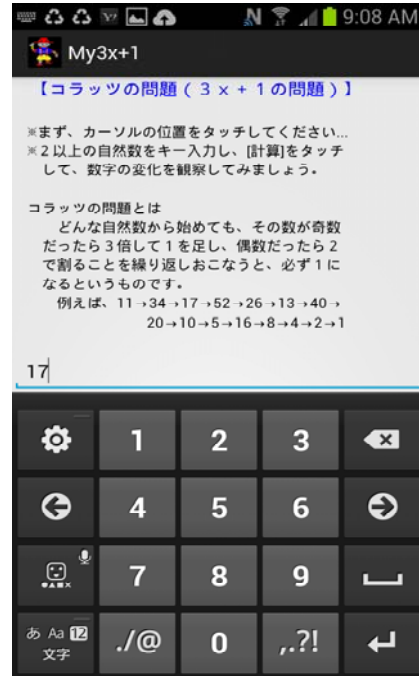


```
//-----
//
//      数字 (テキスト) を入力して計算する方法
//      3 x + 1 の問題
//      (コラッツの問題)
//      Ver2
//      Copyright (C) K.Niwa 2014.08.10
//-----
```



【エミュレータ画面例】



スマートフォン画面例
(Android 4.0)

【アプリの概要】

どんな自然数から始めても、その数が奇数だったら3倍して1を足し、偶数だったら2で割ることを繰り返しおこなうと、必ず1になるってほんとうでしょうか。

これをコラッツの問題と言い、未だ解けていない難問です。

例えば、11 → 34 → 17 → 52 → 26 → 13 → 40 → 20 → 10 → 5 → 16 → 8 → 4 → 2 → 1
 色々な自然数から始めて確かめてみましょう。

【 1 】 P3xplus1Activity.java

```
package jp.seitoku.p3xplus1;           //パッケージ名を指定

import android.app.Activity;           //ライブラリを読み込む
import android.graphics.Color;
import android.os.Bundle;
import android.text.Editable;
import android.view.*;
import android.view.View.OnClickListener;
import android.widget.*;

public class P3xplus1Activity extends Activity { //Activityクラスを継承したKorattsu2Activityクラス

    TextView text, text0, text1, text2, text3, text4, text5, text6, text7, text8, text9, text10, text11, text12, text13, text14, text15; //テキストを表示する「テキストビュー型」で宣言する
    EditText edit1; //テキストを入力する「エディットテキスト型」で宣言する
    Button button1, button2; //「ボタン型」で宣言する

    int n; //整数型で宣言する 入力する最初の自然数
    int res; //整数型で宣言する 1に達するまでの途中のそれぞれの計算結果
    int flag=0; //整数型で宣言し初期化する 初期化識別子
```

```

    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        LinearLayout layout = new LinearLayout(this);           //リニアレイアウト型で宣言する
        layout.setOrientation(LinearLayout.VERTICAL);          //部品の並べ方を縦に宣言する

        text = new TextView(this);                             //テキストビュー型として実体化
        text.setTextColor(Color.BLUE);                         //テキストを白色に指定する
        text.setTextSize(16f);                                 //テキストの大きさを15
        text.setText(" 【コラッツの問題 (3 x + 1の問題) 】 "); //テキストを表示する
        layout.addView(text);                                  //テキストビ

        text1 = new TextView(this);                             //テキストビュー型として実体化
        text1.setTextColor(Color.BLACK);                       //テキストを白色に指定する
        text1.setTextSize(15f);                                //テキストの大きさを15
        text1.setText("");                                     //テキストを表示する
        layout.addView(text1);                                 //テキストビューをレイ

        text2 = new TextView(this);                             //テキストビュー型として実体化
        text2.setTextColor(Color.BLACK);                       //テキストを白色に指定する
        text2.setTextSize(13f);                                //テキストの大きさを15
        text2.setText(" ※まず、カーソルの位置をタッチしてください... "); //テ

        text3 = new TextView(this);                             //テキストビュー型として実体化
        text3.setTextColor(Color.BLACK);                       //テキストを白色に指定する
        text3.setTextSize(13f);                                //テキストの大きさを15
        text3.setText(" ※2以上の自然数をキー入力し、[計算]をタッチ"); //テキストを表示する
        layout.addView(text3);                                 //テキストビューをレイ

        text14 = new TextView(this);                             //テキストビュー型とし
        text14.setTextColor(Color.BLACK);                       //テキストを白色に指定する
        text14.setTextSize(13f);                                //テキストの大きさを15
        text14.setText(" して、数字の変化を観察してみましょう。 "); //テキストを表示する
        layout.addView(text14);                                 //テキストビューをレイ

        text4 = new TextView(this);                             //テキストビュー型として実体化
        text4.setTextColor(Color.BLACK);                       //テキストを白色に指定する
        text4.setTextSize(13f);                                //テキストの大きさを15
        text4.setText("");                                     //テキストを表示する
        layout.addView(text4);                                 //テキストビューをレイ

        text5 = new TextView(this);                             //テキストビュー型として実体化

