ST. ANNE'S
COLLEGE OF ENGINEERING AND TECHNOLOGY
ANGUCHETTYPALAYAM, PANRUTI – 607 110

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

LAB OBSERVATION NOTE

NAME   :   --

REGISTER NO.   :   --

YEAR/ SEM   :   **III / VI**

SUBJECT   :   **CS8662, MOBILE APPLICATION DEVELOPMENT LABORATORY**

DURATION   :   **MAR 2022 –JUN2022**

CS8662 MOBILE APPLICATION DEVELOPMENT LABORATORY

LIST OF EXPERIMENTS:

1. Develop an application that uses GUI components, Font and Colours

2. Develop an application that uses Layout Managers and event listeners.

3. Write an application that draws basic graphical primitives on the screen.

4. Develop an application that makes use of databases.

5. Develop an application that makes use of Notification Manager.

6. Implement an application that uses Multi-threading.

7. Develop a native application that uses GPS location information

8. Implement an application that writes data to the SD card.

9. Implement an application that creates an alert upon receiving a message

10. Write a mobile application that makes use of RSS feed

11. Develop a mobile application to send an email.

12. Develop a Mobile application for simple needs (Mini Project)

TABLE OF CONTENT

S.NO	DATE	NAME OF THE EXPRIEMENT	MARKS	SIGNATURE
1		Develop an application that uses GUI components, Font and Colors.		
2		Develop an application that uses Layout Managers and event listeners.		
3		Write an application that draws basic graphical primitives on the screen.		
4		Develop an application that makes use of databases.		
5		Develop an application that makes use of Notification Manager.		
6		Implement an application that uses Multi-threading.		
7		Develop a native application that uses GPS location information.		
8		Implement an application that writes data to the SD card.		
9		Implement an application that creates an alert upon receiving a message		
10		Write a mobile application that makes use of RSS feed		
11		Develop a mobile application to send an email.		
12		Develop a Mobile application for simple needs (Mini Project).		

X. NO. : 1	**Develop an application that uses GUI components, Font and Colors**
DATE :	

AIM:

To develop an android application that uses GUI Components, Font and colors.

ALGORITHM:

1. Create a New Android Project:
   - Click New in the toolbar.
   - In the window that appears, open the Android folder, select Android Application Project, and click next.
   - Provide the application name and the project name and then finally give the desired package name.
   - Choose a launcher icon for your application and then select Blank Activity and then click Next
   - Provide the desired Activity name for your project and then click Finish.
2. Create a New AVD (Android Virtual Device):
   - click Android Virtual Device Manager from the toolbar.
   - In the Android Virtual Device Manager panel, click New.
   - Fill in the details for the AVD. Give it a name, a platform target, an SD card size, and a skin (HVGA is default).
   - Click Create AVD and Select the new AVD from the Android Virtual Device Manager and click Start.
3. Design the graphical layout with a text view and two command buttons.
4. Run the application.
5. On pressing the change color button, color of the text gets changed.
6. On pressing the change font size button, the size of the font gets altered.
7. Close the Android project.

PROGRAM CODE
MainActivity.java

```
packagecom.example.gui;
importandroid.os.Bundle;
importandroid.app.Activity;
importandroid.graphics.Typeface;
importandroid.graphics.Color;
importandroid.view.View;
importandroid.widget.Button;
importandroid.widget.TextView;


publicclassMainActivityextends Activity {
float font = 24;
inti = 1;
```

```java
    protectedvoidonCreate(Bundle savedInstanceState) {
            super.onCreate(savedInstanceState);
            setContentView(R.layout.activity_main);
            finalTextView t1 = (TextView)findViewById(R.id.textView1);
            Button b1 = (Button)findViewById(R.id.button1);
            b1.setOnClickListener(newView.OnClickListener() {

                            publicvoidonClick(View view) {
                            t1.setTextSize(font);
                            font = font+4;
                            if(font==40)
                                    font = 20;
            }
            });
            Button b2 = (Button)findViewById(R.id.button2);
            b2.setOnClickListener(newView.OnClickListener() {

                    publicvoidonClick(View view) {
                            switch(i)
                            {
                            case 1:
                            t1.setTextColor(Color.parseColor("#0000FF"));
                            break;
                            case 2:
                                    t1.setTextColor(Color.parseColor("#00FF00"));
                                    break;
                    case 3:
                                    t1.setTextColor(Color.parseColor("#FF0000"));
                                    break;
                    case 4:
                                    t1.setTextColor(Color.parseColor("#800000"));
                                    break;

                    }
                    i++;
                    if(i==5)
    i = 1;
                }
            });
    }
}
```

activity_main.xml

```xml
<LinearLayoutxmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="fill_parent"
android:layout_height="fill_parent"
android:orientation="vertical"
>

<TextView
android:id="@+id/textView1"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:text="WELCOME"
android:layout_margin="20sp"
android:gravity="center"
android:textSize="20sp"
android:textStyle="bold"
    />

<Button
android:id="@+id/button1"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_margin="20sp"
android:gravity="center"
android:text="Change Font Size" />

<Button
android:id="@+id/button2"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:gravity="center"
android:layout_margin="20sp"
android:text="Change Color" />

</LinearLayout>
```

OUTPUT:
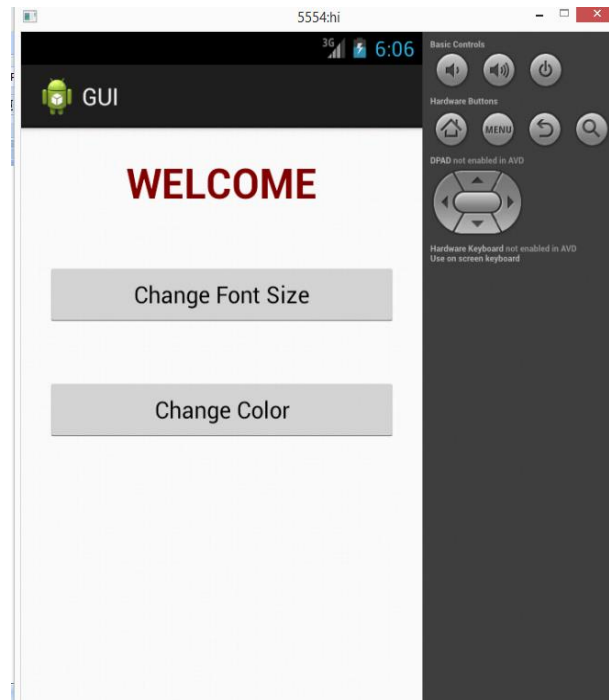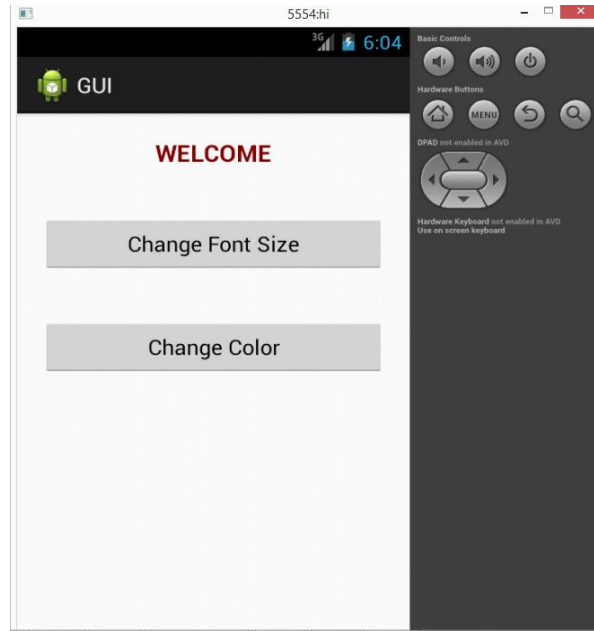
RESULT:

Thus, the program for android application GUI Components, Font and colors was executed successfully.

EX. NO. :2	**Develop an application that uses Layout Managers and Event Listeners**
DATE :	

AIM:

To develop an android application that uses Layout Managers and event listeners.

ALGORITHM:

1. Create a New Android Project:
   - Click New   in the toolbar.
   - In the window that appears, open the Android folder, select Android Application Project, and click next.
   - Provide the application name and the project name and then finally give the desired package name.
   - Choose a launcher icon for your application and then select Blank Activity and then click Next
   - Provide the desired Activity name for your project and then click Finish.
2. Create a New AVD (Android Virtual Device):
   - click Android Virtual Device Manager from the toolbar.
   - In the Android Virtual Device Manager panel, click New.
   - Fill in the details for the AVD. Give it a name, a platform target, an SD card size, and a skin (HVGA is default).
   - Click Create AVD and Select the new AVD from the Android Virtual Device Manager and click Start.
3. Design the graphical layout with buttons, edit text and text view.
4. Run the application.
5. Provide the required inputs to perform the desired arithmetic operation.
6. Display the result.
7. Close the Android project.

