EVySDV4Z5M88pZ/9JVTNK20J3ktFP2sUaNGWZprNfTDPpGAJQEMdEsaNpbPkjfBnAgNhRokuJu
ePw/NbtyA6hkZUIPUp3/Wyn2Syz6+cWOIa9cObpDPBnXrQq/Bg8Hb29tadVyHBGQloIu87lITOiP19vXro
TYJ6K6HD0M18l2qcqVtWzjZpw906tkTOnftKpVYlIMEbBLQzQscbFoocMPhvXsh68QJGEJe61SJvCFG7nK
9ZUsIJaP8E2PGcG+Klbs/lG9MAhjof/t9y5o10DYgAFpduaLanrB58mToT14Q2Zgc5mNBAlISoIFu6Kvum
//6Cy6PHw9PkBdBqhnk1KnjVq8Gn2eeAf+ff4bU1FQp/Ww4WXTHxlKegCEDnb6g8cCcOfAkCapmZFlLZQz
58XEdOxZWk7fEYhFOgL6k86FfjgL9xPIPAcMF+kryCub6L7wA/cj5uFaLW14ejCd6Hvf1hcuXLmlVTU3qt

```
Xd7IFS5kwN7tgdoUj+1lDJMoN+9exf2fvwxTCQjZSXyamcWSqeTJ6HRlCngv2oVC+pqQseMsPOcHllhFzW
hj1aUMESghwQFwT1yLj5wzx6tcOethwt5JfSY+fPhr99+494Tz7uhASveu3cPGqw4zlleb0UEZGVlGZCCd
ZN1H+jb/f2h+wcfQG3GL3A998cfsPfTT4HOwMNincC2dRvAVFx6tEY/t/tvsl7RgGt1HehrV66EEd9/rxu
3Dtq9G06+/TY39VY3RkloiEdQ+esZnvvx8N2MV7eBvnzJEhj3++9mO3Xz2TkiAq6+/jrkkQt2WP4hEBcXB
3X2lZ8HUXv/FbhBZjdiAX3eR9+ydi28uHSpbv3b7tw5OPnhh1CkwOw9ViDu27nbqqr7dllfb7Wyjlfqbk
/cuAAjCQTYPReuh09CgHffad3M3nbV29tpNW6DdZEWV1vtJW6CvSbN29Ch//+l5nbZ87ubCM3b4Y/FixwV
gzz7aNOnYLq529btcPrwm2IJLcpjV50Fegx334Lnrm5hvLphBUrYPfWrYayuaKxUSHhFVeV+346JKLcdyN
+0U2gryTzm7uFhBjRh9Dt118NOz++hEx+arQg1K7fGy8MNfwcBF0EejgJ8Il+fnadreeNXmRiSOSPP+rZR
Ju27QvYCW6pOTa30w1VyPZ9Abvs1tH7Rl0Euis5fDV66U/m7gds3Gg4DGnhF3jZnBZ2jlc9vVZiPtD/JCP
5I2fO6NU/guzqQGbPGemWW05ODjTws39+bgbYYHmEoScaMR3odG7z02TnxlJKoAG567BKx/MHKvp5+/qNY
Crk9zgqrbfDwFNimQ50/z//BHeDXWWvuLNX/D6WZMvJz8+vuFqX310PCpviWkVgfT1BYzbQC8lTXU+TB1a
wlCdAby+uIXP89V4SEhLAZ1f5ue2ObK5D6tN2RizMBvoacgGuOj6GaHWfHUUm0ug9w8qugECrtjtaGUByA
xqxMJvumeZ601oaKC3tQHveew9GT5igJZVE6xIfHw/Rx09B4q2b4H0iAaqH3gC3O+JTcOfXqQp3+zSD9G6
NoVG9+tChexdo0qSJaP203pDm0GPyBQ6nIyOhncZyvWnN2V5RZI43g4FOnzY7E34CbqWkgHd4HNQMjwfXt
ByoSgC3kQgy/ZHw2X6e+6Mir8MquFzLEzJ6NoGMXk2hbh0f6NijKzRv3lyiHtUXw2SgnyFvTWmnPjtNa9C
HPNyTlpYGtWrV0qye165dg7PhJyE1LRVqHI0lQR0Hrnfz7b4BRy5j6I8JPec3n/fHwRqI8XKDzJ5NIbNvc
6hVpzYX/C1JLn4WC5OH7ilDh0L1zEwWeSuqs//MmTB5+nRF++TbWdDefeDy4iK+1TVTr2Tt6zBw6GDN6MN
HESbzukdHR2OQ8/EuqdMoJoZnTeWrDR4+DK5+OVL5jp3okerLWpCbzWXuqvtxctiOhR+BzhpnNc13BiTO6
MnPGJVrJU7vCVRfVgtzgd6cnNdh4UfAk0wRpW+C1XJ5+j+zIbeJrXfSakPz3MY14Ok5s7WhjEgtmAv0Lho
fpUT6QbZm9HXPWi7VqlUDr5+nallF8PplKlA9WS5MBfqtW7dwyqvAva06A1OE+wzoT87XRwi0TJnqVC+qH
+uFqUC/gYftgve3JudL31wiuKHCDab5vgYJvr0V7tV+d1QfqpceClOBHntR2EMMenCQszY0IWmQWSljP5o
NOS28NaFuTnNvoPropTAV6PkMHIZqbceoRd5Qk50tfrqokvZ4enpCrZ9fBTCZlOz2wb5I/7V+eRWoPnopT
AW6x/37euGuqB00QQMrpUef3hCj8vl6DLlfTvXQU2Er0BnaYbW0k7AU6JTb1JkzIOHNfqogpP1OnanN2YT
OAGEq0D3Jq4+xCCfAyqG7pWXj/m8WZLeqbblK9uXslrWA9qvHwlagZ2To0Qey28TaiE6BuLu7g88v06FEo
fN12o/PrzO4fmV3iAodMBXoOd7auCKrgp+c6pLVi0rdevaA2LmjnLKdb+PrX40E2p9eC1uB7uWlVz/Ialf
VqvRpbjbLKzNehfi35Z2wEvdWf5jymv7Oyy09zlSg5+rodoelE+ReZnVEN3MZ9+5MKHaTJ3UClfvMO+w+r
GJm5OiTrUCvUsWRPbjdCgHWA50mdKyUX2jFMudXUbmJiYnOC9K4BKYC3c3DQ+M4tadeWu3awPKhOyUaf/W
6rGDlli+r8jyFMxXozdthAimefi2rFt+sWdkyqwuJJDmknCXRACmgmQr0Zozm65JzJ3UkO/7hhx1V0fz2w
vsFsupYlC+vfFmV5ymcqUCvV68e5OHhO0/X1lbL0sEFzGo35J0/Ibd8QQ6TqTJTgU4ZnOrTRyYU+hTbtTf
7c7bdopNkdY7bWbwYJytgMcJj8fCdN7YcMpp37NiRd32tVvS8liaralWvp8sqXwvCmRvRu+OIznu/idQBK
/rG3CppubxtFlPRNT0XsnT+ei/mAr1Dhw5wt2ZNMf40XJvEVq2Yt1mpe9xJSfKeHqjtCOYCnQLbPXas2ty
Y6P/JceOY0NOekvFXlMn6G3dZuznw7fHhu43JQO+kg0NSvg4SWy90yBBNv46Jr11iRtqkse35ii+rl5SMI
3oZDK0sPNq5M9xo0UIr6mhSj6zHHtOkXkKVKhZ4j/sqeQptwtIvgX4KSkk10xRbofbKVZ/JEZ3CCBs2TC4
mzMtNJy9WfIq8VloPhe8V97sP+0C1nR/BNJKdhhb6WS3wQ7jbvi4vDHz74SVMg5WYDfRJU6fCXXxs1eout
evZZ6FSJWZdW84m93PJ5b5b+3L9P0NgZNBv0KV7t3Kb6feRB+YD3e6oyH2v3lH/cm9ndm9wcXGBHToZtaR
0cg6ZOTjplVfKRNLbUwuWsffWUrMB9u5xF5LXGt/5dTy88s5soPuDtULX0+13fhkPBaS+rWKvH1ttWFrPb
KBTyONfegmnxFbY27ZOmgRubv/s0Gs2rIONJeFQXFxcoab2v94lOQJdM6zfQ781/lHoEPYzjJv4Ai9Dxk1
6ATqS+rdesH7twjUzDzJ1/CpupgOdvg9rx8sv83K0ESol168PL88oPUc123uk4DzcgzzYsHWTeRUzn7auu
NPXJD3/+6dQty6/82+zwbT+8799YvP1T7b6M7dn+ZPpQKfgX5o+Hc516sSyDyTTPZocsleuXLlM3qnIU3D
JVHqOezz9Qtl6VhbiK9zbzmrrAx47/s/p1yTR1yx5BnwAWe18yqGo2F+5jYx/YT7QKf8CcmHO6OXI8OHw1
HPPlcMQfOpY2ffg4gtw48aNsu8sLCQml/5IUV2vfzQYhh+cD9179ZREdZoIcngQuVD34eNl8pJu3ixb1tu
CLgK9Z79+sGbaNL35hrc9WdWrQ+d33y1X/z55q83m4vBy63bu3VXuu+a/5BUAveCW8vPz8Mq7r4Orq6ukK
lN5r7z3BnehrtCrCpgE3rOXVBmZheki0CmjqbNmwfG+fWXGpU3xJ958E2qTlFGW5a8Na8mZefmECjtKTll
W0fyy6908eOTYT/DsixNk1ZVeqOsQ9gu4kAtyei26CXTqoIc+/BDo7SUjlXXktGWklbn/YTkPvnn2jikLd
u5hZ1R/8ZN3gCYbUaLQC3W0P70WXQV6fXLVOfqzz6BYobd7qL1T7CYPrbwy+8FX+16+fBkiIdaqeuGJZ62
ux5X6JqCrQKeu6k8e5tj9jn5/mc274wlymvLU//2f+Wu5z32HD5T7bvllV1Ek3L5923IVLhuAgO4Cnfrsm
YkTYcWrr+rWfedI1phHv/qq3K00s7F0YszmkgjzV6ufW7ZvtboeV+qXgC4DnbrL19cXNr/+uu48F9WzJ7S
dPx/oZCFrZf3WjZBlsn9RaV/JGWtNcZ2OCeg20KnPJk6ZArvff1837js8ciR0+eknuy9kOJ56waG9cZAKw
SHBDuthBf0Q0HWgUzeNeeEFiPj2W0itcPuJNRduJFN9h33xhd17yfHkRQch8ODVdmu2hl5m61abNRtwHX8