```

```

する
    text5.setTextColor(Color.BLACK); //テキストを白色に指定する
    text5.setTextSize(13f); //テキストの大きさを15
fに指定する
    text5.setText(" コラッツの問題とは"); //テキストを表示する
    layout.addView(text5); //テキストビューをレイ
アウトに貼り付ける

    text6 = new TextView(this); //テキストビュー型として実体化
する
    text6.setTextColor(Color.BLACK); //テキストを白色に指定する
    text6.setTextSize(13f); //テキストの大きさを15
fに指定する
    text6.setText(" どんな自然数から始めても、その数が奇数"); //テキストを表示する
    layout.addView(text6); //テキストビューをレイ
アウトに貼り付ける

    text7 = new TextView(this); //テキストビュー型として実体化
する
    text7.setTextColor(Color.BLACK); //テキストを白色に指定する
    text7.setTextSize(13f); //テキストの大きさを15
fに指定する
    text7.setText(" だったら3倍して1を足し、偶数だったら2"); //テキストを表示する
    layout.addView(text7); //テキストビューをレイ
アウトに貼り付ける

    text8 = new TextView(this); //テキストビュー型として実体化
する
    text8.setTextColor(Color.BLACK); //テキストを白色に指定する
    text8.setTextSize(13f); //テキストの大きさを15
fに指定する
    text8.setText(" で割ることを繰り返しおこなうと、必ず1に"); //テキストを表示する
    layout.addView(text8); //テキストビューをレイ
アウトに貼り付ける

    text9 = new TextView(this); //テキストビュー型として実体化
する
    text9.setTextColor(Color.BLACK); //テキストを白色に指定する
    text9.setTextSize(13f); //テキストの大きさを15
fに指定する
    text9.setText(" なるというものです。"); //テキストを表示する
    layout.addView(text9); //テキストビューをレイ
アウトに貼り付ける

    text10 = new TextView(this); //テキストビュー型とし
て実体化する
    text10.setTextColor(Color.BLACK); //テキストを白色に指定する
    text10.setTextSize(13f); //テキストの大きさを15
fに指定する
    text10.setText(" 例えば、11→34→17→52→26→13→40→"); //テキストを表示する
    layout.addView(text10); //テキストビューをレイ
アウトに貼り付ける

    text11 = new TextView(this); //テキストビュー型とし
て実体化する
    text11.setTextColor(Color.BLACK); //テキストを白色に指定する
    text11.setTextSize(13f); //テキストの大きさを15
fに指定する
    text11.setText(" 20→10→5→16→8→4→2→1"); //テキストを表示する
    layout.addView(text11); //テキストビューをレイ
アウトに貼り付ける

    text15 = new TextView(this); //テキストビュー型とし
て実体化する
    text15.setTextColor(Color.BLACK); //テキストを白色に指定する
    text15.setTextSize(13f); //テキストの大きさを15
fに指定する
    text15.setText(""); //テキストを表示する