PROGRAM CODE:

MainActivity.java

```
packagecom.example.layout;
importandroid.os.Bundle;
importandroid.app.Activity;
importandroid.view.View;
importandroid.view.View.OnClickListener;
importandroid.widget.Button;
importandroid.widget.EditText;
importandroid.widget.Toast;

publicclassMainActivityextends Activity {
        EditText txtData1,txtData2;
        float num1,num2,result1,result2;
```

```java
    protectedvoidonCreate(Bundle savedInstanceState) {
            super.onCreate(savedInstanceState);
            setContentView(R.layout.activity_main);

    Button add = (Button)findViewById(R.id.button1);
    add.setOnClickListener(newOnClickListener(){
    publicvoidonClick(View v){
            try
            {
                    txtData1 = (EditText)findViewById(R.id.editText1);
                    txtData2 = (EditText)findViewById(R.id.editText2);
    num1 = Float.parseFloat(txtData1.getText().toString());
    num2 = Float.parseFloat(txtData2.getText().toString());
    result1 = num1+num2;

    Toast.makeText(getBaseContext(),"ANSWER:"+result1,Toast.LENGTH_SHORT).show();
            }
            catch(Exception e)
            {
            Toast.makeText(getBaseContext(),e.getMessage(),Toast.LENGTH_SHORT).show();

            }
    }
        });
        Button sub = (Button)findViewById(R.id.button3);
        sub.setOnClickListener(newOnClickListener(){

            publicvoidonClick(View v)
            {
                    try
                    {
                            txtData1 = (EditText)findViewById(R.id.editText1);
                            txtData2 = (EditText)findViewById(R.id.editText2);
            num1 = Float.parseFloat(txtData1.getText().toString());
            num2 = Float.parseFloat(txtData2.getText().toString());
            result2 = num1-num2;

Toast.makeText(getBaseContext(),"ANSWER:"+result2,Toast.LENGTH_SHORT).show();
                    }

                    catch(Exception e)
                    {
            Toast.makeText(getBaseContext(),e.getMessage(),Toast.LENGTH_SHORT).show();
                    }
            }

        });
```

```
Button clear = (Button)findViewById(R.id.button2);
        clear.setOnClickListener(newOnClickListener() {
        publicvoidonClick(View v)
                {
                        try
                        {
                                txtData1.setText("");
                                txtData2.setText("");
                        }

                        catch(Exception e)
                        {

        Toast.makeText(getBaseContext(),e.getMessage(),Toast.LENGTH_SHORT).show();
                }
              }
            });
        }
}
```

activity_main.xml

```
<RelativeLayoutxmlns:android="http://schemas.android.com/apk/res/android"
android:id="@+id/relativeLayout1"
android:layout_width="fill_parent"
android:layout_height="fill_parent">

<LinearLayout
android:id="@+id/linearLayout1"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignParentLeft="true"
android:layout_alignParentRight="true"
android:layout_alignParentTop="true">

<TextView
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="ADDITION"
android:layout_gravity="center"
android:textSize="20dp">
</TextView>
</LinearLayout>

<LinearLayout
android:id="@+id/linearLayout2"
```

```
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignParentLeft="true"
android:layout_alignParentRight="true"
android:layout_below="@+id/linearLayout1">
<TextView
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Enter No 1" />
<EditText
android:id="@+id/editText1"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_weight="0.20"
android:inputType="number">
</EditText>
</LinearLayout>

<LinearLayout
android:id="@+id/linearLayout3"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignParentLeft="true"
android:layout_alignParentRight="true"
android:layout_below="@+id/linearLayout2">

<TextView
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Enter No 2" />

<EditText
android:id="@+id/editText2"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_weight="0.20"
android:inputType="number">
</EditText>
</LinearLayout>

<LinearLayout
android:id="@+id/linearLayout4"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignParentLeft="true"
android:layout_alignParentRight="true"
android:layout_below="@+id/linearLayout3">
```

```xml
<Button
android:id="@+id/button1"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_weight="0.50"
android:text="Addition" />

<Button
android:id="@+id/button3"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_weight="0.50"
android:text="Subtraction" />

<Button
android:id="@+id/button2"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_weight="0.50"
android:text="Clear" />
</LinearLayout>

<View
android:id="@+id/linearLayout4"
android:layout_width="fill_parent"
android:layout_height="2px"
android:background="#DDFFDD" />
</RelativeLayout>
```

OUTPUT:
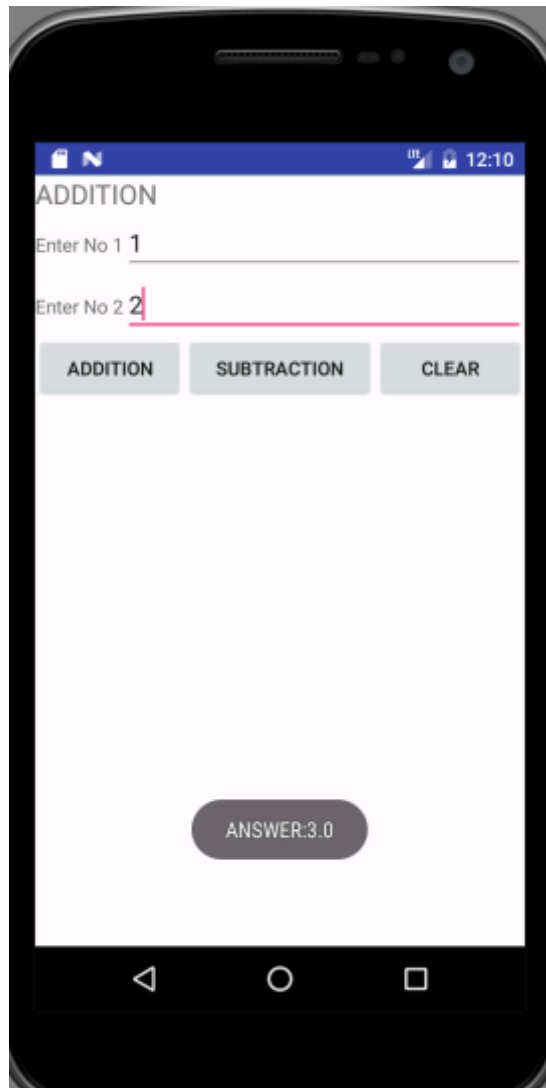
RESULT:

Thus, the program for android application that uses layout managers and event listeners was executed successfully.

EX. NO. :3	Write an application that draws basic graphical primitives on the screen
DATE :	

AIM:

To develop an android application that draws basic graphical primitives on the screen.

ALGORITHM:

1. Create a New Android Project:
     - Click New   in the toolbar.
     - In the window that appears, open the Android folder, select Android Application Project, and click next.
     - Provide the application name and the project name and then finally give the desired package name.
     - Choose a launcher icon for your application and then select Blank Activity and then click Next
     - Provide the desired Activity name for your project and then click Finish.
2. Create a New AVD (Android Virtual Device):
     - click Android Virtual Device Manager from the toolbar.
     - In the Android Virtual Device Manager panel, click New.
     - Fill in the details for the AVD. Give it a name, a platform target, an SD card size, and a skin (HVGA is default).
     - Click Create AVD and Select the new AVD from the Android Virtual Device Manager and click Start.
3. Design the layout by adding a ImageView button.
4. Run the application.
5. Draw straight line by moving the mouse over the screen.
6. Close the Android project.

PROGRAM CODE:

MainActivity.java

```
import android.app.Activity;
import android.graphics.Bitmap;
import android.graphics.Canvas;
import android.graphics.Color;
import android.graphics.Paint;
import android.os.Bundle;
import android.view.Display;
import android.view.MotionEvent;
import android.view.View;
import android.view.View.OnTouchListener;
import android.widget.ImageView;
public class MainActivity extends Activity implements OnTouchListener {
    private ImageViewimageView;
```

```java
        private Bitmap bitmap;
        private Canvas canvas;
        private Paint paint;
        private float downx = 0, downy = 0, upx = 0, upy = 0;
        public void onCreate(Bundle savedInstanceState) {
                super.onCreate(savedInstanceState);
                setContentView(R.layout.activity_main);
                imageView = (ImageView) this.findViewById(R.id.imageView1);
                Display currentDisplay = getWindowManager().getDefaultDisplay();
                float dw = currentDisplay.getWidth();
                float dh = currentDisplay.getHeight();
                bitmap = Bitmap.createBitmap((int) dw, (int) dh, Bitmap.Config.ARGB_8888);
                canvas = new Canvas(bitmap);
                paint = new Paint();
                paint.setColor(Color.BLUE);
                imageView.setImageBitmap(bitmap);
                imageView.setOnTouchListener(this);
        }

        public booleanonTouch(View v, MotionEvent event) {
                int action = event.getAction();
                switch (action) {
                case MotionEvent.ACTION_DOWN:
                        downx = event.getX();
                        downy = event.getY();
                        break;
                case MotionEvent.ACTION_MOVE:
                        break;
                case MotionEvent.ACTION_UP:
                        upx = event.getX();
                        upy = event.getY();
                        canvas.drawLine(downx, downy, upx, upy, paint);
                        imageView.invalidate();
                        break;
                case MotionEvent.ACTION_CANCEL:
                        break;
                default:
                        break;
                }
                return true;
        }
}
```

activity_main.xml

```xml
<RelativeLayoutxmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
```

```xml
android:layout_width="match_parent"
android:layout_height="match_parent">
<ImageView
android:id="@+id/imageView1"
android:layout_width="wrap_content"
android:layout_height="match_parent"
android:layout_alignParentLeft="true"
android:layout_alignParentRight="true"
android:layout_alignParentTop="true"
android:src="@drawable/ic_launcher" />
</RelativeLayout>
```