Cug90iqL/4MFQzd8fDo8YwZ+MRmoWkqevtpH75C+SP5ODuwmBu3fy1nprUYTuEyLyhmGAioYIdOrH6mT22
HByAWstmRvPyu23M127QuLKlTCe54M7AcB/lC6GEli/yd8AuziaSAkYJtDN7p4ycybcXL8eQsjccK2WfHd
38Cd6dl+0CNq0bctLzYA9O+E23OVV11zpcME58yJ+6pyA4QKd+rMFyTc3ZO5cCHz7bc3lnttGnicv2LoVJ
pMjDyElIkH4RJhLkAynIiOFdIN1GSVgyEA3+2rciy9CG3LuvpNMsLn60EPm1ap8bp48GVK2bIHxRJeK89Y
dKXTnzh3YXRzlqJrV7Ucij1ldjyv1RcBUQoq+TBJvzeE9eyDr1CkYsm0bVCoqEi+IZ8vr5PVSoeRC4RNjx
gCdviu2LFq6GJbDIVHN3cAVDryyDKpUqSKqPTbSPoFF5BQQA92Kn7Kzs2E7OY+vHRsLXY8cgWok75pU5Qo
55z5B8tI/Sia+dCYX26Qozy15A+JMqaJFzfYcBVNexEw9ogFqvCENdOsZ9TSuuNzqVa1aFSZaPN9+9uxZO
BEaCtXJD0CzCxeg6fXrUMNBfrESkoU1vkkTiGvXDm40agQNyNNRvcjo3cHbGzpIaMBhMvHFmSCnqoTlXIA
pEuqEorRHAAOdh0/oG0krvpWUjvqWfwUFBdyMNU/yBlP6Q1GjRg1oTYK9NQ/5zlQ5don/LTVb/ZyCWLh06
RK05XmF35YcXK9dAhjoIn1Dg5n+qVloKudtxcclUeHAkYMY6JKQ1KYQQ19116ZL+GtFUzkXgTRpnDcxmhK
```

aPy1j18RAZ9j/NJWzVCWLJJ7atH2LVOJQjsYIYKBrzCF81TlJbgOaUznzbeOoXkQqzpRzxIjV7RjojHruS
GSo5JofJimh4+LiJJeLAtUngIGuvg8Ea1Caytl+FhnBQv9uwFLySLE2GrEdBjqDXv+LXISrmMpZKjNYSwk
tld16l4OBzqCHw7IdZ5ERa1aK6S7s2rtbbHNsp1ECGOgadYwttbhUzqZYW5slWR8WjznlJAGpISEY6BpyB
h9V7KVy5tOeT51dxZFAn4jDoh8CGOgM+ZJPKmepzNm0dbNUolCOBghgoGvACXxV8OeRypmvLEf19hULT2T
hSCZuV48ABrp67AX3HJEm3Uw4R53Hme7AkaNHHFXD7YwQwEBnxFFcKueSS4pqG3rppKL9YWfyEcBAl4+tp
JIDdgdKKo+PsK1Fx4E+IYeFfQIY6Iz4MJC8H1XpQp+MW7dxvdLdYn8yEMBAlwGq1CLFpHKWSofD9/FBF6l
YqikHA11N+jz7DldxAgt9Qg5TQvN0lIarYaBr2DlUNTpxZU/JaVW1DMGU0Kryl6JzDHQpKMooY+OWTTJK5
yd6Y1EY0CfmsLBLAANd477bXxKtuob0Sbm1eFFOdT84owAGujP0ZG5bmspZG3PO5XxiTmaMKJ4QwEDX8G5
wTEMTVk7CdaBPzmFhkwAGukb9VprK+YSmtNsfHKQpfVAZ/gQw0PmzUrTmWglTOUul+OaSCKBP0GFhjwAGu
kZ9FixhKmepTLwLuZgSWiqYCsvBQFcYOJ/u5EjlzKdfPnUiUpV7go6PPliHHwEMdH6cFK0VLEMqZ6kMOFx
8HuiTdFjYIoCBrjF/laZyDteYVuXVCdy9s/wK/KZ5AhjoGnPRmo3rIB8KNaZVeXUCSpx/g2t5ifhNbgIY6
HITFij/mIypnAWqYrP6bZoSet8em9txg/YIYKBryCf0HeWR5F3lLJQIFZ+oY4GP1nTEQNeQR1iakBJYdAp
SU1M1RA9VsUcAA90eHQW3camc4biCPTrflRaerHPeCjhxJYYAAD0CSURBVGNIwEDXiJ/9t26CLDIhhaWyH
1NCM+MuDHSNuErJVM5SmXyDpIQOCQ2RShzKkZEABrqMcPmKpu8kDym5yLe6puqFXsSU0JpyiA1lMNBtgFF
ydeCeXUp2J2lfW4oiIDs7W1KZKEx6Ahjo0jMVLJHlCSg0JTR90g6LtglgoKvsH5rKmb6TnOUSfB8fdNG6/
zDQVfZQRAL77yK/aEqCqNPqZqpV2Y2a7x4DXUUXpaSkwO5ifQRI8MmjKpLErh0RwEB3REjG7Zu2bpFRurK
iNxSGQUFBgbKdYm+8CWCg80YlfcV9Jewftpup5JswJbSZhRY/MdBV8gpN5Rxv0tdc8bB7F1Siid06IoCB7
oiQTNu1lMpZKhNPwDW4cuWKVOJQjoQEMNAlhMlXlBZTOfPV3VG9fYcOOKqC21UggIGuAvS1G9YDnWiix0J
TQpeUlOjRNKZtwkBXwX3BBfp95/hdUy5s3rFVBarYpT0CGOj26MiwTcupnKUyN+y2+i+GlMoWvcjBQFfYk
0c0nMpZKhSHS85DQkKCVOJQjgQEMNAlgMhXBE3lvKk4gm91pusF7ApkWn+9KY+BrqBHS1M5G2P2WGBJpIJ
ksStHBDDQHRGScDsLqZylMveWKRP27N8rlTiU4yQBDHQnAfJtzlIqZ742OaoXHqefKb6ObNX6dgx0hTx04
MhBhXrSTjc0JXRaWpp2FDKwJhjoCjifpnLeRCaSGK2UQAls2LzRaGZr0l4MdAXcsmEbe6mcpcKCKaGlIum
cHAx05/jxam3kd4rTlNBHj4Xy4oSV5COAgS4fW04yTeV8hNFUzlKhCb3A1htopLJbS3Iw0GX2xk6GUzlLh
QZTQktFUrwcDHTx7Hi13IHvEidvey+GdRvX8+KFleQhgIEuD1dOaqAOUjlLhecIpoSWCqUoORjoorDxaxS
ecJZfRQPUOg+JcPqMPjLesuguDHSZvFaayjlKJulsig0+jimh1fIcBrpM5Ddt3SyTZHbFbijGlNBqeQ8DX
Sby+0rwsL0i2jwogHWb/Cuuxu8KEMBAlwGyHlM5S4UpLAvf0yYVSyFyMNCF0OJZN+zyKZ41jVftOEkJffX
qVeMZrrLFGOgSO4Cmct5ahDPB7GHde3C/vc24TQYCGOgSQ9VzKmepUG2B45gSWiqYPOVgoPMExbfa4QLMg
OqIVSbkYEpoR5Ak3o6BLiFQmsr5summhBL1KyoiRb+57bXoNQx0Cb0SEnVMQmn6FnWw+BwkJibq20gNWYe
BLpEzaCrnjUXhEkkzhhhMCa2cnzHQJWK9ZsM6yCcTQrDwJxBYjLch+dNyriYGunP8ylqHZuO7wctg8Fy4S
VJC7z2wj2dtrOYMAQx0Z+j93Zamco4yxUogyXgiwm/gE21KeB0DXQLK+w2YylkCbJyIAEwJLRVKu3Iw0O3
icbyRpnLeXIIX4RyTsl4DU0Jb5yL1Wgx0J4mWpnLOc1KKsZsHFeMkI7n3AAx0JwmHp+LTWE4ihOumFAgNw
zkIznK01x4D3R4dB9toKucQg6dydoCI9+ZjF07wrosVhRPAQBfOrKwFpnIuQ+H0wsaCMMjJyXFaDgqwTgA
D3ToXXmu3l5zkVQ8rOSZQZCqGtWTSERZ5CGCgi+QauHcX3DFliWyNzawROJKPk46scZFiHQa6SIph8TjRQ
yQ6m83OmxLgzFl8p7pNQE5swEAXAY+mct5TjIEuAp3DJocjMCW0Q0giKmCgi4CGqZxFQOPZZCNJCV1YWMi
zNlbjSwADnS8pi3r7AVM5W+CQdDEX7mNKaEmJlgrDQBcINTjkCMRBqsBWWF0IgWN3cRKSEF586mKg86FkU
Sf0Mt5Ss8Ahy+JxiIGYmBhZZBtVKAa6AM9nZWWRVM4RAlpgVbEEMCW0WHLW22GgW+dide168jqhYiixug1
XSktgS0kEpoSWECkGugCYh+5j5lIBuJyqmkFSQm8J2OaUDGz8DwEM9H9Y2F06FUlTOSfbrYMbpSUQcRt/W
KUiioHOk+SRSHyMkicqyarR59STkpIkk2dkQRjoPLyPqZx5QJKpyvbAHTJJNpZYDHQe/sZUzjwgyVRlZ0m
kTJKNJRYDnYe/j+XgU1U8MMlShaaE3heEb191Fi4GugOCNJVzJMQ6qIWb5SQQHosPEDnLFwPdAcEDmMrZA
SH5N+8oOgHp6enyd6TjHjDQ7TiXpnKmT1NhUZcAnaLkv2mDukow3rsL4/rLqn6lSpXg+65vytoHCkcCShD
AQHdAuWvXrg5q4GYkoH0CeOiufR+hhkjAaQIY6E4jRAFIQPsEMNC17yPUEAk4TQAD3WmEKAAJaJ8ABrr2f
YQaIgGnCWCgO40QBSAB7RPAQNe+j1BDJOA0AQx0pxGiACSgfQIY6Nr3EWqIBJwmgIHuNEIUgAS0T8C0cOH
CEl9fX+1rihoiASSABJAAEkACVgksWrQI8MjdKhpciQSQABJAAkiALQI4oLPlL9QWCSABJIAEkIBVAjigW
8WCK5EAEkACSAAJsEUAB3S2/IXaIgEkgASQABKwSgAHdKtYcCUSQAJIAAkgAbYI4IDOlr9QWySABJAAEkA
CVgnggG4VC65EAkgACSABJMAWARzQ2fIXaosEkAASQAJIwCoBHNCtYsGVSAAJIAEkgATYIoADOlv+Qm2RA