```

```

        layout.addView(text15); //テキストビューをレイ
アウトに貼り付ける

        edit1 = new EditText(this); //エディットテキスト型として実
体化する
        layout.addView(edit1); //エディットテキストを
レイアウトに貼り付ける

        button1 = new Button(this); //ボタン型として実体化
する
        button1.setTextSize(15f); //テキストの
大きさを15fに指定する
        button1.setText("計算"); //テキストを
表示する
        layout.addView(button1); //ボタンをレ
イアウトに貼り付ける
        button1.setOnClickListener(new MyBtnClickAdapter()); //「計算」ボタンにイベント処理リスナーを設定す
る

        button2 = new Button(this); //ボタン型として実体化
する
        button2.setTextSize(15f); //テキストの
大きさを15fに指定する
        button2.setText("初期化"); //テキストを
表示する
        layout.addView(button2); //ボタンをレ
イアウトに貼り付ける
        button2.setOnClickListener(new MyBtnClickAdapter2()); //「初期化」ボタンにイベント処理リスナーを設定
する

        text12 = new TextView(this); //テキストビュー型とし
て実体化する
        text12.setTextColor(Color.BLACK); //テキストを白色に指定する
        text12.setTextSize(13f); //テキストの大きさを15
fに指定する
        text12.setText(""); //テキストを表示する
        layout.addView(text12); //テキストビューをレイ
アウトに貼り付ける

        text13 = new TextView(this); //テキストビュー型とし
て実体化する
        text13.setTextColor(Color.BLACK); //テキストを白色に指定する
        text13.setTextSize(13f); //テキストの大きさを15
fに指定する
        text13.setText(" Copyright(C) K.Niwa 2014.08.10"); //テキストを表示する
        layout.addView(text13); //テキストビューをレイ
アウトに貼り付ける

        setContentView(layout); //レイアウトを画面に貼
り付ける
    } //public void onCreate(Bundle savedInstanceState)

    class MyBtnClickAdapter implements OnClickListener { //「計算」ボタンのイベント処理リスナー
        public void onClick(View view) { //「計算」ボタンをタッチしたとき

            flag++;
//初期化識別子に1を加える

            Editable ed = edit1.getText(); //edをEditable型として宣言し、edにエデ
ィットテキストに入力した文字を入力する
            try {
                n=Integer.parseInt(ed.toString()); //edを文字列型に変換し、さらに
整数型に変換して、nに代入する

            } catch (NumberFormatException e) { //nが整数以外だったら、nを0
にする

```

```

        n=0;
    }

    if (flag==1) { //初期状態だったら、resにnを代入する
        res=n;
    }

    if (res%2==0) { //resが偶数だったら、2で割る
        res=res/2;
    }
    else if (res%2==1 && res!=1) { //resが奇数で1でなかったら、3倍して1を加える
        res=3*res+1;
    }

    text15.setTextSize(20f); //テキストの大きさを20fにする
    text15.setTextColor(Color.RED); //テキストの色を赤色にする
    text15.setText(" 計算結果 : "+res); //計算結果を表示する

    //Toast toast = Toast.makeText(getApplicationContext(),"計算結果 : "+res+"です。
    ", Toast.LENGTH_LONG);
    //toast.show();

} //public void onClick(View view)
} //class MyBtnClickAdapter implements OnClickListener

class MyBtnClickAdapter2 implements OnClickListener { //「初期化」ボタンのイベント処理リスナ
    public void onClick(View view) { //「初期化」ボタンをタッチしたとき
        flag=0;
        //初期化識別子を0にする
        text15.setTextSize(15f); //テキストの
        大きさを15fにする //大きさを15fにする
        text15.setTextColor(Color.WHITE); //テキストの色を白色にする
        text15.setText(""); //テキストを表示する
    } //public void onClick(View view)
} //class MyBtnClickAdapter2 implements OnClickListener

} //public class MyAndroidAppActivity extends Activity

```

[2] 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="@string/hello_world" />
</LinearLayout>

```