OUTPUT:
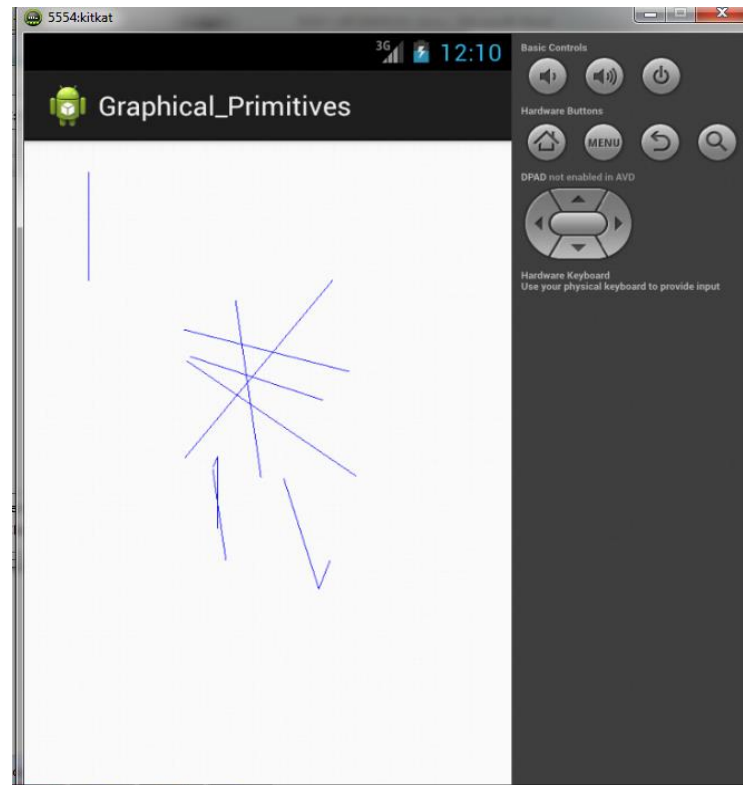
RESULT:

Thus, the program for android application that draws basic graphical primitives on the screen. was executed successfully.

EX. NO:4	
DATE :	**Develop an application that makes use of database**

AIM:

     To develop an android application that makes use of database.

ALGORITHM:

1. Create a New Android Project:
   - Click New in the toolbar.
   - In the window that appears, open the Android folder, select Android Application Project, and click next.
   - Provide the application name and the project name and then finally give the desired package name.
   - Choose a launcher icon for your application and then select Blank Activity and then click Next
   - Provide the desired Activity name for your project and then click Finish.
2. Create a New AVD (Android Virtual Device):
   - click Android Virtual Device Manager from the toolbar.
   - In the Android Virtual Device Manager panel, click New.
   - Fill in the details for the AVD. Give it a name, a platform target, an SD card size, and a skin (HVGA is default).
   - Click Create AVD and Select the new AVD from the Android Virtual Device Manager and click Start.
3. Design the graphical layout.
4. Run the application.
5. Perform the database operation.(Insert/delete/view/update)
6. Close the Android project.

PROGRAM CODE

MainActivity.java
```
package com.example.admin.myapplication;

import android.app.Activity;
import android.app.AlertDialog.Builder;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.EditText;

public class MainActivity extends Activity implements OnClickListener
{
EditText Rollno,Name,Marks;
```

```java
    Button Insert,Delete,Update,View,ViewAll;
SQLiteDatabasedb;
/** Called when the activity is first created. */
@Override
public void onCreate(Bundle savedInstanceState)
    {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);


Rollno=(EditText)findViewById(R.id.Rollno);
Name=(EditText)findViewById(R.id.Name);
Marks=(EditText)findViewById(R.id.Marks);
Insert=(Button)findViewById(R.id.Insert);
Delete=(Button)findViewById(R.id.Delete);
Update=(Button)findViewById(R.id.Update);
View=(Button)findViewById(R.id.View);
ViewAll=(Button)findViewById(R.id.ViewAll);


Insert.setOnClickListener(this);
Delete.setOnClickListener(this);
Update.setOnClickListener(this);
View.setOnClickListener(this);
ViewAll.setOnClickListener(this);


// Creating database and table
db=openOrCreateDatabase("StudentDB", Context.MODE_PRIVATE, null);
db.execSQL("CREATE TABLE IF NOT EXISTS
student(rollnoVARCHAR,nameVARCHAR,marks VARCHAR);");
    }
public void onClick(View view)
    {
// Inserting a record to the Student table
if(view==Insert)
      {
// Checking for empty fields
if(Rollno.getText().toString().trim().length()==0||
Name.getText().toString().trim().length()==0||
Marks.getText().toString().trim().length()==0)
        {
showMessage("Error", "Please enter all values");
return;
        }
db.execSQL("INSERT INTO student VALUES('"+Rollno.getText()+"','"+Name.getText()+
"','"+Marks.getText()+"');");
showMessage("Success", "Record added");
clearText();
      }
// Deleting a record from the Student table
if(view==Delete)
      {
// Checking for empty roll number
if(Rollno.getText().toString().trim().length()==0)
        {
showMessage("Error", "Please enter Rollno");
return;
```

```java
        }
        Cursor c=db.rawQuery("SELECT * FROM student WHERE
rollno='"+Rollno.getText()+"'", null);
if(c.moveToFirst())
        {
db.execSQL("DELETE FROM student WHERE rollno='"+Rollno.getText()+"'");
showMessage("Success", "Record Deleted");
        }
else
{
showMessage("Error", "Invalid Rollno");
        }
clearText();
        }
// Updating a record in the Student table
if(view==Update)
        {
// Checking for empty roll number
if(Rollno.getText().toString().trim().length()==0)
        {
showMessage("Error", "Please enter Rollno");
return;
        }
        Cursor c=db.rawQuery("SELECT * FROM student WHERE
rollno='"+Rollno.getText()+"'", null);
if(c.moveToFirst()) {
db.execSQL("UPDATE student SET name='" + Name.getText() + "',marks='" +
Marks.getText() +
"' WHERE rollno='"+Rollno.getText()+"'");
showMessage("Success", "Record Modified");
        }
else {
showMessage("Error", "Invalid Rollno");
        }
clearText();
        }
// Display a record from the Student table
if(view==View)
        {
// Checking for empty roll number
if(Rollno.getText().toString().trim().length()==0)
        {
showMessage("Error", "Please enter Rollno");
return;
        }
        Cursor c=db.rawQuery("SELECT * FROM student WHERE
rollno='"+Rollno.getText()+"'", null);
if(c.moveToFirst())
        {
Name.setText(c.getString(1));
Marks.setText(c.getString(2));
        }
else
{
showMessage("Error", "Invalid Rollno");
```

```java
clearText();
        }
    }
// Displaying all the records
if(view==ViewAll)
    {
        Cursor c=db.rawQuery("SELECT * FROM student", null);
if(c.getCount()==0)
        {
showMessage("Error", "No records found");
return;
        }
StringBuffer buffer=new StringBuffer();
while(c.moveToNext())
        {
buffer.append("Rollno: "+c.getString(0)+"\n");
buffer.append("Name: "+c.getString(1)+"\n");
buffer.append("Marks: "+c.getString(2)+"\n\n");
        }
showMessage("Student Details", buffer.toString());
    }
  }
public void showMessage(String title,String message)
  {
    Builder builder=new Builder(this);
builder.setCancelable(true);
builder.setTitle(title);
builder.setMessage(message);
builder.show();
  }
public void clearText()
  {
Rollno.setText("");
Name.setText("");
Marks.setText("");
Rollno.requestFocus();
  }
}
```