BJAAkgACVglgAO6VSy4EgkgASSABJAAWwRwQGfLX6gtEkACSAAJIAGrBHBAt4oFVyIBJIAEkAASYIsADuh
s+Qu1RQJIAAkgASRglQAO6Fax4EokgASQABJAAmwRwAGdLX+htkgACSABJIAErBLAAd0qFlyJBJAAEkACS
IAtAjigs+Uv1BYJIAEkgASQgFUCOKBbxYIrkQASQAJIAAmwRQAHdLb8hdoiASSABJAAErBKAAd0q1hwJRJ
AAkgACSABtgjggM6Wv1BbJIAEkAASQAJWCeCAbhULrkQCSAAJIAEkwBYBHNDZ8hdqiwSQABJAAkjAKgEmB
/TU1FSYMf1VePaZsTBh/POwdcsWq8aJXZmdnQ3du3aBDu0fhm/mfS1WDLZDAkgACSABJKAYARfFepKoozt
3UmDwoEFQXFwMU6dNgxXLl8PHc/4DNWvWhEGPP17Wy08//ghJSYmQn58Pr702E65fvw6rVv0BN5OTYcDAQ
fD5f/8Lrq6uZfXNC+np6TD3qy9hIOmDlm7du3OftN95X3/NtSkqKoJPP/scfvzfD3AwKAgaNWoM33z3HdA
Dje++/QZirl6F1q3bwOdf/Bceeqg1157+t2P7dlL/AJw5cwbuZmVBk8aN4eVXpsCYsWPL6tCFQwcPwsIFv
8NVIuf+/fvQpEkTeOe99+FOym2YOOlFru6mTRvBf906uB4bC40aNoIXJk6ACRMmlpODX5AAEkACSMA4BJg
7Q3991ixuMO/arRu89/6/Ydiw4Zy3Pvv0EygpKSnzXBdyhk0H56ADB2DCC+Phj5UrYOjQoaQOwLatW2Dc2
DFldS0X3N3cYPDgIXDs2DHYvWsXHNi/n9vs4eHJrafydu3cCX169SSDbQH07NULTp+OglEjhsOMV6dBd3I
A0KZNG4iKioSxo0eTQfkK1/7kiRPw0YcfwN69e+HLuXNh3Xp/uHTpEsz5z0ewbu2aMhU++L9/wxuvz4a0t
DQIDQuHj/4zB+Li4uCdt/7FHYgkJyVBl8cehc8++QT6DxgIR0OPwaOPPgpfffFEFd1WhsKCgTBYuIAEkgAS
QgHEIMDWg/zZ/Ppw7d47zDh0g6SXxffv2ct/p2fFPP/6vzHMDyVl4x46duO9eXl6wcfMWmOk7i5zhTuLW0

QHTWvHw9IQnnnwSGjRoUG5z1apV4cmnnoI6Pj7c+tGjx8Bccsb+7Xffg8lk4ta9+a9/wTvvvgcff/pZWds
7KXe4ZTrovjJlKtd+xquvwuinniyrk5mZWbbcs2cvbjmZXEkY2L8ffD33K+57J9K+YcOGsGGDP3fWTlcuW
rgAOj/aCTZu3MDVyc3NBdoOCxJAAkgACRiPABOX3JPIWenRkBBuAKMuooP1T7/8AlWqVAE6iD3/7LMQG3s
dlvv5kbPjtuTMtT/UqFET8vPyOI/mkcvu5pKbm8Mt0kvx9kpebmnbivXMMiu7VOaaF5AzYvOVAfMFArrOX
Mx9TySXxC+cPw8dOnSAQ4eD4UhwMMzynclVy8sr1WVnYCB8+snHnF30lkBrcqZvIv+aNG0K9ICClq5du5H
/F3PLH3z0EUye/DJ3FWAOWc7JyYFG5DI+FiSABJAAEjAeAdPChQtLfH19NWt5YWEhdzYbHx9fNqjdu3cPH
h88GH7+5Vd46olR5F55ElSrVg3oQEoHtZGjRkH02bOQkJAAnuSMmw76nTo9Ck+PGQ1zv/ySuw/ODcLkzHr
J0mXcZXIzgPPnz8HLL73EyXJzd4dcIs+HnJV/+/0P4PvaDCgmo3Yl0o4O9HRAPRcdDXQgruziAlXIPfmNm
zbDxAkTICvrLpjbT5k6FcY9+xz8+7134eLFi1xXzZo3h4kTJ8H/fvie66tX795Az/D//f77kJSYCO3atSv
VnRyUXL1yhatTvXp1OHosDG7dusVdcj96NIST5UL67tChI7z1zjvlbDHbhJ9IAAkgASSgbwKLFi0CzQ/o+
nZBeet++flnWLqk9Oy7fv360KxZM0gltwauxcRw8wYGDxkCv87/rXwj/IYEkAASQAKGJ0AHdCYuuRvFU2+
9/TbQP1qyyCz4K5cvk1sHNbjL6O7kagEWJIAEkAASQAK2COCAbouMyuvpRL4uXbuqrAV2jwSQABJAAqwQY
GqWOytQUU8kgASQABJAAkoTwAFdaeLYHxJAAkgACSABGQjggC4DVBSJBJAAEkACSEBpAjigK00c+0MCSAA
JIAEkIAMBHNBlgIoikQASQAJIAAkoTQAHdKWJY39IAAkgASSABGQggAO6DFBRJBJAAkgACSABpQnggK40c
ewPCSABJIAEkIAMBHBAlwEqikQCSAAJIAEkoDQBHNCVJo79IQEkgASQABKQgQAO6DJARZFIAAkgASSABJQ
mgAO60sSxPySABJAAEkACMhDAAV0GqCgSCSABJIAEkIDSBHBAV5o49ocEkAASQAJIQAYCOKDLABVFIgEkg
ASQABJQmgAO6EoTx/6QABJAAkgACchAAAd0GaCiSCSABJAAEkACShPAAV1p4tgfEkACSAAJIAEZCOCALgN
UFIkEkAASQAJIQGkCOKArTRz7QwJIAAkgASQgAwEXGWSiSCQAeXl5kJGRAenp6ZBx5w6k37wJGXfvwr2CA
vAsLgaXoiJwLSwEV/K9Cvlzyc8H19xcqEL+SkpKoMDTEwo8PKDQzQ3uu7pCAfkrdHGBgsqVIadSJahGvte
sXh2869eHmnXqgLe3N9SsWRPc3d2RPhJAAkjAkARwQDek2503+t69e5CYmAgJly9DfHIyVLl/HxokJECDa
9egQVISuJMB2pt0Q/+ULJnkACC5YUNIbtkSkhs3hvtVqkCTBg2gUevW0KhRI/Dy8lJSHewLCSABJKAYARz
QFUPNZkdF5Ez6GhmkL0VGwu3UVGgZGwutz56FOikp0JyYRP+0VOiBRIvr17k/a3ol+vjAlY4d4Vrz5lC3d
m1o27kztCSDf2Vy5o8FCSABJMAyARzQWfaeDLonkbPr48HBkEMG7DZnzkArcgbeNCcHmsrQlxoi6YFInaA
g6G3ReRa5vB/Tpg1c7tQJPMmA333AAGhIzvKxIAEkgARYIoADOkvekkHXuLg4bgAvIGffjx07Bi3I2fhIc
g/bSMWDHLB0iIri/ji7f/wRLpKz9tO9e4MLOYunA3zTpno5pDGSZ9FWJGAsAjigG8vfUEgmoh0PDYVL585
Bz5AQaEnOwJ8yGAOH5pIDmhYxMdwfV/eXX+ACuQcf3r8/tGnfHnr07QsuZIIeFiSgFIHgA0Fwc+MRqPdsf
xg4dLBS3WI/jBHAXyXGHCZG3btkdvnR/fshMz4e+pDPLmTGeRcxggzcpuWVK0D/aEkiM+tDhw6FGk2aQF/
yWZ3MtseCBOQicOPGDYgPOwNNNp6BhMY1IPahltCczAHBggQqEsABvSIRnXynk9lCDhyA5AsXYMi2bTA4K
0snlqlvRj1yQPTMn39yitz97TdYN2YMNHj4Yeg3ZAhOrlPfPbrS4D55emTfzt3QatExzq6GC4/B/toe8PK
0qVCFPMGBBQlYEsAB3ZKGDpavkxneh/fuhS7kfngvclkdi7wEqpMDJfPgfm7NGjhF7rsPHD4cWrRoIW/HK
N0QBAI2bIYGayKhcl4hZ2+l/EJo8Gck7KhWA559aYIhGKCR/AnggM6flWZr0rPxfTt2QBF5nGzA7t0wkRz
VY1GeQFtyAEX/7v/xBwSOGgWVyeNxw55+Gs/alXeFLno8efwEFIdcAq+LKeXs8bqUArlHL8HJtiega/du5
bbhF2MTwAGdYf/TbGw7NmwAH/J42eOHDjFsib5Up5nvhm7fDkD+jhw9Cinkcbinn38es9jpy82yWpNCHq+
MPhoBLch9c2ulLll/rl1daNKsKdStW9daFVxnQAI4oDPodHpfbfny5dCFJHsZfeoUgxYYR+Xe9ECL/F0gf
2f79IFnJ04ET/LcOxYkYIsATX0csHkrNCX3y+2Vhr+FQIBbZZg6cwaYTCZ7VXGbQQjggM6Qo+lAvoncp21
35AhMJWflWNgh8DDxF/0LIZMUb3foAM9OmoSTmthxn6Ka7t0RCLW2nwPXtBy7/VbJyIPapN7ehoEwYjQ+f
GoXlkE24oDOiKP3kJnqlY4fh3F79jCiMappjUC/w4cByN8h8pw7dO8Ow0ePtlYN1xmUwKVLlyA97AI0jIj
nRcD7eAIkP3YBLrZ5CNq1a8erDVbSLwEc0DXu26gTJ+AKScX6xMaN4ELuzWLRB4GBZPJiIXmscMPVq9CaZ
KJ7rBtObtKHZ8VbkZ2dDUcOHITWyyMECalP6h9tXB2akLwIVatWFdQWK+uLAL4PXaP+zCcvGVm9bBl4zZ0
Lo9euxcFco35yRi16gEZ9S31MfU19jsW4BLav3wiN/Y6DqahYEARav7FfBGxbt0FQO6ysPwJ4hq5Bn0aTe
60X9+2D59avh0oGy6uuQXfIrlJj8trZ55YsgR2ZmdBu2DDoQGbFYzEWgaOHj0CVAxfBMy5DlOEepJ170CU
