1 **. Calculator (Working, APK done)**

**activity\_main.xml:-**

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout 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"

android:orientation="vertical">

<EditText

android:id = "@+id/input1"

android:layout\_width="match\_parent"

android:layout\_height="58dp"

android:hint="Input 1"

android:textSize="24dp"

android:layout\_marginTop="12dp"/>

<EditText

android:id = "@+id/input2"

android:layout\_width="match\_parent"

android:layout\_height="58dp"

android:hint="Input 2"

android:textSize="24dp"

android:layout\_marginTop="12dp"/>

<Button

android:id = "@+id/add"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="+"

android:textSize="24dp"/>

<Button

android:id = "@+id/sub"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="-"

android:textSize="24dp"/>

<Button

android:id = "@+id/mult"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="\*"

android:textSize="24dp"/>

<Button

android:id = "@+id/div"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="/"

android:textSize="24dp"/>

<TextView

android:id = "@+id/output"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:hint="Output"

android:textSize="24dp"

android:layout\_marginTop="12dp"/>

</LinearLayout>

**mainactivity.java:-**

package com.example.calculator;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;

import android.view.View;

import android.widget.Button;

import android.widget.EditText;

import android.widget.TextView;

public class MainActivity extends AppCompatActivity {

EditText ip1,ip2;

TextView op;

Button add,sub,mult,div;

int input1,input2,output;

String ip11,ip22,op11;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

ip1 = (EditText)findViewById(R.id.input1);

ip2 = (EditText)findViewById(R.id.input2);

op = (TextView) findViewById(R.id.output);

add = (Button)findViewById(R.id.add);

sub = (Button)findViewById(R.id.sub);

mult = (Button)findViewById(R.id.mult);

div = (Button)findViewById(R.id.div);

add.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

ip11 = ip1.getText().toString();

ip22 = ip2.getText().toString();

input1 = Integer.valueOf(ip11);

input2 = Integer.valueOf(ip22);

output = input1+input2;

op11 = Integer.toString(output);

op.setText(op11);

}

});

sub.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

ip11 = ip1.getText().toString();

ip22 = ip2.getText().toString();

input1 = Integer.valueOf(ip11);

input2 = Integer.valueOf(ip22);

output = input1-input2;

op11 = Integer.toString(output);

op.setText(op11);

}

});

mult.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

ip11 = ip1.getText().toString();

ip22 = ip2.getText().toString();

input1 = Integer.valueOf(ip11);

input2 = Integer.valueOf(ip22);

output = input1\*input2;

op11 = Integer.toString(output);

op.setText(op11);

}

});

div.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

ip11 = ip1.getText().toString();

ip22 = ip2.getText().toString();

input1 = Integer.valueOf(ip11);

input2 = Integer.valueOf(ip22);

output = input1/input2;

op11 = Integer.toString(output);

op.setText(op11);

}

});

}

}

**2. Traffic Light (Working, Apk done)**

activity\_main.xml:-

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout 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"

android:orientation="vertical">

<RadioGroup

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:orientation="vertical"

android:id = "@+id/colour">

<RadioButton

android:id="@+id/red"

android:layout\_width="162dp"

android:layout\_height="64dp"

android:text="Red Light"

android:textSize="20dp" />

<RadioButton

android:id="@+id/yellow"

android:layout\_width="162dp"

android:layout\_height="64dp"

android:text="Yellow Light"

android:textSize="20dp" />

<RadioButton

android:id="@+id/green"

android:layout\_width="162dp"

android:layout\_height="64dp"

android:text="Green Light"

android:textSize="20dp"

/>

</RadioGroup>

<Button

android:layout\_width="162dp"

android:layout\_height="64dp"

android:id = "@+id/display"

android:textSize="24dp"

android:text="Display"/>

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="64dp"

android:textSize="24dp"

android:id = "@+id/text"

android:hint="Output"/>

</LinearLayout>

**mainactivity.java:-**

package com.example.trafficlight;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;

import android.view.View;

import android.widget.Button;

import android.widget.RadioButton;

import android.widget.RadioGroup;

import android.widget.TextView;

public class MainActivity extends AppCompatActivity {

RadioButton red, green, yellow, radiobutton;

TextView output;

String disp;

Button display;