activity_main.xml

```xml
<?xml version="1.0" encoding="utf-8"?>
<AbsoluteLayoutxmlns:android="http://schemas.android.com/apk/res/android"
android:layout_width="match_parent"
android:layout_height="match_parent">
<TextView
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_x="50dp"
android:layout_y="20dp"
android:text="Student Details"
android:textSize="30sp" />

<TextView
android:layout_width="wrap_content"
```

```xml
        android:layout_height="wrap_content"
        android:layout_x="20dp"
        android:layout_y="110dp"
        android:text="Enter Rollno:"
        android:textSize="20sp" />

    <EditText
        android:id="@+id/Rollno"
        android:layout_width="150dp"
        android:layout_height="wrap_content"
        android:layout_x="175dp"
        android:layout_y="100dp"
        android:inputType="number"
        android:textSize="20sp" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_x="20dp"
        android:layout_y="160dp"
        android:text="Enter Name:"
        android:textSize="20sp" />

    <EditText
        android:id="@+id/Name"
        android:layout_width="150dp"
        android:layout_height="wrap_content"
        android:layout_x="175dp"
        android:layout_y="150dp"
        android:inputType="text"
        android:textSize="20sp" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_x="20dp"
        android:layout_y="210dp"
        android:text="Enter Marks:"
        android:textSize="20sp" />

    <EditText
        android:id="@+id/Marks"
        android:layout_width="150dp"
        android:layout_height="wrap_content"
        android:layout_x="175dp"
        android:layout_y="200dp"
        android:inputType="number"
        android:textSize="20sp" />

    <Button
        android:id="@+id/Insert"
        android:layout_width="150dp"
        android:layout_height="wrap_content"
        android:layout_x="25dp"
        android:layout_y="300dp"
```

```xml
    android:text="Insert"
    android:textSize="30dp" />

<Button
    android:id="@+id/Delete"
    android:layout_width="150dp"
    android:layout_height="wrap_content"
    android:layout_x="200dp"
    android:layout_y="300dp"
    android:text="Delete"
    android:textSize="30dp" />

<Button
    android:id="@+id/Update"
    android:layout_width="150dp"
    android:layout_height="wrap_content"
    android:layout_x="25dp"
    android:layout_y="400dp"
    android:text="Update"
    android:textSize="30dp" />

<Button
    android:id="@+id/View"
    android:layout_width="150dp"
    android:layout_height="wrap_content"
    android:layout_x="200dp"
    android:layout_y="400dp"
    android:text="View"
    android:textSize="30dp" />

<Button
    android:id="@+id/ViewAll"
    android:layout_width="200dp"
    android:layout_height="wrap_content"
    android:layout_x="100dp"
    android:layout_y="500dp"
    android:text="View All"
    android:textSize="30dp" />

</AbsoluteLayout>
```
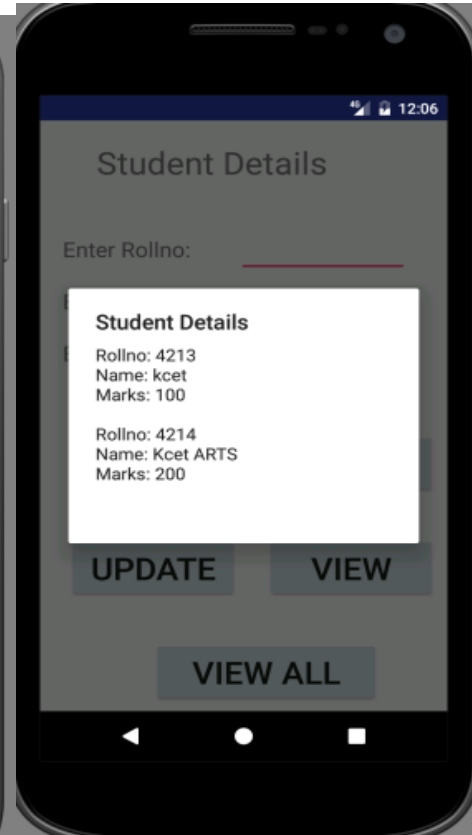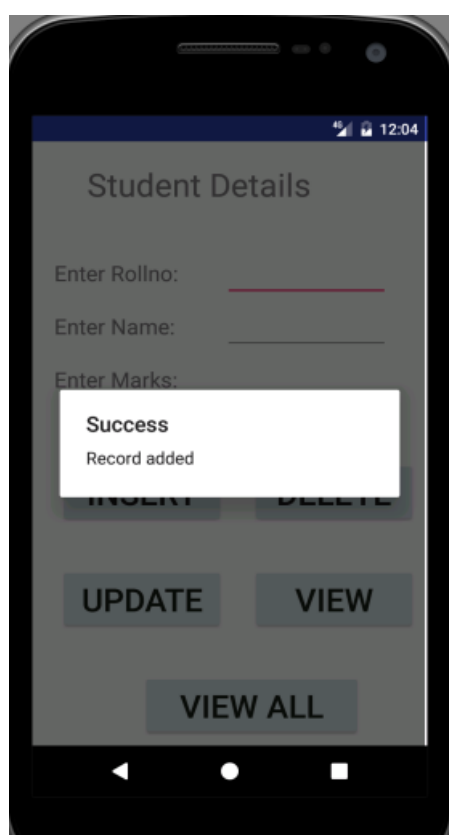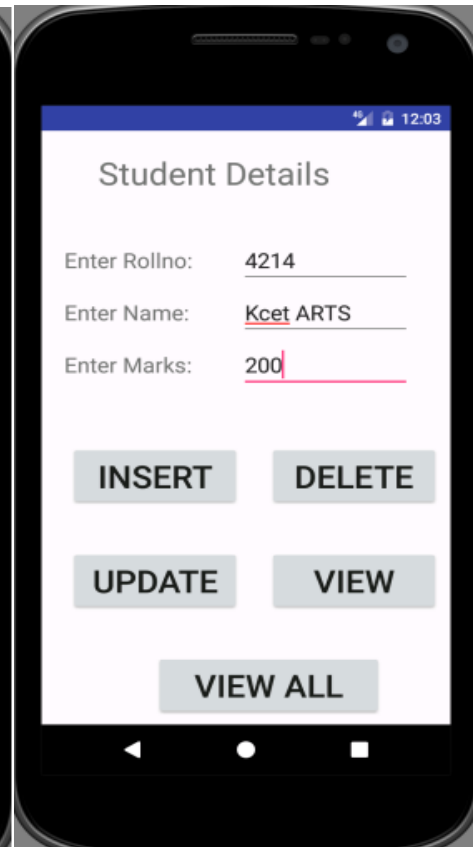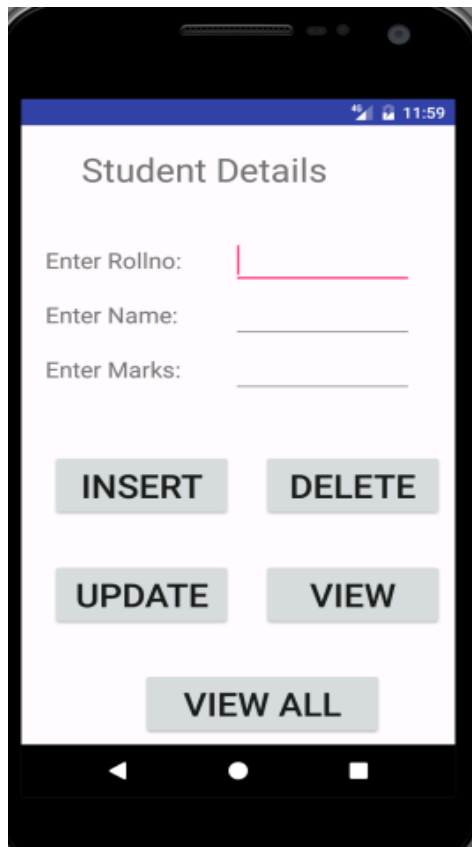
OUTPUT:

RESULT:

Thus, the program for android application that makes use of database was executed successfully.

EX. NO:5	**Develop an application that makes use of Notification Manager**
DATE :	

AIM:

To develop a Android Application that that makes use of Notification Manager.

ALGORITHM:

1. Create a New Android Project:
    *   Click New in the toolbar.
    *   In the window that appears, open the Android folder, select Android Application Project, and click next.
    *   Provide the application name and the project name and then finally give the desired package name.
    *   Choose a launcher icon for your application and then select Blank Activity and then click Next
    *   Provide the desired Activity name for your project and then click Finish.
2. Create a New AVD (Android Virtual Device):
    *   click Android Virtual Device Manager from the toolbar.
    *   In the Android Virtual Device Manager panel, click New.
    *   Fill in the details for the AVD. Give it a name, a platform target, an SD card size, and a skin (HVGA is default).
    *   Click Create AVD and Select the new AVD from the Android Virtual Device Manager and click Start.
3. Design the graphical layout using buttons, text and ImageView.
4. Creating Second Activity for the Android Application:
5. Click on File -> New -> Activity -> Empty Activity.
6. Type the Activity Name as Second Activity and click Finish button.
7. Run the application.
8. Display the output by clicking the Notify button.
9. Close the Android project.

PROGRAM CODE:

MainActivity.java

```
package com.example.admin.myapplication;
import android.app.Notification;
import android.app.NotificationManager;
import android.app.PendingIntent;
import android.content.Intent;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.widget.Button;
```

```java
import android.widget.EditText;

public class MainActivity extends AppCompatActivity
{
    Button notify;
EditText e;
    @Override
protected void onCreate(Bundle savedInstanceState)
    {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);

notify= (Button) findViewById(R.id.button);
e= (EditText) findViewById(R.id.editText);

notify.setOnClickListener(new View.OnClickListener()
        {
        @Override
public void onClick(View v)
            {
            Intent intent = new Intent(MainActivity.this, Main2Activity.class);
PendingIntent pending = PendingIntent.getActivity(MainActivity.this, 0, intent, 0);
Notificationnoti = new Notification.Builder(MainActivity.this).setContentTitle("New
Message").setContentText(e.getText().toString()).setSmallIcon(R.mipmap.ic_launcher).setContentI
ntent(pending).build();
NotificationManager manager = (NotificationManager)
getSystemService(NOTIFICATION_SERVICE);
noti.flags|= Notification.FLAG_AUTO_CANCEL;
manager.notify(0, noti);
        }
    });
    }
}
```

activity_main.xml

```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayoutxmlns:android="http://schemas.android.com/apk/res/android"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:layout_margin="10dp"
android:orientation="vertical">

<TextView
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Message Form KCET"
android:textSize="30sp" />

<EditText
android:id="@+id/editText"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:textSize="30sp" />
```

```xml
<Button
android:id="@+id/button"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_margin="30dp"
android:layout_gravity="center"
android:text="Notify"
android:textSize="30sp"/>

</LinearLayout>
```

Main2Activity.java

```java
package com.example.admin.myapplication;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
public class Main2Activity extends AppCompatActivity {
    @Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main2);
    }
}
```