IaRUM/QYNECUDG7FPAM/QNeRDmmd97YoV4Pr11/D0unU4mGvIN3KrQg/cqM+p7+k+QPcFLMYgEE9SMseGR
YHP3stOGVyHtKdyqDwsxiSAA7pG/J5Jzs5WL10Ko1auhOZ0whQWQxKgvh9FBnR6CZ7uE1j0TYAeuO0N3AU
NFhyVxNBGC0NhT8BOPCCUhCZ7QnBA14DPaLrW3WQgn0j+6Ks8sRibgEduLkwkgzrdJ+i+gUW/BGhq13rro
8AlR5oJr5WJnPpEXsCGLfqFhpbZJIADuk00ymw4TnKuXyY/3mPJyz5MxcImwyijIfaiBgG6L9B9gu4bdB/
Boj8Cp0liqILQS1A9+pakxlU/d5vIvQhUPhZjEcBJcSr6O5RkDyvZuhUGkfSgWJCANQKDdu2CE+T1t6FkB
nyfQYOsVcF1DBJIS0uDyJAwaLkuShbt6xG5kW3qkNSwzaBWrVqy9IFCtUcAz9BV8gk968onP9bdcDBXyQP
sdEv3Ebqv4Jk6Oz5zpOm2DZug8QJ5r7xQ+bQfLMYhgAO6Cr6ml8LSyA90n6AgFXrHLlkkQPcVus/gZVQWv
Vde531kElytnRegyp3s8hsk/kbl035of1iMQQAvuSvsZ5raMTEwEIaQH2csSEAIAXr5PahKFXDz8MA0n0L
AqVSXJgqib027c+cO3LweBynkM7voPnglZUGT0BuKaOVN+ontfA4Wx98Az8pVwKdOHajfoinUIZ8+Pj7g5
uamiB7YiTIEcEBXhjPXC03teJScaU0MCFCwV+xKTwQGkX1nbe3amOZTRafm5uZyA3XKrVtw80Yi3ElLhYL
CAvBMy4MqMXfAPS4d3BLvglvqP0+s0PfrNVNJ5+a/l7+0f5foQf+ukb/82p6Q37A65DXzhoJWPpBd2x1cK
7tAbe9a0KB5Y/CpV48b+D3IQSQW7RPAAV1BHwWSd5fTM/NKRUUK9opd6YkA3XeG7NwJgeQHdvyUKXoyTXV
b6AH37du3IeXWbbgZGw93MtIByNMGnik54HrlNrjFZ4AHObt2zcgt09WLLNE/Vgs96KB/1c/eJCZcKGcGv
SFA/2LJX0ENd8ht6AX5Tb3h/kM+kOvjCSaXylC7pjfUa9oY6tavx531e3mxTIMYynjBAV0hB4aHhEDjkye
hfnKyQj1iN3olUP/mTWhC9qWwVq2gV//+ejVTFruysrJg43p/gLwC8Lh1D6pcTQH3hExwJ2fULln/5NKvS
Xqnf1hKCbhm5gH9gwspAHvKZ7Sj1GhuOvpXWK0K5DUiZ/yNa0JB67qQXY8M/O5V4LkXxgMO9qUs5fwfJ8X
JSfdv2enp6XA1Kgq6hoUp0Bt2YQQCXci+dO30aaCPP2HhT4AOKq9MmwpVa3iBR3Qy1Nl/FapdTCk3mPOXh
jUrEnC5dx+qXboDdQ5cBfezSVCtRnWONw7mFUnJ8x0HdHm4lpMaQd5/3ZmcoWNBAlISoPvUcfJqXSzCCFS

qVAnGT34RHv3pTbj2yVAocscLlcII2q9NeV77eCh0+ukNjjPljUUZArgny8yZnkGlkbdpDcL87DKTNp74l
mSfiiIv4khNTYXaZKIcFmEEmjdvDi+/MRM2164JLptPQa1gOk0MizME0vq3gIJxXeDlSRMAB3JnSIpri4d
O4rjxbhVBssE9duQI7/pYEQkIIdCZJJ05Tq4AYRFHgA46z704EbqQs/WYT4ZAkYerOEEGb1VMuFF+nQnH5
1+ahIO5SvsDnqHLCJ4+h5pEZs0+fg2P/GXEbGjRLchZeih5fIrua/hMsfhdoWnTpjDlzVnkbL0WVNp6Emo
fwpjlSzNtUCsoHNsZppCzcpPJxLcZ1pOBAA7oMkA1i7xx4wY0SUoyf8VPJCALAbqPxcbGQtu2bWWRbxShd
DB69sUJED+gL+zfGgDNfjgs2VvQ9MiwyNMVYt8fBEPHPsnlRdCjjazZhAO6jB67fu4cNLt4UcYeUDQSAGh
Csg/Gnj+PA7pEO0OTJk1gyhu+sIWcrZu2n4LaZMY2lvIEUgc/BCVju8CUCePxrLw8GlW/4YAuI/44crm9X
1ycjD2gaCRABnSyjx0h+xoW6QjQs/Vxk16ABHK2vm/LDmj+YzBUJo9kGb0UVa0Cse8NgGHPPA2NGzc2Og7
N2Y8DukwuyczMhGoFBeCCWeFkIoxizQToPlbt/n3IyMiAmjUxHYqZixSfdNCaSu6tb/EhTxFsj4Ta+8onV
ZGiD1ZkpA5rAzCa3Ct/4Xk8K9eo03BAl8kxNI2kRx7JrIQFCShAwJNMisvJycEBXSbWz5BLy0kD+sHuzdu
g+f+CgSZQMUqh2d9uvDcQhpOz8kaNGhnFbCbtxAFdJrfRH9eq9+7JJB3FIoHyBDzJvkb3OSzyEWjYsCFMe
2MWbCVvKSsOOAV1dl+SrzONSL4zsi1UeqoLTH3hOY1ohGrYI4ADuj06TmyjP64e5BIoFiSgBAG6r+GArgR
pgLFkcEsm99Z39aJn62QmfJb+ztYLvei98oEwatwYaNCggTJgsRenCeCA7jRCOwLwmUw7cHCTpARwX5MUp
yNhdJCb9rovbK/rAwXkbN1np36eZkl5oh24krPyac8/6wgDbtcYARzQZXKIp6cn3KpRQybpKBYJlCeQS/a
1+mSfw6IsgdFk0LvZvy/s7LEVmpGZ8K53/3ljm7KaON9bgZcbxL47AJ58bizUr1/feYEoQXECOKDLhJwO6
NnVqskkHcUigfIEcsi+Rvc5LMoToIMfPVvfUa8eFOwk99Z3nFdeCSd7vPN0e3B9ogu8+twzTkrC5moSwFz
uMtGvWrUq5Lq7yyQdxSKB8gRyyL5G9zks6hHo3q835Latq54CTvSc16YudOvbywkJ2FQLBHBAl8kLNcgl0
HuurlBYubJMPaBYJFBKgO5jdF+j+xwW9QgkkRS87uQd6ywWN/Lucqo/FrYJ4IAuo/+akUtw8c2aydgDikY
CAPHkxSJN67J5Zqgn/8VduAKeV+8waVJVovcNoj8WtgnggC6j/5q3bw/x+MIMGQmjaEogvl07aPHIIwhDZ
QK3szLA83q6ylqI694jNh3u3MuEkpIScQKwlSYI4IAuoxuaN28OcfgMp4yEUTQlEE8SnjTDK0Gq7gw3b96
Eqhn3wVRUrKoeYjs3FZdAtfR8SE5m85aBWLv11g4HdBk9WqVKFWhMZsBea9VKxl5QtJEJXGvZEhqSy+34L
nR194LEhATwiGb7HrQ7vY+emKguSOzdKQI4oDuFz3Hj7gMHQmS/fo4rYg0kIIJAVP/+0GPQIBEtsYmUBOI
uXgWPq6lSilRcFtU/7lKM4v1ih9IRwAFdOpZWJdWqVQtqkxcaxDz0kNXtuBIJiCUQQ6781CJvA6P7GBb1C
ND7zink/rPn9TT1lJCgZ09yHz3lXgbeR5eApVoicEBXgDw9g4rs21eBnrALIxGIIld+epArQFjUJUDvn9P
7z/Q+NMvFRA5MvNLu4310hp2IA7oCzvP29oaHHnsMTvbCxA0K4DZEF6fIvtTy0UeB7ltY1CVA75+7ndHHv
WdqB7UHC5sEcEBXyG89ydlUQteucBNnvStEXL/d3CQTLePJvtSL3D/Hoj4Bev+86jW2L7ebKVa9hvfRzSx
Y/MQBXUGvPfn883Bg1CgoxuxxClLXV1d03znwxBNA9yUs6hPg7p9nZ4KHjM+f5zWqDtf+Mxjyl0yFa3OGA
P0uV/GIzYDU7LtQXMzm43dycWFFLr6cRUFP0VzbfQcPhoOpqTBk2zYFe8au9ELg0FNPQZ/HH8e87RpxKH1
u2yudPH8ucUIW+uazW1O6g3vrhjBi9FPl/J09Ixv27giE3EsJUO+PE+CaJd0b3krvo5c+j96ITObFwhYBH
NAV9ldbkjkun/woH7p/Hwbt2qVw79gdywQOkas7DZ98EtqRzHBYtEEgMZ7cPz8tzf3zYtdKkDKuE+Q91pg
M4k/CEBvpfOmJwTMTxnMAUmanwJ7tgeAWGQ8+W85CpftFToOh9iQlJAIO6E6jVFwADuiKIwfoRCbI5efmQ
mh+PvQJClJBA+ySNQKh5MpOLTKgP9q5s1XVFy1axK339fW1uh1XykMg7tJV8Ixx7vnztEGtIG3IQzBw2BA
YJDAJlY+PD7z06hTOuGvvXoNDe/eD94ErUPvQNdEGe9L76JevQveePUTLwIbqEMABXR3u0L13b25AP0HO1
LuFhKikBXbLAoETZEKlGxnM6T5jrRyLCIPLlW/BfSiE4JBgGNBvgLVquE5iAvQ+c2pOFtS5kSFYclY7H7g
54TF4tFsXGCDRwNmSZA1s6fsaADmmOxEeAadPnIJ666Og+oUUQfp5EnvisrO4++iVKuE0K0HwVK6MA7qKD
ugzaBCcIO+xPlS9OgzauVNFTbBrrRKgl9npmbmtwTw+Ph5Cz0ZACFzkTGh6pQ40atAIWgk809Oq/VrWi7t
/nkaeP+d5/zyvYXVIfqUrNGz7EAwfPgRcyStv5SrdyEEC/SucWUjO2g9AIrmS0GDlCXBPuuu4S2JPtTt53