RadioGroup colour;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

red = (RadioButton)findViewById(R.id.red);

yellow = (RadioButton)findViewById(R.id.yellow);

green = (RadioButton)findViewById(R.id.green);

output = (TextView)findViewById(R.id.text);

colour = (RadioGroup)findViewById(R.id.colour);

display = (Button)findViewById(R.id.display);

display.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

int selectId = colour.getCheckedRadioButtonId();

radiobutton = (RadioButton) findViewById(selectId);

if (radiobutton == red) {

disp = "STOP!";

output.setText(disp);

} else if (radiobutton == yellow) {

disp = "READY!";

output.setText(disp);

} else if (radiobutton == green) {

disp = "GO!";

output.setText(disp);

} else {

disp = "Nothing Selected!";

output.setText(disp);

}

}

});

}

}

**3. Primitives / Graphical (Working, APK Done)**

**Activity\_main.xml**

<?xml version="1.0" encoding="utf-8"?>

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

<ImageView

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:id="@+id/imageView"

android:contentDescription="TODO" />

</RelativeLayout>

**MainActivity.java**

package com.example.primitives;

import android.app.Activity;

import android.graphics.Bitmap;

import android.graphics.Canvas;

import android.graphics.Color;

import android.graphics.Paint;

import android.graphics.drawable.BitmapDrawable;

import android.os.Bundle;

import android.widget.ImageView;

public class MainActivity extends Activity

{

@Override

public void onCreate(Bundle savedInstanceState)

{

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

//Creating a Bitmap

Bitmap bg = Bitmap.createBitmap(720, 1280, Bitmap.Config.ARGB\_8888);

//Setting the Bitmap as background for the ImageView

ImageView i = (ImageView) findViewById(R.id.imageView);

i.setBackgroundDrawable(new BitmapDrawable(bg));

//Creating the Canvas Object

Canvas canvas = new Canvas(bg);

//Creating the Paint Object and set its color & TextSize

Paint paint = new Paint();

paint.setColor(Color.CYAN);

paint.setTextSize(50);

//To draw a Rectangle

canvas.drawText("Rectangle", 420, 150, paint);

canvas.drawRect(400, 200, 650, 700, paint);

//To draw a Circle

canvas.drawText("Circle", 120, 150, paint);

canvas.drawCircle(200, 350, 150, paint);

//To draw a Square

canvas.drawText("Square", 120, 800, paint);

canvas.drawRect(50, 850, 350, 1150, paint);

//To draw a Line

canvas.drawText("Line", 480, 800, paint);

canvas.drawLine(520, 850, 520, 1150, paint);

}

}

**4. GPS (Working Android Studio Project running directly , APK working) (GPSNew.zip)**

**5. Alarm (Working Android Studio Project running directly , APK working) (Alarm.zip)**

**6. SD Card (sdcard2.zip and sdcard.apk) (Not tested)**

**7. Notification (android-notification.apk and android-notification.zip) => for android 8 and android 9**

**Python Programs**

**CELLULAR REUSE:-**

Input - Enter i and j

i=int(input("Enter i: "))

j=int(input("Enter j: "))

print("Cluster Size:", i\*\*2+i\*j+j\*\*2)

mesh=[['.' for z in range(10\*j)]for k in range(10\*i)]

mesh[5+i][5+j]='o'

mesh[5+i-j][5+j+i+j]='z'

mesh[5+i+j][5+j-i-j]='z'

mesh[5+i-i-j][5+j-j]='y'

mesh[5+i+i+j][5+j+j]='y'

mesh[5+i-i-j][5+j+i]='x'

mesh[5+i+i+j][5+j-i]='x'

mesh[5+i-i][5+j-i-j]='x'

mesh[5+i+i][5+j+i+j]='x'

for m in mesh:

print(m)

**CDMA.PY**

Input - Data bits for 4 channels and station no

def mult(c,d):

return list(map(lambda x : x \* d, c))

c1=[1,1,1,1]

c2=[1,-1,1,-1]

c3=[1,1,-1,-1]

c4=[1,-1,-1,1]

C=[c1,c2,c3,c4]

d=[int(x) for x in input("Enter data bits for 4 channels:").split()]

result=[]

for i in range(4):

result.append(mult(C[i],d[i]))

print(result)

channel=[]

for i in range(4):

res=0

for j in range(4):

res+=result[j][i]

channel.append(res)

station=int(input("Enter station you want to listen:"))

res2=0

for i in range(4):

res2+=channel[i]\*C[station-1][i]

print("Data bit transmitted:",res2//4)

**A3.PY**

import random

print("A3 algo")

m = random.getrandbits(128)

print("RAND number provided is: ", m)

c,d = input("Enter key Ki present in SIM: ").split()

# Any operation can be chosen below. Addition/Subtraction/Combination of them

n = int(c)\*\*int(d)

ans = m + n

y = 3 \*\* 100

z = m + y

if (z==ans):

print("Generated SRES has matched. User is authenticated.")

else:

print("Generated SRES does not match. Please retry.")