OUTPUT:
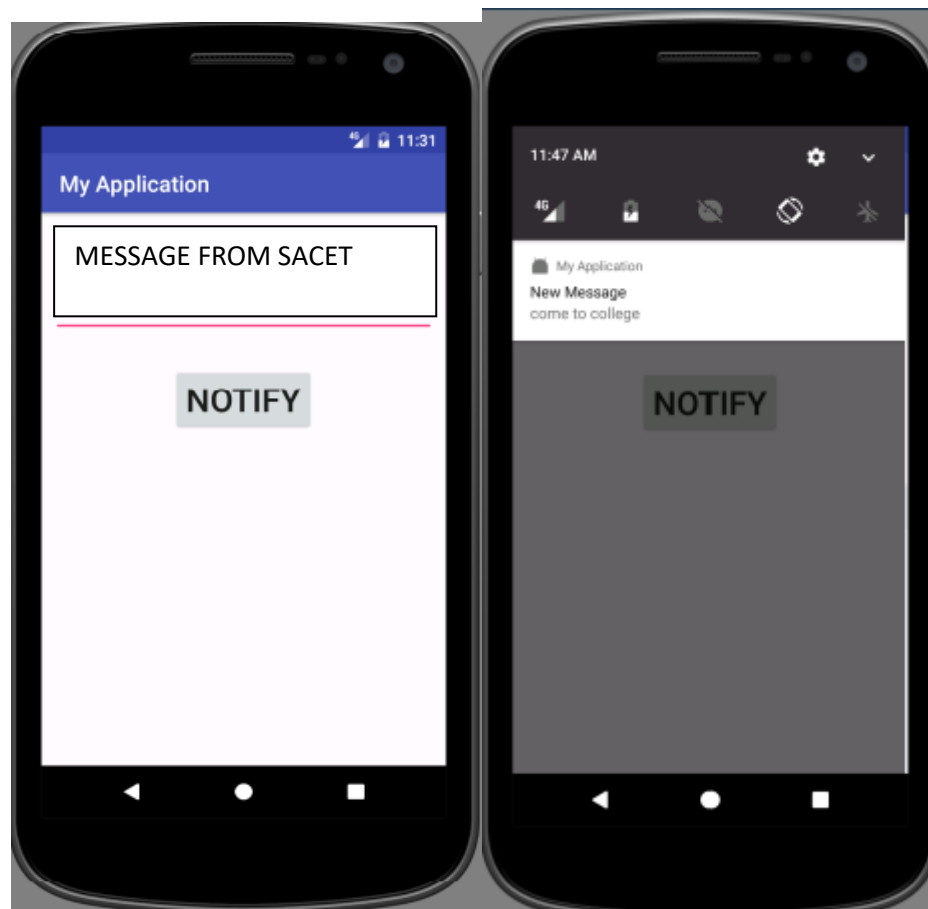
Result:

        Thus Android Application that that makes use of Notification Manager is developed and executedsuccessfully.

EX. NO. :6	Implement an application that implements Multithreading
DATE :	

AIM:

       To develop an android application that implements multithreading.

ALGORITHM:

1. Create a New Android Project:
   - Click New   in the toolbar.
   - In the window that appears, open the Android folder, select Android Application Project, and click next.
   - Provide the application name and the project name and then finally give the desired package name.
   - Choose a launcher icon for your application and then select Blank Activity and then click Next
   - Provide the desired Activity name for your project and then click Finish.
2. Create a New AVD (Android Virtual Device):
   - click Android Virtual Device Manager from the toolbar.
   - In the Android Virtual Device Manager panel, click New.
   - Fill in the details for the AVD. Give it a name, a platform target, an SD card size, and a skin (HVGA is default).
   - Click Create AVD and Select the new AVD from the Android Virtual Device Manager and click Start.
3. Design the graphical layout.
4. Run the application.
5. The requested data is retrieved from the database named myFriendsDb.
6. Close the Android project.

PROGRAM CODE

MainActivity.Java

```
package com.example.admin.multithreading;

import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;
public class MainActivityextends AppCompatActivity
{
ImageViewimg;
   Button bt1,bt2;
   @Override
protected void onCreate(Bundle savedInstanceState)
   {
super.onCreate(savedInstanceState);
```

```java
setContentView(R.layout.activity_main);

bt1 = (Button)findViewById(R.id.button);
bt2= (Button) findViewById(R.id.button2);
img= (ImageView)findViewById(R.id.imageView);

bt1.setOnClickListener(new View.OnClickListener()
    {
        @Override
public void onClick(View v)
        {
new Thread(new Runnable()
        {
            @Override
public void run()
            {
img.post(new Runnable()
                {
                    @Override
public void run()
                {
img.setImageResource(R.drawable.india1);
                }
                });
            }
        }).start();
        }
    });

bt2.setOnClickListener(new View.OnClickListener()
    {
        @Override
public void onClick(View v)
        {
new Thread(new Runnable()
        {
            @Override
public void run()
            {
img.post(new Runnable()
                {
                    @Override
public void run()
                {
img.setImageResource(R.drawable.india2);
                }
                });
            }
        }).start();
        }
    });
    }
}
```

activity_main.xml

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

<ImageView
android:id="@+id/imageView"
android:layout_width="250dp"
android:layout_height="250dp"
android:layout_margin="50dp"
android:layout_gravity="center" />

<Button
android:id="@+id/button"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_margin="10dp"
android:layout_gravity="center"
android:text="Load Image 1" />

<Button
android:id="@+id/button2"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_margin="10dp"
android:layout_gravity="center"
android:text="Load image 2" />

</LinearLayout>
```

OUTPUT:
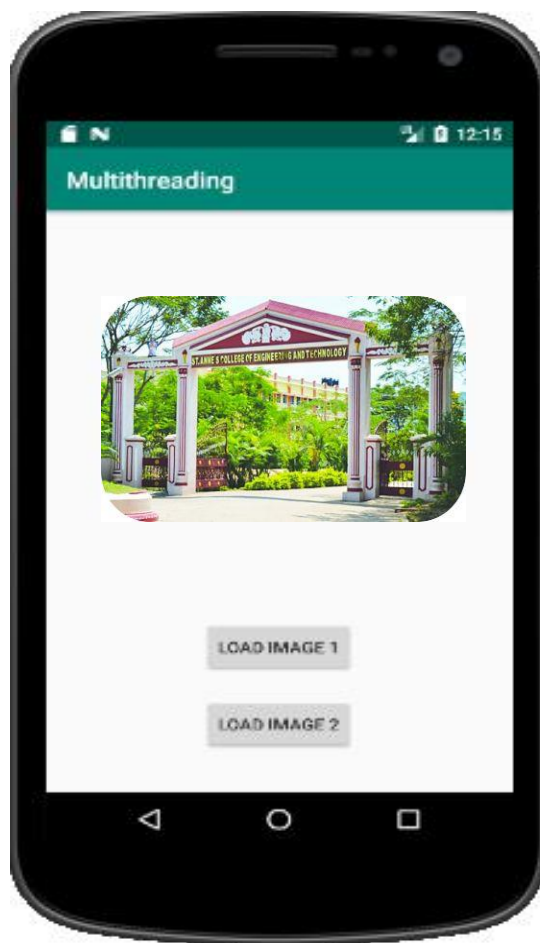
RESULT:

Thus, the program for android application that makes use of multithreading was executed successfully.

EX. NO. :7	**Develop a native application that uses GPS location information**
DATE :	

AIM:

To develop an android application that uses GPS location information.

ALGORITHM:

1. Create a New Android Project:
   - Click New in the toolbar.
   - In the window that appears, open the Android folder, select Android Application Project, and click next.
   - Provide the application name and the project name and then finally give the desired package name.
   - Choose a launcher icon for your application and then select Blank Activity and then click Next
   - Provide the desired Activity name for your project and then click Finish.
2. Create a New AVD (Android Virtual Device):
   - click Android Virtual Device Manager from the toolbar.
   - In the Android Virtual Device Manager panel, click New.
   - Fill in the details for the AVD. Give it a name, a platform target, an SD card size, and a skin (HVGA is default).
   - Click Create AVD and Select the new AVD from the Android Virtual Device Manager and click Start.
3. Design the graphical layout.
4. Run the application.
5. The requested data is retrieved from the database named myFriendsDb.
6. Close the Android project.

PROGRAM CODE

UseGps.java

```
package com.emergency;
import android.app.Activity;
import android.content.Context;
import android.location.Location;
import android.location.LocationListener;
import android.location.LocationManager;
import android.os.Bundle;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
```

```java
public class UseGps extends Activity
{
        Button buttonSend;
        EditTexttextSMS;
        EditTexttextlon;
public void onCreate(Bundle savedInstanceState)
{
super.onCreate(savedInstanceState);
setContentView(R.layout.main);
buttonSend = (Button) findViewById(R.id.buttonSend);
textSMS = (EditText) findViewById(R.id.editTextSMS);
textlon = (EditText) findViewById(R.id.textlon);
LocationManagermlocManager                                                =
(LocationManager)getSystemService(Context.LOCATION_SERVICE);
LocationListenermlocListener = new MyLocationListener();
mlocManager.requestLocationUpdates( LocationManager.GPS_PROVIDER, 0, 0, mlocListener);
}
public class MyLocationListener implements LocationListener
{
public void onLocationChanged(Location loc)
{
loc.getLatitude();
loc.getLongitude();
Double lat=loc.getLatitude();
Double lon=loc.getLongitude();
textSMS.setText(lat.toString());
textlon.setText(lon.toString());
}
public void onProviderDisabled(String provider)
{
Toast.makeText( getApplicationContext(),"Gps Disabled",Toast.LENGTH_SHORT ).show();
}
public void onProviderEnabled(String provider)
{
Toast.makeText( getApplicationContext(), "Gps Enabled", Toast.LENGTH_SHORT).show();
}
}
}
```

main.xml

```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayoutxmlns:android="http://schemas.android.com/apk/res/android"
android:orientation="vertical"
android:layout_width="fill_parent"
android:layout_height="fill_parent"
>
<TextView
android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:text="Emergency Alert System"
    />

<EditText
android:id="@+id/editTextSMS"
android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:gravity="top" />

<EditText
android:id="@+id/textlon"
android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:gravity="top" />

<Button
android:id="@+id/buttonSend"
android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:text="Send" />