OtUG5PERVjYIYADusq+6kaeJ44ljyFtIdm+xq5ZQ5JT4OxSlV2iie5LyJnR1kmToNeYMUBf8mOt5OXlQeD
unbAZIso2by4MB6+D7vBS3brg5eVVth4XpCfAPX9+OsGu4EKvKnBzcjdwb9sYhpP74tWqVbNbX+qNLi4uM
PSJEQDk796Me7CX3G/PvZwADVadBBc7k+ncz5Y+j44DutQekVceDujy8uUlnf5ge0+ZAmvd3OCZtWvBIye
HVzuspE8CuZ6esGXCBBhHBvQaNWrYNPJP/zUQUHIKCkz/TIQqhhLYUhwBLhsrw/Qpr4LJZLLZHjc4RyCe5
D239vx5iWtluD3mEcjr2pSb3DaYHFxpodCDiXGTXuBUuT37NjeZzt08ma6g/IkEnRcQfzkGemIyLC24jrc
OOKDzRiVvRfrDPXn6dNhABvUue/ZA8xh8SYK8xLUpPZZcKj81YgRMnjwZ6NmVrRKwJxBC8y7AbdODl1AzT
DlwuPA8+ARuh7FPjbElAtc7QcB8/9zH4v55Wv8WkDa8DQwYOhgGavzdDXXJQcbk6VM5AjHvxsDhfQeg9v4
r4H24dDKdZ1wmxOXcw/voTuwjajS1/YuhhjYG75P+gE+cOhWiSRawHfv2wZPr10MlnvfnDI6OefOLyZl04
AsvQLthw2Bip0527TkbHQ0RCdEQbUqwWe8yJEP47XPQ5FQT6Nqli816uEEcgaSkJHKfORfuta3DTW7r2KU
zjOnVg8krInS+Bf0rmVnCTaY7cyIS6pPJdP/f3rk/t3Fdd/y7sq3armRKlk2JsuWJ7LqRJ3aaZuzEddK0H
TtOMpNmJmnjSVqPm/Gkk8nkh/wL/TPyB9TiAwBf4vslvgmAJPgACBKS+BBEk7JISZTFNwFsz4VrWrQokiA
WwO7F985AAnb3nnvO51zy8GLvnnOc99EPNzly2IsBPYfwHzX06/IL/VV5Xr1MVu3v1NbiReZWfhQqLY7Py
sajXnm+/ANZle9X13xxcRHt3k40mMP72t4aD+L0cAGKZI/G2bNn972eFxycgPLTkfOF+HHbnzK6ue3gGqV
/pbo989bb30++tv6whZa6hn3nY/qjUoKVBB77+c9//j9vvvmmlTIpywICarX+d7KyispXd17ZBf9yJILHu
GHOArL2ERGTXc41sio/JoH8J5JsaK+v2JXWsVgM/1v8CVzwYQtf3Tffy6Ir5jyevLaOb7/+Bh5jyuG9UKV
0TiV3ef3b+jJVc+XVC9/ckaEuJUC8OOsEBgYGwBV61rGnNuB35I8t9WqUr8SO9PfjR3J/nc35BDp++lPgr
bfw61/84sDGFJeXosUMYtU4eKrPDQn9tYkA/sr9JH73Hx8deCxeSAIk4DwCDOgO8dlP5PGlTXke2SMr9sK
xMfywvd0hmlPNBwmE5XbKhFRJ+zfZwX706NEHT+35vkMSxviWxhE1Us9KdtO4B+/qBF683Ir3/uXdPcfhS
RIgAecSYEB3kO9UAPitPN62Ko+1lckz62/09eG10VEHWZC/qo7I7ZOApG39+OOP8fcpBHJFbGpqCn2RQfi
Nwz/5EDCncW5mBOciL0LVE2AjARLQjwADugN9+rQ8p6wecVv/8ENUu1wolKD+NlfstvSkV7IB3pJV+b9Ku
dPvSVbAVNvy8jIaW5tQhYFUuz50fXWsHye7noZ6ZOnkyZMPnecBEiABZxNgQHew/56UAPFr2VAVj8clA1Q
1zGAQ/yj32I9Kfni23BHYlBX42A9+gLuSe/19uUeezma0i64SCeaDsgVuZ+KPw1hnSid3wovHPEfw3x//H

szTfRiK7EMC9iXAgG5f3xxYMxUwfvbLXwLympYA39HUhO/K1/HflHvtbNkjEPnWtxCQIP5P77+Pt8+fT3v
gippKdG2FccdYTlvWlwKWZUNdWyKEk1Xl+OCX//7lYf5PAiSgAQEGdA2c+KAJ5yWQnP/DHxCXr+S7W1sxN
z6O92T1/sznD2cUe7Af3x+OwOeSL71VNiwWvfYafvjuu/i2RY+GBYYC8N0MIWLMH06xPXpNGwvw3RnDS/5
zePt739/jSp4iARJwEgEGdCd5KwVd1apdrRSlrBPuy0asNsk8d08qc73T0oLTN2+mIImXfp3AZ5Kopfe99
1Bw7hx+IP//RvIEWNlUFa/OQC/azMx9w9KVmMDZ0LM4e6YIL730kpXqUxYJkECOCDCg5wh8NodVVbd+9qt
fJYeM/elP6O/tRUS+jv++1GF/+cqVbKri2LGmXn0VPnnc7Jvytfpb77yD3+yRZz0dIzdl/0NVTfWOCmrpy
Nurryfuw183HsV/ffgRnnrqqb0u5TkSIAEHEGBAd4CTrFRRZSP7hx/9KPnCH/+IaDSK/s5ObN2+je/Ifff
z8ogU8j1/vKTAnH75ZYzI/fDHT53CW8LrNVnFvmalIx4h6xPZBKfSuq4bW4+4wrrDMdlqV20O4qjrKD7+6
ItCHdZJpyQSIIFsE2BAzzZxm42nvm59SR5/S7Y//xmq6MRAVxdWbt3Cq/I43CuScvbptTWbaW2tOquyOp2
UZ7OvyuNlTz//fDKAX5Dc5xesHWZfac2XW9C3EsacsbTvtVZdsGjcR/fGOIqaGvCz9yV7HRsJkIBjCTCgO
9Z1mVFcFfH4heQX/7KpR+Kmp6cxEQjglqziz8/M4G/l8bjnFha+vMRR/y9KwL76xhuY+sY3UCir7wuS8EV
tJHxT9hzksqLBFbn14Z0axrBxPes8Q7iBl2ZHcS70Il5//fWsj88BSYAErCHAgG4NR22lqM11fyPpZtXrw
aYSnsxKFbhPr17FDdnEpZ59L5LPRfKVfZGs8p/cOHi+8Qflpvt+XapgzcsfJfPylfm8VDFTz4SfKyrCC3I
P/EX5/MKxY3hBBvnndAeysP/S0hKaO9tQawYslJqaqLr4EJ7zHcfp06fxvPzRw0YCJOA8AgzozvOZLTQ+J
oHxwoULydduCq2vr0MFquRLSn7ekaC/dO8elqVi2NNSNe5xWfk/Ie+f2NpK/jHwuPwB8IT0eULS2qq2Jdn
wtiRxTkwCtArKW1KZbEvu/8fkD4zVI0dwTN6fkPKyJ2XH+UkJQCdOnEi+CqRPgfTP9tflSaUP8U9CWJR6y
lCe8CFhqNQvuWuuuBePVz2G3//u430rv+VOS45MAiTwKAIM6I8iw+NpEVBZ7M5IsFUvtkcTKK10oz0+hvv
G+qMvytKZNWMTzYlRHC934T8/+G2WRuUwJEACVhE4YpUgyiEBEkiNgNfvRf/tMCaNW6l1zODVN6Sam+/zM
Lp6ujI4CkWTAAlkggADeiaoUiYJ7EPghiT56Q750Y3IPldm/7TXvIbeSACTk4ev7pZ9rTkiCZAAAzrnAAl
kmYDaX1DbUIeKuD/LIx98uErRramtGWrzIxsJkIAzCDCgO8NP1FIjAp+UFaMGAWxJYhe7NlXdrVJKtpa4S
+2qIvUiARL4GgEG9K8B4UcSyCSBmsY69KyHcQv2L5ZzFyvokGpvVbWXMomEskmABCwiwIBuEUiKIYH9CAR
DIfhujCJkzO53qW3ORzAnVd+CCAwN2UYnKkICJLA7AT62tjsXHiUBSwksyrP47d5ONJojlsrNhrCWRBCnA
wUokkcQiyRJDxsJkIA9CXCFbk+/UCuNCMQkgY67wgOPad9NcPvhdie8qJav3lU1ODYSIAF7EmBAt6dfqJV
GBErKy9BihrCK3KTDtQLlBmKoTwyhmJvkrMBJGSSQEQIM6BnBSqEk8AWBju5O+O6NI2osOh6JqgLnXZ1Aa
3ub422hASSgIwEGdB29SptsQWBKCtX0XQnAh2u20McKJQbNKfimhxGRsrpsJEAC9iLATXH28ge10YSASsj
S0NqEKvRrYtFXZlTF+nGi6+lkZTZVFIeNBEjAHgS4QreHH6iFZgSKXSWoMgckdUxCM8sAVRPOI9Xhyjwuq
GpxbCRAAvYgwIBuDz9QC40IVNRUoVMSstw1VjSyaqcpqjpcWzwET3XFzhP8RAIkkDMCDOg5Q8+BdSQwGJB
75jdDiBjzOpq3w6YpqRLnWxyDv9+5j+PtMIgfSMDhBBjQHe5Aqm8fAvPz8+ga6kWbPKKWL63THEd30A9VP
Y6NBEggtwQY0HPLn6NrQkAlXKmqqU7eW9bEpAOb4Yl7k9XjVBU5NhIggdwRYEDPHXuOrBGBi+4SNJjDkjo
mppFVBzNFVY1T1eMuykZANhIggdwRYEDPHXuOrAmB5sst6Fseh0q8kq9NVY/rWR9HfVNDviKg3SSQcwIM6
Dl3ARVwMgGVYMUriVaGjBknm2GJ7kFE4ZsNYmxszBJ5FEICJJAaASaWSY0XryaBbQJLS0to6WyTr5sHt4/
l+5u6eADPe48nk84899xz+Y6D9pNAVglwhZ5V3BxMFwIqoUqppwzlUkFNJVph+4qASyqzeSrLoarMsZEAC
WSPAAN69lhzJI0IlFW6cTk+BpVghW0nAVVVrjkxirIK984T/EQCJJBRAgzoGcVL4ToS8Pq98N8JQyVWYdu