------------------------------------OR-------------------------------------------

import random

print("A3 algo")

m = random.getrandbits(128)

print("RAND number provided is: ", m)

c,d = input("Enter key Ki present in SIM: ").split()

# Any operation can be chosen below. Addition/Subtraction/Combination of them

n = int(c)\*\*int(d)

ans = m + n

y = 3 \*\* 100

z = m + y

if (z==ans):

print("Generated SRES has matched. User is authenticated.")

else:

print("Generated SRES does not match. Please retry.")

------------------Method -2

table0=[197, 235, 60, 151, 98, 96, 3, 100, 248, 118, 42, 117, 172, 211, 181, 203, 61,

126, 156, 87, 149, 224, 55, 132, 186, 63, 238, 255, 85, 83, 152, 33, 160,

184, 210, 219, 159, 11, 180, 194, 130, 212, 147, 5, 215, 92, 27, 46, 113,

187, 52, 25, 185, 79, 221, 48, 70, 31, 101, 15, 195, 201, 50, 222, 137,

233, 229, 106, 122, 183, 178, 177, 144, 207, 234, 182, 37, 254, 227, 231, 54,

209, 133, 65, 202, 69, 237, 220, 189, 146, 120, 68, 21, 125, 38, 30, 2,

155, 53, 196, 174, 176, 51, 246, 167, 76, 110, 20, 82, 121, 103, 112, 56,

173, 49, 217, 252, 0, 114, 228, 123, 12, 93, 161, 253, 232, 240, 175, 67,

128, 22, 158, 89, 18, 77, 109, 190, 17, 62, 4, 153, 163, 59, 145, 138,

7, 74, 205, 10, 162, 80, 45, 104, 111, 150, 214, 154, 28, 191, 169, 213,

88, 193, 198, 200, 245, 39, 164, 124, 84, 78, 1, 188, 170, 23, 86, 226,

141, 32, 6, 131, 127, 199, 40, 135, 16, 57, 71, 91, 225, 168, 242, 206,

97, 166, 44, 14, 90, 236, 239, 230, 244, 223, 108, 102, 119, 148, 251, 29,

216, 8, 9, 249, 208, 24, 105, 94, 34, 64, 95, 115, 72, 134, 204, 43,

247, 243, 218, 47, 58, 73, 107, 241, 179, 116, 66, 36, 143, 81, 250, 139,

19, 13, 142, 140, 129, 192, 99, 171, 157, 136, 41, 75, 35, 165, 26 ]

table1=[170, 42, 95, 141, 109, 30, 71, 89, 26, 147, 231, 205, 239, 212, 124, 129, 216,

79, 15, 185, 153, 14, 251, 162, 0, 241, 172, 197, 43, 10, 194, 235, 6,

20, 72, 45, 143, 104, 161, 119, 41, 136, 38, 189, 135, 25, 93, 18, 224,

171, 252, 195, 63, 19, 58, 165, 23, 55, 133, 254, 214, 144, 220, 178, 156,

52, 110, 225, 97, 183, 140, 39, 53, 88, 219, 167, 16, 198, 62, 222, 76,

139, 175, 94, 51, 134, 115, 22, 67, 1, 249, 217, 3, 5, 232, 138, 31,

56, 116, 163, 70, 128, 234, 132, 229, 184, 244, 13, 34, 73, 233, 154, 179,

131, 215, 236, 142, 223, 27, 57, 246, 108, 211, 8, 253, 85, 66, 245, 193,

78, 190, 4, 17, 7, 150, 127, 152, 213, 37, 186, 2, 243, 46, 169, 68,

101, 60, 174, 208, 158, 176, 69, 238, 191, 90, 83, 166, 125, 77, 59, 21,

92, 49, 151, 168, 99, 9, 50, 146, 113, 117, 228, 65, 230, 40, 82, 54,

237, 227, 102, 28, 36, 107, 24, 44, 126, 206, 201, 61, 114, 164, 207, 181,

29, 91, 64, 221, 255, 48, 155, 192, 111, 180, 210, 182, 247, 203, 148, 209,

98, 173, 11, 75, 123, 250, 118, 32, 47, 240, 202, 74, 177, 100, 80, 196,

33, 248, 86, 157, 137, 120, 130, 84, 204, 122, 81, 242, 188, 200, 149, 226,

218, 160, 187, 106, 35, 87, 105, 96, 145, 199, 159, 12, 121, 103, 112]

def comp128v23\_internal(KXOR,RAND):