</LinearLayout>
```

OUTPUT:
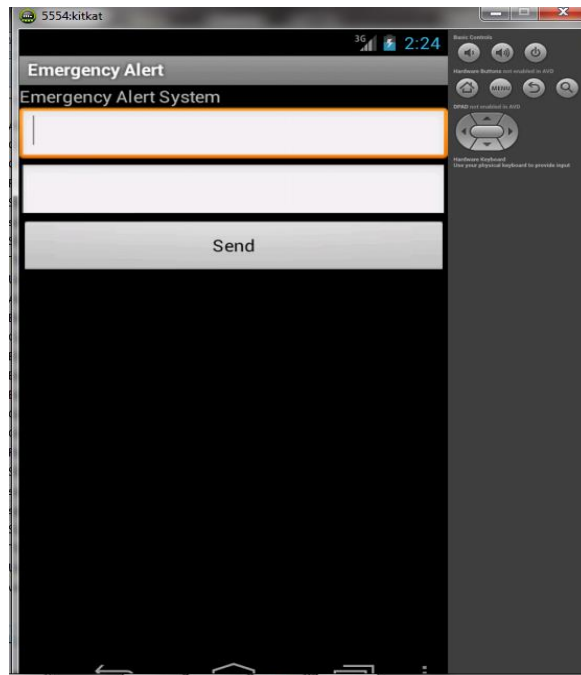
RESULT:

Thus, the program for android application that makes use of GPS information was executed successfully.

EX. NO. :8	**Implement an application that writes data to the SD card**
DATE :	

AIM:

To develop an android application that writes data to the SD card.

ALGORITHM:

1. Create a New Android Project:
   - Click New in the toolbar.
   - In the window that appears, open the Android folder, select Android Application Project, and click next.
   - Provide the application name and the project name and then finally give the desired package name.
   - Choose a launcher icon for your application and then select Blank Activity and then click Next
   - Provide the desired Activity name for your project and then click Finish.
2. Create a New AVD (Android Virtual Device):
   - click Android Virtual Device Manager from the toolbar.
   - In the Android Virtual Device Manager panel, click New.
   - Fill in the details for the AVD. Give it a name, a platform target, an SD card size, and a skin (HVGA is default).
   - Click Create AVD and Select the new AVD from the Android Virtual Device Manager and click Start.
3. Design the graphical layout using buttons, Textview.
4. Run the application.
5. Writes the data into a file in the SD card.
6. Display the output by clicking the View button.
7. Close the Android project.

PROGRAM CODE:

MainActivity.java
```
package com.example.admin.myapplication;

import android.app.Activity;
import android.os.Bundle;
import android.os.Environment;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;

import java.io.BufferedReader;
import java.io.DataInputStream;
import java.io.File;
import java.io.FileInputStream;
```

```java
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStreamReader;


public class MainActivityextends Activity {
EditTextinputText;
TextViewresponse;
   Button saveButton,readButton;

private String filename = "SampleFile.txt";
private String filepath= "MyFileStorage";
   File myExternalFile;
   String myData= "";

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

inputText= (EditText) findViewById(R.id.myInputText);
response = (TextView) findViewById(R.id.response);



saveButton=
         (Button) findViewById(R.id.saveExternalStorage);
saveButton.setOnClickListener(new OnClickListener() {
      @Override
public void onClick(View v) {
try {
FileOutputStreamfos = new FileOutputStream(myExternalFile);
fos.write(inputText.getText().toString().getBytes());
fos.close();
         } catch (IOException e) {
e.printStackTrace();
         }
inputText.setText("");
response.setText("SampleFile.txt saved to External Storage...");
      }
   });

readButton= (Button) findViewById(R.id.getExternalStorage);
readButton.setOnClickListener(new OnClickListener() {
      @Override
public void onClick(View v) {
try {
FileInputStreamfis = new FileInputStream(myExternalFile);
DataInputStream in = new DataInputStream(fis);
BufferedReaderbr =
new BufferedReader(new InputStreamReader(in));
         String strLine;
while ((strLine = br.readLine()) != null) {
myData= myData+ strLine;
         }
in.close();
```

```java
            } catch (IOException e) {
e.printStackTrace();
            }
inputText.setText(myData);
response.setText("SampleFile.txt data retrieved from Internal Storage...");
        }
    });

if (!isExternalStorageAvailable() || isExternalStorageReadOnly()) {
saveButton.setEnabled(false);
    }
else {
myExternalFile= new File(getExternalFilesDir(filepath), filename);
    }


    }
private static booleanisExternalStorageReadOnly() {
    String extStorageState = Environment.getExternalStorageState();
if (Environment.MEDIA_MOUNTED_READ_ONLY.equals(extStorageState)) {
return true;
    }
return false;
  }

private static booleanisExternalStorageAvailable() {
    String extStorageState = Environment.getExternalStorageState();
if (Environment.MEDIA_MOUNTED.equals(extStorageState)) {
return true;
    }
return false;
  }


}
```

activity_main.xml

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

<TextViewandroid:layout_width="fill_parent"
android:layout_height="wrap_content"
android:text="Reading and Writing to External Storage"
android:textSize="24sp"/>

<EditTextandroid:id="@+id/myInputText"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:ems="10" android:lines="5"
android:minLines="3" android:gravity="top|left"
android:inputType="textMultiLine">
```

```xml
<requestFocus/>
</EditText>

<LinearLayout
android:layout_width="match_parent" android:layout_height="wrap_content"
android:orientation="horizontal"
android:weightSum="1.0"
android:layout_marginTop="20dp">

<Button android:id="@+id/saveExternalStorage"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:text="SAVE"
android:layout_weight="0.5"/>

<Button android:id="@+id/getExternalStorage"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_weight="0.5"
android:text="READ" />

</LinearLayout>

<TextViewandroid:id="@+id/response"
android:layout_width="wrap_content"
android:layout_height="wrap_content" android:padding="5dp"
android:text=""
android:textAppearance="?android:attr/textAppearanceMedium" />

</LinearLayout>
```

AndroidMainfest.xml

```xml
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.admin.myapplication">

<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>

<application
android:allowBackup="true"
android:icon="@mipmap/ic_launcher"
android:label="@string/app_name"
android:supportsRtl="true"
android:theme="@style/AppTheme">
<activity android:name=".MainActivity">
<intent-filter>
<action android:name="android.intent.action.MAIN" />

<category android:name="android.intent.category.LAUNCHER" />
</intent-filter>
</activity>
```
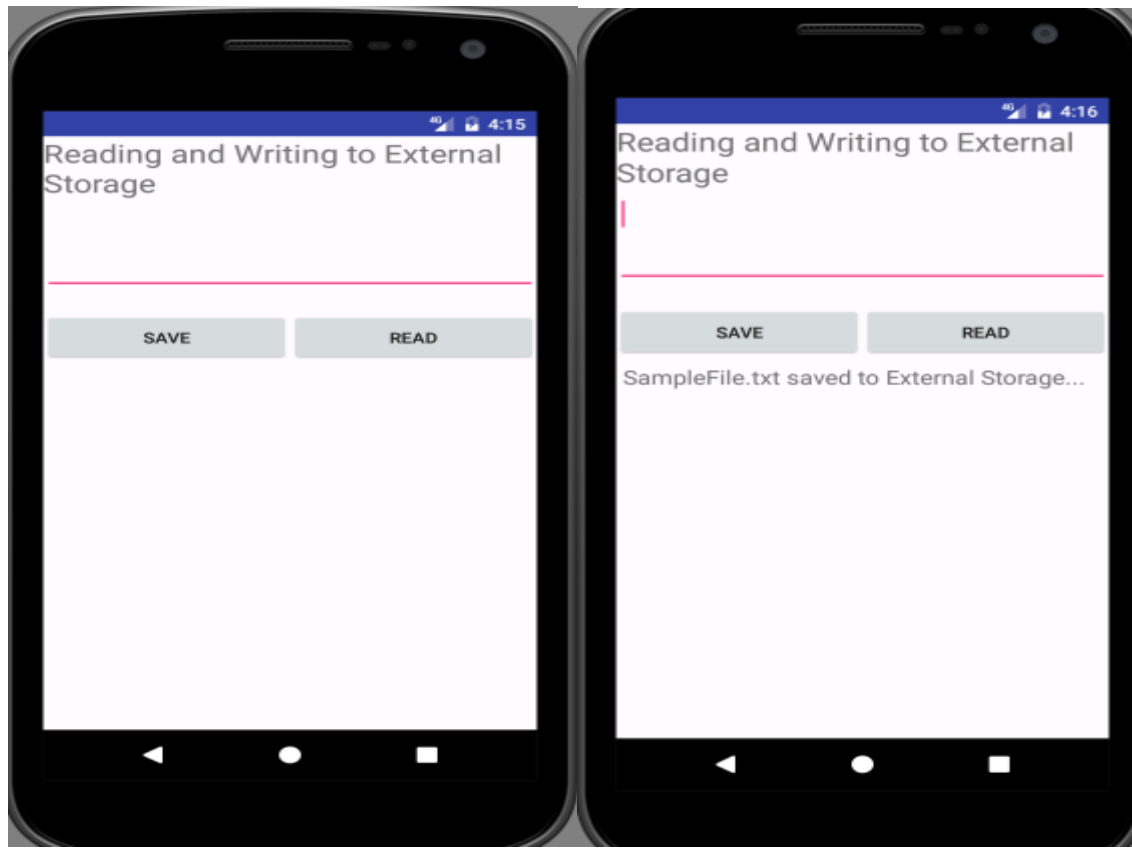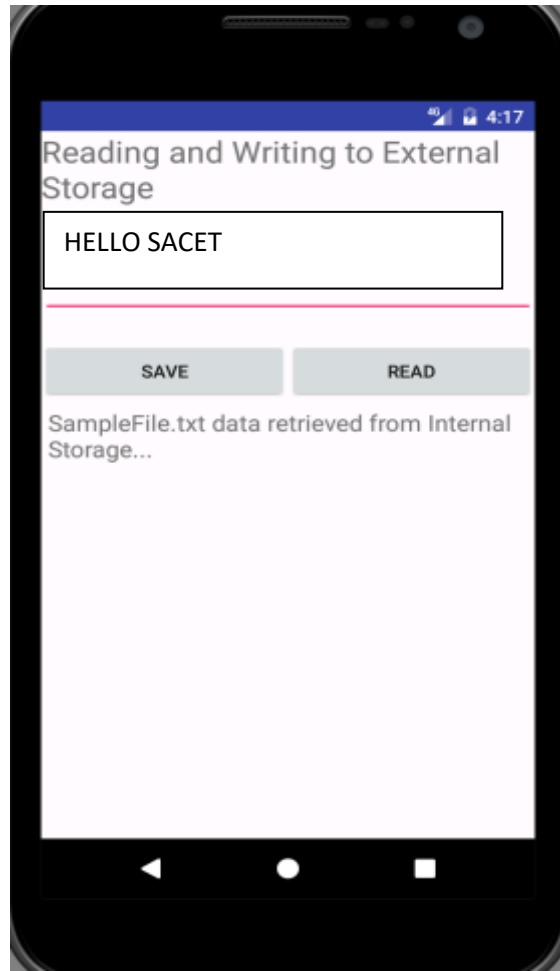
```
</application>
</manifest>
```

OUTPUT:

RESULT:

Thus, the program for android application that writes data into the SD Card was executed successfully.