dQNS4LVXmwuju7d79Ah4lARKwnAADuuVIKVBnAtFoNJlIpdtktbH9/NxnXkXvxCCmp6f3u5TnSYAELCDAg
G4BRIrIDwJra2uobaxHRYKpTg/q8Yq4Hw0tTVDV59hIgAQyS4ABPbN8KV0jAp+4ilFjBqASqbAdjICqNld
p9qPEXXqwDryKBEjg0AQY0A+Njh3ziUBNY10yccqC8Xk+mW2JrarqnKo+V1V3yRJ5FEICJLA7AQb03bnwK
AlsEwiGQvDdGMWYMbt9jG9SIzCBOfjnQxgaHk6tI68mARI4MAEmljkwKl6YjwQWFhbQ7u1EozmSj+ZbarN
6lO10oABFZ87gjLzYSIAErCXAFbq1PClNIwIqMYpKkOJO+DSyKremuKUym6pKp6rTsZEACVhLgAHdWp6Up
hGBkvIytEht8zWDwccqt67LlsL6xDBKJMseGwmQgLUEGNCt5UlpmhDo6O6EVxKjRI1FTSyyjxlzxl14V8b
R1nHZPkpRExLQgAADugZOpAnWEpiamkJfZBB+TFormNK2CQyYU/BODuPKlSvbx/iGBEggPQLcFJceP/bWj
IBKgNLQ2oQqDGhmmf3MqY7340TnU8nKbAUFBfZTkBqRgMMIcIXuMIdR3cwSKHaVJoO5SojCllkCCalTp6r
VlbrLoKrXsZEACaRHgAE9PX7srRGBipoqSYAyhrtY0cgqe5vyOdaSVevKL1XaW1FqRwIOIMCA7gAnUcXME
xgMBOC7GULEmM/8YBxhB4FJ4zP4F8Pw9/fvOM4PJEACqRFgQE+NF6/WkMD8/Dy6hnrRJo+oseWGQHtiDD1
BH2ZnmY0vNx7gqDoQYEDXwYu04dAEVIITlejEw+Qxh2ZoVUdP3Iea+lqsr69bJZJySCCvCDCg55W7aezXC
Vx0l6ABI9hA7Oun+DnLBDbFB6qaXTErs2WZPIfThQADui6epB0pE2i+3II+SXAyJ9vg2OxB4JZUs+tZC6O
+udEeClELEnAQAQZ0BzmLqlpHIBKJwDc9giHMWCeUkiwhMIoo/FLdLhwOWyKPQkggXwgwsUy+eJp2bhNYW
lpCa9dlXDKZPGYbis3e1MYDONV7LJl05tSpUzbTjuqQgD0JcIVuT79QqwwRUAlMSj0ueCShiZmhMSjWGgI
e0wd3hQfxeNwagZRCApoTYEDX3ME0bycBV5VHEpmEcF8SmrDZm8CKbFVsSQRRVuG2t6LUjgRsQoAB3SaOo
BqZJ9Dn88J/O4wp41bmB+MIlhC4LtXu/PfG0dPXa4k8CiEBnQkwoOvsXdq2TSAajaIn5EeXObF9jG+cQaA
nEUHv+ACmp6edoTC1JIEcEWBAzxF4Dps9Amtra6hrrEd5wp+9QTmSpQQqJOlMQ0sTVlaYZ99SsBSmFQEGd
K3cSWN2I3DRVSw72gOStoSbq3bj44RjMal+V2X2o9hV4gR1qSMJ5IQAA3pOsHPQbBGoaxD9/o4FiRhCZu
zCdwxVtC5GUZ1fY2zDaH2JJAhAgzoGQJLsbknEAwF4Z8NYsxgwY/ce8MaDSaMOfjnghgZHbFGIKWQgEYEm
FhGI2fSlK8ILCwsoN3bhQZz+KuDfKcFgabEKAr7C3Dm9Jlk4hktjKIRJGABAa7QLYBIEfYiEIvF4Kksh5s
V1OzlGAu1cSe8qLxUha2tLQulUhQJOJsAA7qz/UftdyFQUl6GZjOINWNzl7M8pAOBdWyh0RxBiadMB3NoA
wlYQoAB3RKMFGIXAh3dnfDeC+OGcdsuKlGPDBGYxR14l8dxubM9QyNQLAk4iwADurP8RW33IDA5OQnvlQD
8mNzjKp7SiUC/KT6fHMLVq1d1Mou2kMChCHBT3KGwsZPdCCwvL6OxrRmV6LebatQnwwQqt/pR0P5UcoPcM
888k+HRKJ4E7EuAK3T7+oaapUCg2FWKKgxI6phECr14qQ4ETMNEhXwvU+oug2myhp4OPqUNhyPAgH44bux
lIwLllyrRsRXCXTAtqI3cklVV7kn1vMuxENRcYCOBfCXAgJ6vntfE7sGA3DP/bAxXjJuaWEQzDkvgmvEZf
Atj6B8YOKwI9iMBRxNgQHe0+/Jb+bm5OXQN9aHNDOU3CFq/TaA9MYbuUR9mZ5kdcBsK3+QNAQb0vHG1XoZ

```
ubm6iuvYSPEweo5djLbDGHetDTUMdNjY2LJBGESTgHAIM6M7xFTV9gMBFdwkaMIINSTDCRgIPEtgy4qiT6
nrFMkfYSCCfCDCg55O3NbG1qa0FvcthzMk2ODYS2I3ATdxDz+o4GqWGOhsJ5AsBBvR88bQmdkYiEfhmhjF
sXNfEIpqRKQIjuA5fdATj4+OZGoJyScBWBJhYxlbuoDJ7EVhaWkJr12XUmIN7XcZzJLBNoDY+hFM9x5NJZ
5599tnt43xDAjoS4ApdR69qaFMikUCZxwWP6QNTh2jo4AyZZMpsUZXZXOVuqDnERgI6E2BA19m7GtnmqvK
gLR7CfaxrZBVNyQaBFWMDrYkQyirc2RiOY5BAzggwoOcMPQc+KIE+nxe+22FMGbcO2oXXkcAOAjPGAnx3w
+jp691xnB9IQCcCDOg6eVNDW6LRKHpCfnSbExpaR5OySaDHjKBvfAAzMzPZHJZjkUDWCDCgZw01B0qVwNr
aGuoa61Ge8KfaldeTwK4EPDEf6psbsLq6uut5HiQBJxNgQHey9zTX/aKrBNWyoz0mNdTYSMAKAnEjgUuQp
DMyt9hIQDcCDOi6eVQTe2ob69CzMY5F474mFtEMuxBYlK2VXRthXKqvtYtK1IMELCHAgG4JRgqxkkAwFIR
vNogQblgplrJIYJtA2PgUvrkgRoOj28f4hgScToCJZZzuQc30X1hYQIevGw2JYc0sozl2I9CUGMFpfwHOn
D6DwsJCu6lHfUggZQJcoaeMjB0yRSAWi8FTWQ5X3JupISiXBHYQUElnKqoroeYeGwk4nQADutM9qJH+peV
laEEQa8amRlbRFDsTWMMmGmWlXuIps7Oa1I0EDkSAAf1AmHhRpgl0dHfC+/k4orid6aEonwR2EJg17sB7P
4zLne07jvMDCTiNAAO60zymob6Tk5PwXgnAZ17T0Dqa5AQCfnMSvmvDuHaNc9AJ/qKOuxPgprjdufBolgj
cv38fTW3NqACTx2QJOYd5BIHKuB8F7U8mK7MdP378EVfxMAnYlwBX6Pb1jfaamaaJEncZKjGABGuoae9vu
xuo5mB5ol+SzpRCzU02EnAaAQZ0p3lMI30ra6vQsRXCXaxoZBVNcTKBe8YqOmIhVNRUOdkM6p6nBBjQ89T
xuTY7MCT3zD8bwxXjZq5V4fgksIPAVeMz+G+NYTAwuOM4P5CA3QnwHrrdPaShfnNzc+ga6kOb1KhmIwE7E
lBzs2j4BIrOFOHs2bN2VJE6kcBDBLhCfwgJD2SSwObmJi7V1cAd92VyGMomgbQJqARHaq5ubGykLYsCSCA
bBBjQs0GZY2wTUBXU6hJD2MDW9jG+IQE7EtiUOn+1MleL3azMZkf/UKeHCTCgP8yERzJEoLmtBb0r45g3l
jI0AsWSgLUEbspc7V2dQFNrs7WCKY0EMkCAAT0DUCnyYQKRSAS+mREMGzMPn+QRErAxgWHMwHd9BBMTEzb
WkqqRAMBNcZwFGSdw9+5dtHRdxiVzIONjcQASyASBmngAp3qOJZPOnDx5MhNDUCYJpE2AK/S0EVLAXgQSi
QRc5W6Umz6mjtkLFM/ZmoAps9ed8KHM44Ka02wkYEcCDOh29IpGOrkqPWiNB3Ef6xpZRVPykcCyzGH1OJu
a02wkYEcCDOh29IomOvX5vPDdCWPaWNDEIpqR7wTUXFZzWs1tNhKwGwEGdLt5RBN9otEoesf60W1yI5EmL
qUZ/09AzekemdvXr18nExKwFQEGdFu5Qw9l1tbWUNdYDw+Tx+jhUFrxEIFymdt1TfVYXV196BwPkECuCDC
g54q8xuOq5DHV5qCk5YhrbCVNy2cCam5fMgOSdKY0nzHQdpsRYEC3mUOcrk5NYx2618NYNO473RTqTwJ7E
lBzXM11NefZSMAOBBjQ7eAFTXQIhoLwzwYxZsxqYhHNIIG9Cai57v80CDX32Ugg1wSYWCbXHtBk/IWFBXT
4utGQGNbEIppBAgcj0BAfRqHvGZwuPI3CwsKDdeJVJJABAlyhZwBqvomMxWIor6qQxBt8lCfffE97vyCg5
n5FdSXUzwIbCeSKAAN6rshrNG5puQvN5ihWsamRVTSFBA5OQM199TOgfhbYSCBXBBjQc0Vek3E7ujrRd28
MUdzWxCKaQQKHI6B+BtTPgvqZYCOBXBDgPfRcUNdkzMnJSYyPhRE7Esd38Q1NrKIZJHB4AupxNvUz8eLZF
/DKK68cXhB7ksAhCDCgHwIau3xBQP3C4i8tzgYSIAESsAcBfuVuDz9QCxIgARIgARJIiwADelr42JkESIA
ESIAE7EGAAd0efqAWJEACJEACJJAWAQb0tPCxMwmQAAmQAAnYgwADuj38QC1IgARIgARIIC0CDOhp4WNnE