"""Internal part of the COMP128v23 algo, should not be called manually"""

temp = [0] \* 16

KM\_RM = RAND + KXOR

for i in range(5):

for z in range(16):

temp[z] = table0[table1[KM\_RM[16+z]] ^ KM\_RM[z] ]

j = 0

while ( (1 << i) > j):

k = 0

while ( (1 << (4 - i)) > k ):

KM\_RM[((2 \* k + 1) << i )+j] = table0[table1[temp[(k << i) + j]] ^ (KM\_RM[(k << i) + 16 + j])]

KM\_RM[ (k << (i + 1)) + j] = temp[(k << i) + j]

k = k+1

j = j + 1

output = [0]\*16

for i in range(16):

for j in range(8):

output[i] = output[i] ^ (((KM\_RM[(19 \* (j + 8 \* i) + 19) % 256 / 8] >> (3 \* j + 3) % 8) & 1) << j)

return output

def comp128v23(K, RAND, version = 2):

"""The entry point for COMP128v2 and COMP128v3 algorithm

K = The secret Ki number (that should be inside of your SIM card) - Format: list of integers

RAND = The random number generated by the tower - Format: list of integers

version = Version selecting integer (can be 2 or 3) - Format: integer

"""

assert version in [2,3] , "This function only support COMP128 version 2 and 3!"

assert len(K) == 16 , "Ki incorrect (length must be 16)"

assert len(RAND) == 16 , "RAND incorrect (length must be 16)"

K\_MIX = [0]\*16

RAND\_MIX = [0]\*16

KATYVASZ = [0]\*16

output = [0]\*16

for i in range(8):

K\_MIX[i] = K[15 - i]

K\_MIX[15 - i] = K[i]

for i in range(8):

RAND\_MIX[i] = RAND[15 - i]

RAND\_MIX[15 - i] = RAND[i]

for i in range(16):

KATYVASZ[i] = K\_MIX[i] ^ RAND\_MIX[i]

for i in range(8):

RAND\_MIX = comp128v23\_internal(KATYVASZ,RAND\_MIX)

for i in range(16):

output[i] = RAND\_MIX[15-i]

if version == 2:

output[15] = 0

output[14] = 4 \* (output[14] >> 2)

s = 8

i = 0

while i < 4:

output[s+i-4] = output[s+i]

output[s+i] = output[s+i+4]

i = i+1

#the algorithm uses 16 bytes until this point, but only 12 bytes are effective

#also 12 bytes coming out from the SIM card

output\_final = output[:12]

return output\_final

def hex2intarr(input):

"""converts hex string to an array of integers

"""

return map(lambda a: int(a.encode('hex'),16), (a for a in input.decode('hex')))

def intarr2hex(input):

"""converts array of integers to hex strings

"""

return ''.join('{:02x}'.format(x) for x in input).upper()

if \_\_name\_\_ == '\_\_main\_\_':

import argparse

parser = argparse.ArgumentParser(description='Process some integers.')

parser.add\_argument('Ki', metavar='Ki', default = "AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA" ,nargs='?', help='The super secret Ki key')

parser.add\_argument('RAND', metavar='RAND', default = "6E6989BE6CEE7154543770AE80B1EF0D", nargs='?', help='The RANDom number you recieve from the tower')

parser.add\_argument('version', metavar='version', default = 2, nargs='?', help='The version of the COMP128 algo you wish to use (options: 2 or 3)')

args = parser.parse\_args()

Ki = hex2intarr(args.Ki)

RAND = hex2intarr(args.RAND)

version = args.version

print ('----------- INPUT -------------')

print ('COMP128 version ' + str(version))

print ('Ki: ' + intarr2hex(Ki))

print ('RAND: ' + intarr2hex(RAND))

OUTPUT = comp128v23(Ki, RAND, version)

SRES = OUTPUT[:4]

Kc = OUTPUT[4:]

print ('----------- OUTPUT -------------')

print ("SIM OUTPUT:" + intarr2hex(OUTPUT))

print ("SRES: " + intarr2hex(SRES))

print ("Kc: " + intarr2hex(Kc))