EX. NO. :9	
	Implement an application that creates an alert upon receiving a message
DATE :	

AIM:

To develop an android application that creates an alert upon receiving a message.

ALGORITHM:

1. Create a New Android Project:
   - Click New in the toolbar.
   - In the window that appears, open the Android folder, select Android Application Project, and click next.
   - Provide the application name and the project name and then finally give the desired package name.
   - Choose a launcher icon for your application and then select Blank Activity and then click Next
   - Provide the desired Activity name for your project and then click Finish.
2. Create a New AVD (Android Virtual Device):
   - click Android Virtual Device Manager from the toolbar.
   - In the Android Virtual Device Manager panel, click New.
   - Fill in the details for the AVD. Give it a name, a platform target, an SD card size, and a skin (HVGA is default).
   - Click Create AVD and Select the new AVD from the Android Virtual Device Manager and click Start.
3. Design the layout by adding a text box and a command button.
4. Run the application.
5. If the entered E-mail doesn't match the given E-mail id, then an alert will be displayed.
6. If the entered E-mail id matches with the provided mail-id then login is successful.
7. Close the Android project.

PROGRAM CODE:

MainActivity.java

```
package com.pa.Alert;
import android.os.Bundle;
import android.app.Activity;
import android.content.Intent;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
```

```java
public class MainActivity extends Activity {
        private Button BTN;
        private EditText email;

        protected void onCreate(Bundle savedInstanceState) {
                super.onCreate(savedInstanceState);
                setContentView(R.layout.activity_main);
                BTN = (Button) findViewById(R.id.btn);
                email = (EditText) findViewById(R.id.emailInput);
                BTN.setOnClickListener(new OnClickListener() {
                    public void onClick(View v) {
                        String val = email.getText().toString();
                            if (val == null || val.length() <= 0) {
                                Toast.makeText(getApplicationContext(),
                                        "Please Enter the email", Toast.LENGTH_LONG).show();
                            } else if (val.equals("enpboss@gmail.com")) {
                                    Intent intent = new Intent(getApplicationContext(),
                                        SecondActivity.class);
                                    startActivity(intent);
                                    Toast.makeText(getApplicationContext(),
                                        "Login Success", Toast.LENGTH_LONG).show();
                            } else {
                                Toast.makeText(getApplicationContext(),
                                "Please Enter valid email", Toast.LENGTH_LONG)
                                                .show();
                            }
                    }
                });
        }
}
```

SecondActivity.java
```java
package com.pa.Alert;

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

public class SecondActivity extends Activity {
        @Override
        protected void onCreate(Bundle savedInstanceState) {
                // TODO Auto-generated method stub
                super.onCreate(savedInstanceState);
                setContentView(R.layout.second_activity);
        }
}
```

Main_activity.xml

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

<EditText
android:id="@+id/emailInput"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:ems="10" />

<Button
android:id="@+id/btn"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_margin="20sp"
android:gravity="center"
android:text="Login" />

</LinearLayout>
```

AndroidMainfest.Xml

```xml
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.admin.myapplication">

<application
android:allowBackup="true"
android:icon="@mipmap/ic_launcher"
android:label="@string/app_name"
android:roundIcon="@mipmap/ic_launcher_round"
android:supportsRtl="true"
android:theme="@style/AppTheme">

<activity android:name=".MainActivity">
<intent-filter>
<action android:name="android.intent.action.MAIN" />

<category android:name="android.intent.category.LAUNCHER" />
</intent-filter>
</activity>
<activity android:name=".SecondActivity">
<intent-filter>
<action android:name="android.intent.action.MAIN" />
<category android:name="android.intent.category.LAUNCHER" />
</intent-filter>
</activity>
</application>
</manifest>
```

OUTPUT:

RESULT:

      Thus, the program for android application that creates an alert upon receiving a message was executed successfully.

EX. NO. :10	Develop an application that makes use of RSS Feed
DATE :	

AIM:

To develop an android application that makes use of RSS (Rich Site Summary) Feed.

ALGORITHM:

10. Create a New Android Project:
    - Click New   in the toolbar.
    - In the window that appears, open the Android folder, select Android Application Project, and click next.
    - Provide the application name and the project name and then finally give the desired package name.
    - Choose a launcher icon for your application and then select Blank Activity and then click Next
    - Provide the desired Activity name for your project and then click Finish.
11. Create a New AVD (Android Virtual Device):
    - click Android Virtual Device Manager from the toolbar.
    - In the Android Virtual Device Manager panel, click New.
    - Fill in the details for the AVD. Give it a name, a platform target, an SD card size, and a skin (HVGA is default).
    - Click Create AVD and Select the new AVD from the Android Virtual Device Manager and click Start.
12. Design the graphical layout using buttons, text and ImageView.
13. Run the application.
14. Fetch the details from the URL.
15. Display the output by clicking the result button.
16. Close the Android project.

PROGRAM CODE:

MainActivity.java

```
package com.example.admin.myapplication;

import android.app.ListActivity;
import android.content.Intent;
import android.net.Uri;
import android.os.AsyncTask;
import android.os.Bundle;
import android.view.View;
import android.widget.ArrayAdapter;
import android.widget.ListView;
import org.xmlpull.v1.XmlPullParser;
import org.xmlpull.v1.XmlPullParserException;
import org.xmlpull.v1.XmlPullParserFactory;
```

```java
import java.io.IOException;
import java.io.InputStream;
import java.net.MalformedURLException;
import java.net.URL;
import java.util.ArrayList;
import java.util.List;

public class MainActivityextends ListActivity
{
    List headlines;
    List links;

    @Override
protected void onCreate(Bundle savedInstanceState)
    {
super.onCreate(savedInstanceState);
new MyAsyncTask().execute();
    }

class MyAsyncTaskextends AsyncTask<Object,Void,ArrayAdapter>
    {
        @Override
protected ArrayAdapterdoInBackground(Object[] params)
        {
headlines = new ArrayList();
links = new ArrayList();
try
{
            URL url = new URL("https://codingconnect.net/feed");

XmlPullParserFactory factory = XmlPullParserFactory.newInstance();
factory.setNamespaceAware(false);
XmlPullParserxpp = factory.newPullParser();

// We will get the XML from an input stream
xpp.setInput(getInputStream(url), "UTF_8");
booleaninsideItem = false;

// Returns the type of current event: START_TAG, END_TAG, etc..
int eventType = xpp.getEventType();
while (eventType != XmlPullParser.END_DOCUMENT)
            {
if (eventType == XmlPullParser.START_TAG)
                {
if (xpp.getName().equalsIgnoreCase("item"))
                    {
insideItem = true;
                    }
else if (xpp.getName().equalsIgnoreCase("title"))
                    {
if (insideItem)
headlines.add(xpp.nextText()); //extract the headline
}
else if (xpp.getName().equalsIgnoreCase("link"))
                    {
```

```java
if (insideItem)
links.add(xpp.nextText()); //extract the link of article
}
                }
else if(eventType==XmlPullParser.END_TAG&&xpp.getName().equalsIgnoreCase("item"))
                {
insideItem=false;
                }
eventType = xpp.next(); //move to next element
}


        }
catch (MalformedURLException e)
        {
e.printStackTrace();
        }
catch (XmlPullParserException e)
        {
e.printStackTrace();
        }
catch (IOException e)
        {
e.printStackTrace();
        }
return null;
    }
protected void onPostExecute(ArrayAdapter adapter)
    {
        adapter = new ArrayAdapter(MainActivity.this, android.R.layout.simple_list_item_1,
headlines);
setListAdapter(adapter);
    }
  }

  @Override
protected void onListItemClick(ListView l, View v, int position, long id)
  {
    Uri uri = Uri.parse((links.get(position)).toString());
    Intent intent = new Intent(Intent.ACTION_VIEW, uri);
startActivity(intent);
  }

public InputStreamgetInputStream(URL url)
  {
try
{
return url.openConnection().getInputStream();
    }
catch (IOException e)
    {
return null;
    }
  }

AndroidManifest.xml
```

```xml
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.exno10" >




<application
android:allowBackup="true"
android:icon="@mipmap/ic_launcher"
android:label="@string/app_name"
android:supportsRtl="true"
android:theme="@style/AppTheme" >
<activity android:name="com.example.admin.myapplication.MainActivity" >
<intent-filter>
<action android:name="android.intent.action.MAIN" />