iABEiABErAHAQZ0e/iBWpAACZAACZBAWgQY0NPCx84kQAIkQAIkYA8CDOj28AO1IAESIAESIIG0CDCgp4W
PnUmABEiABEjAHgSSqV//8pe/2EMbakECJEACJEACJHAoAv8HwhKKXQyVE4cAAAAASUVORK5CYII=",
Base64.DEFAULT);

        //create a bitmap factory options
        bmfOptions = new BitmapFactory.Options();
        // decode a bitmap using decodeByteArray default Options
        Bitmap bm = BitmapFactory.decodeByteArray(bytes, 0, bytes.length,
bmfOptions);
        // set the decoded bitmap to the ivOne
        ivOne.setImageBitmap(bm);

        //create a bitmap factory options
        bmfOptions = new BitmapFactory.Options();
        // sample the image down by a factor of 4
        bmfOptions.inSampleSize = 4;
        //set the decoded Bitmap to be mutable
        bmfOptions.inMutable = true;
        bm = BitmapFactory.decodeByteArray(bytes, 0, bytes.length, bmfOptions);
        // set the decoded bitmap to the ivTwo
        ivTwo.setImageBitmap(bm);


        //create a bitmap factory options
        bmfOptions = new BitmapFactory.Options();
        // set the original image density
        bmfOptions.inDensity = DisplayMetrics.DENSITY_MEDIUM;
        bmfOptions.inScaled = true;
        //set the decoded Bitmap to be mutable
        bmfOptions.inMutable = true;
```

```
        bm = BitmapFactory.decodeByteArray(bytes, 0, bytes.length, bmfOptions);
        // set the decoded bitmap to the ivThree
        ivThree.setImageBitmap(bm);




        /*
            set the activity content view
            to ll : linearlayout
            it will take the full width and height by default
        */
        setContentView(ll);

    }


}
```

Animations

Main_activity.xml

```
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
     tools:context=".MainActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="BMSCE"
        android:id="@+id/textView"
        android:textSize="35dp"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Animations"
        android:id="@+id/textView2"
        android:textColor="#ff3eff0f"
        android:textSize="35dp"
        android:layout_below="@+id/textView"
        android:layout_centerHorizontal="true" />

    <ImageView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/imageView"
        android:src="@drawable/ic_launcher_background"
        android:layout_below="@+id/textView2"
        android:layout_alignRight="@+id/textView2"
        android:layout_alignEnd="@+id/textView2"
        android:layout_alignLeft="@+id/textView"
        android:layout_alignStart="@+id/textView"/>

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="zoom"
        android:id="@+id/button"
        android:layout_below="@+id/imageView"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:layout_marginTop="40dp"
        android:onClick="clockwise"/>
```

```
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="clockwise"
        android:id="@+id/button2"
        android:layout_alignTop="@+id/button"
        android:layout_centerHorizontal="true"
        android:onClick="zoom"/>

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="fade"
        android:id="@+id/button3"
        android:layout_alignTop="@+id/button2"
        android:layout_alignParentRight="true"
        android:layout_alignParentEnd="true"
        android:onClick="fade"/>

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="blink"
        android:onClick="blink"
        android:id="@+id/button4"
        android:layout_below="@+id/button"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true" />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="move"
        android:onClick="move"
        android:id="@+id/button5"
        android:layout_below="@+id/button2"
        android:layout_alignRight="@+id/button2"
        android:layout_alignEnd="@+id/button2"
        android:layout_alignLeft="@+id/button2"
        android:layout_alignStart="@+id/button2" />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="slide"
        android:onClick="slide"
        android:id="@+id/button6"
        android:layout_below="@+id/button3"
        android:layout_toRightOf="@+id/textView"
        android:layout_toEndOf="@+id/textView" />

</RelativeLayout>
```

Mainactivity.java

```
package com.example.animation2;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;

import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.view.animation.Animation;
import android.view.animation.AnimationUtils;
import android.widget.ImageView;
import android.widget.Toast;

public class MainActivity extends Activity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }

    public void clockwise(View view){
        ImageView image = (ImageView)findViewById(R.id.imageView);
        Animation animation =
AnimationUtils.loadAnimation(getApplicationContext(),
                R.anim.myanimation);
        image.startAnimation(animation);
    }

    public void zoom(View view){
        ImageView image = (ImageView)findViewById(R.id.imageView);
        Animation animation1 =
AnimationUtils.loadAnimation(getApplicationContext(),
                R.anim.clockwise);
        image.startAnimation(animation1);
    }

    public void fade(View view){
        ImageView image = (ImageView)findViewById(R.id.imageView);
        Animation animation1 =
                AnimationUtils.loadAnimation(getApplicationContext(),
                        R.anim.fade);
        image.startAnimation(animation1);
    }

    public void blink(View view){
        ImageView image = (ImageView)findViewById(R.id.imageView);
        Animation animation1 =
                AnimationUtils.loadAnimation(getApplicationContext(),
                        R.anim.blink);
        image.startAnimation(animation1);
    }

    public void move(View view){
        ImageView image = (ImageView)findViewById(R.id.imageView);
        Animation animation1 =
                AnimationUtils.loadAnimation(getApplicationContext(),
```

```
R.anim.move);
        image.startAnimation(animation1);
    }

    public void slide(View view){
        ImageView image = (ImageView)findViewById(R.id.imageView);
        Animation animation1 =
                AnimationUtils.loadAnimation(getApplicationContext(),
R.anim.slide);
        image.startAnimation(animation1);
    }
}
```

/res/anim/myanimation.xml

```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android">

    <scale xmlns:android="http://schemas.android.com/apk/res/android"
        android:fromXScale="0.5"
        android:toXScale="3.0"
        android:fromYScale="0.5"
        android:toYScale="3.0"
        android:duration="5000"
        android:pivotX="50%"
        android:pivotY="50%" >
    </scale>

    <scale xmlns:android="http://schemas.android.com/apk/res/android"
        android:startOffset="5000"
        android:fromXScale="3.0"
        android:toXScale="0.5"
        android:fromYScale="3.0"
        android:toYScale="0.5"
        android:duration="5000"
        android:pivotX="50%"
        android:pivotY="50%" >
    </scale>

</set>
```

/res/anim/clockwise.xml

```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android">

    <rotate xmlns:android="http://schemas.android.com/apk/res/android"
        android:fromDegrees="0"
        android:toDegrees="360"
        android:pivotX="50%"
        android:pivotY="50%"
        android:duration="5000" >
    </rotate>

    <rotate xmlns:android="http://schemas.android.com/apk/res/android"
        android:startOffset="5000"
        android:fromDegrees="360"
        android:toDegrees="0"
        android:pivotX="50%"
        android:pivotY="50%"
        android:duration="5000" >
    </rotate>

</set>
```

/res/anim/fade .xml

```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android"
    android:interpolator="@android:anim/accelerate_interpolator" >

    <alpha
        android:fromAlpha="0"
        android:toAlpha="1"
        android:duration="2000" >
    </alpha>

    <alpha
        android:startOffset="2000"
        android:fromAlpha="1"
        android:toAlpha="0"
        android:duration="2000" >
    </alpha>

</set>
```

/res/anim/blink .xml

```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android">
    <alpha android:fromAlpha="0.0"
        android:toAlpha="1.0"
        android:interpolator="@android:anim/accelerate_interpolator"
        android:duration="600"
        android:repeatMode="reverse"
        android:repeatCount="infinite"/>
</set>
```

/res/anim/move .xml

```
<?xml version="1.0" encoding="utf-8"?>
<set
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:interpolator="@android:anim/linear_interpolator"
    android:fillAfter="true">

    <translate
        android:fromXDelta="0%p"
        android:toXDelta="75%p"
        android:duration="800" />
</set>
```

/res/anim/slide .xml

```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android"
    android:fillAfter="true" >

    <scale
        android:duration="500"
        android:fromXScale="1.0"
        android:fromYScale="1.0"
        android:interpolator="@android:anim/linear_interpolator"
        android:toXScale="1.0"
        android:toYScale="0.0" />
</set>
```