<category android:name="android.intent.category.LAUNCHER" />
</intent-filter>
</activity>
</application>
<uses-permission android:name="android.permission.INTERNET"/>

</manifest>
```
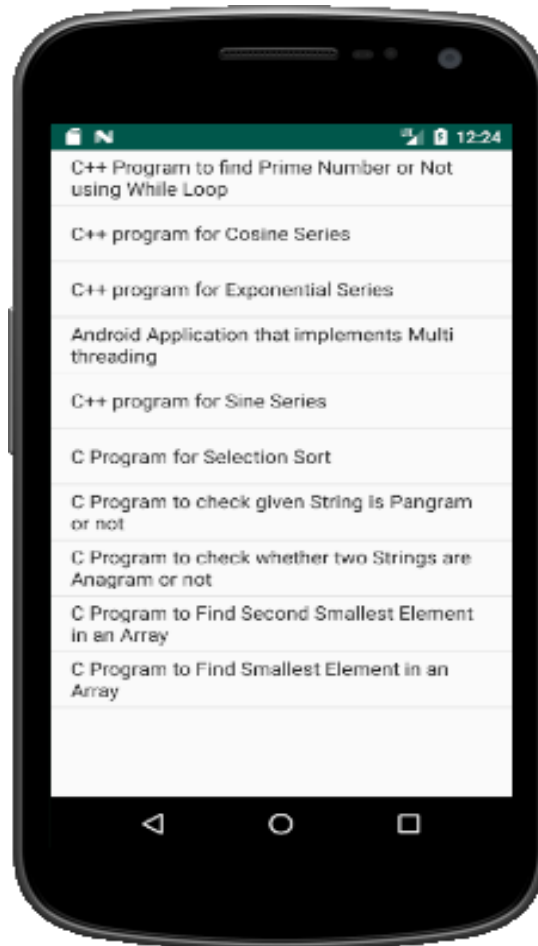
 activity_main.xml

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

<ListView
android:id="@+id/listView"
android:layout_width="match_parent"
android:layout_height="wrap_content" />

</LinearLayout>
```

OUTPUT:

RESULT:

Thus, the program for android application that makes use of RSS Feed was executed successfully.

EX. NO. :11	Develop a mobile application to send an email.
DATE :	

AIM:

      To develop an android application that send an email.

ALGORITHM:

1. Create a New Android Project:
   - Click New in the toolbar.
   - In the window that appears, open the Android folder, select Android Application Project, and click next.
   - Provide the application name and the project name and then finally give the desired package name.
   - Choose a launcher icon for your application and then select Blank Activity and then click Next
   - Provide the desired Activity name for your project and then click Finish.
2. Create a New AVD (Android Virtual Device):
   - click Android Virtual Device Manager from the toolbar.
   - In the Android Virtual Device Manager panel, click New.
   - Fill in the details for the AVD. Give it a name, a platform target, an SD card size, and a skin (HVGA is default).
   - Click Create AVD and Select the new AVD from the Android Virtual Device Manager and click Start.
3. Design the graphical layout.
4. Run the application.
5. When the application starts alarm sound will be invoked.
6. Stop alarm button is clicked to stop the alarm.
7. Close the Android project.

PROGRAM CODE:

MainActivity.java

```
package com.example.admin.myapplication;

import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;

public class MainActivityextends AppCompatActivity {

    @Override
```

```java
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
    Button startBtn = (Button) findViewById(R.id.sendbttn);
startBtn.setOnClickListener(new View.OnClickListener() {
public void onClick(View view) {
sendEmail();
        }
    });
  }

protected void sendEmail() {
Log.i("Send email", "");
    String[] TO = {
"muthuramalingam566@gmail.com"
};
    String[] CC = {
"ramdurai25@gmail.com"
};
    Intent emailIntent = new Intent(Intent.ACTION_SEND);
emailIntent.setData(Uri.parse("mailto:"));
emailIntent.setType("text/plain");
emailIntent.putExtra(Intent.EXTRA_EMAIL, TO);
emailIntent.putExtra(Intent.EXTRA_CC, CC);
emailIntent.putExtra(Intent.EXTRA_SUBJECT, "Your subject");
emailIntent.putExtra(Intent.EXTRA_TEXT, "Email message goes here");
try {
startActivity(Intent.createChooser(emailIntent, "Send mail..."));
        finish();
Log.i("Finished sending email...", "");
    } catch (android.content.ActivityNotFoundException ex) {
Toast.makeText(MainActivity.this, "There is no email client installed.",
Toast.LENGTH_SHORT).show();
    }
  }
}
```

activity_main.xml

```xml
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayoutxmlns: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:paddingBottom="@dimen/activity_vertical_margin"
android:paddingLeft="@dimen/activity_horizontal_margin"
android:paddingRight="@dimen/activity_horizontal_margin"
android:paddingTop="@dimen/activity_vertical_margin"
tools:context="com.example.admin.myapplication.MainActivity">
<EditTextandroid:layout_width="wrap_content"
android:layout_height="wrap_content"
android:inputType="textEmailAddress"
android:ems="10"
```

```xml
        android:id="@+id/editText"
        android:layout_alignParentTop="true"
        android:layout_alignParentRight="true"
        android:layout_alignParentEnd="true" />
    <EditText
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:inputType="textEmailAddress"
        android:ems="10"
        android:id="@+id/editText2"
        android:layout_below="@+id/editText"
        android:layout_alignRight="@+id/editText"
        android:layout_alignEnd="@+id/editText" />
    <EditText
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:inputType="textEmailAddress"
        android:ems="10"
        android:id="@+id/editText3"
        android:layout_below="@+id/editText2"
        android:layout_alignRight="@+id/editText2"
        android:layout_alignEnd="@+id/editText2" />
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="SEND MAIL"
        android:id="@+id/sendbttn"
        android:layout_centerVertical="true"
        android:layout_alignLeft="@+id/editText3"
        android:layout_alignStart="@+id/editText3" />
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Recipient"
        android:id="@+id/textView"
        android:layout_alignBottom="@+id/editText"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true" />
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="subject"
        android:id="@+id/textView2"
        android:layout_alignBottom="@+id/editText2"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true" />
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Message Body"
        android:id="@+id/textView3"
        android:layout_alignBottom="@+id/editText3"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true" />
</RelativeLayout>
```

OUTPUT:
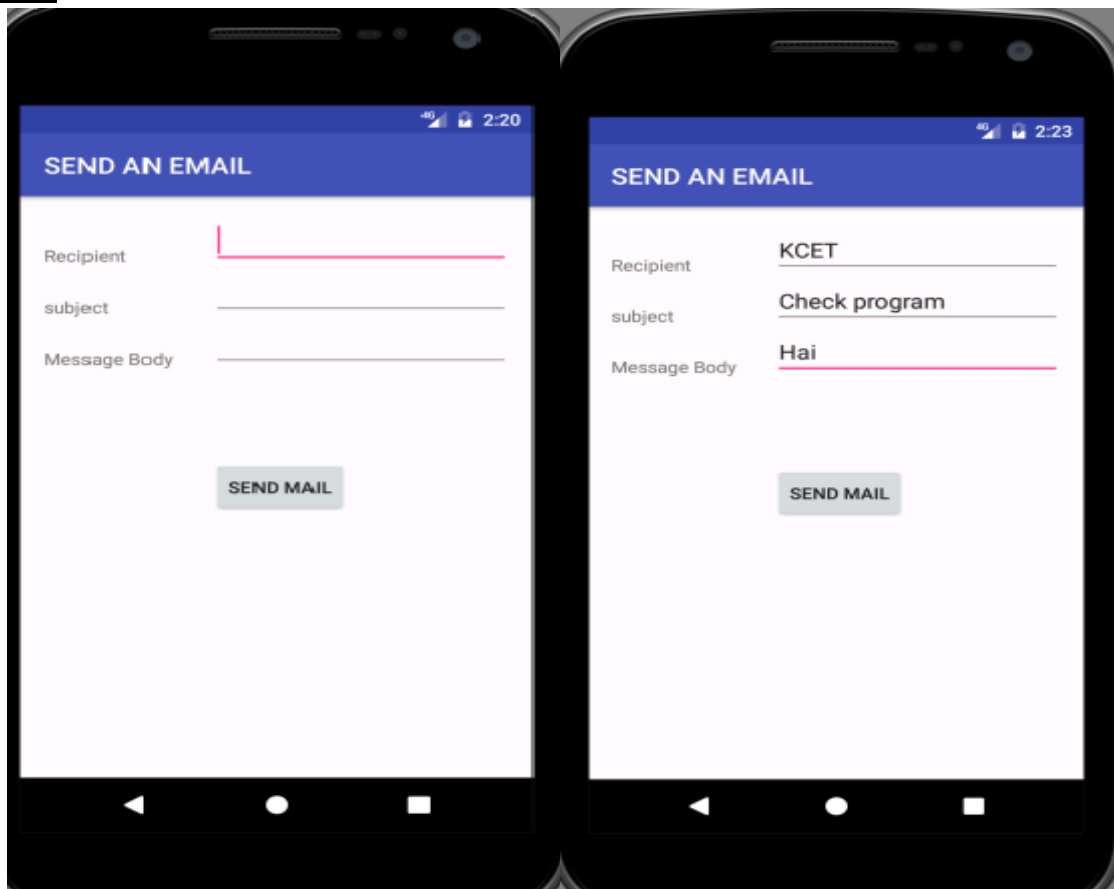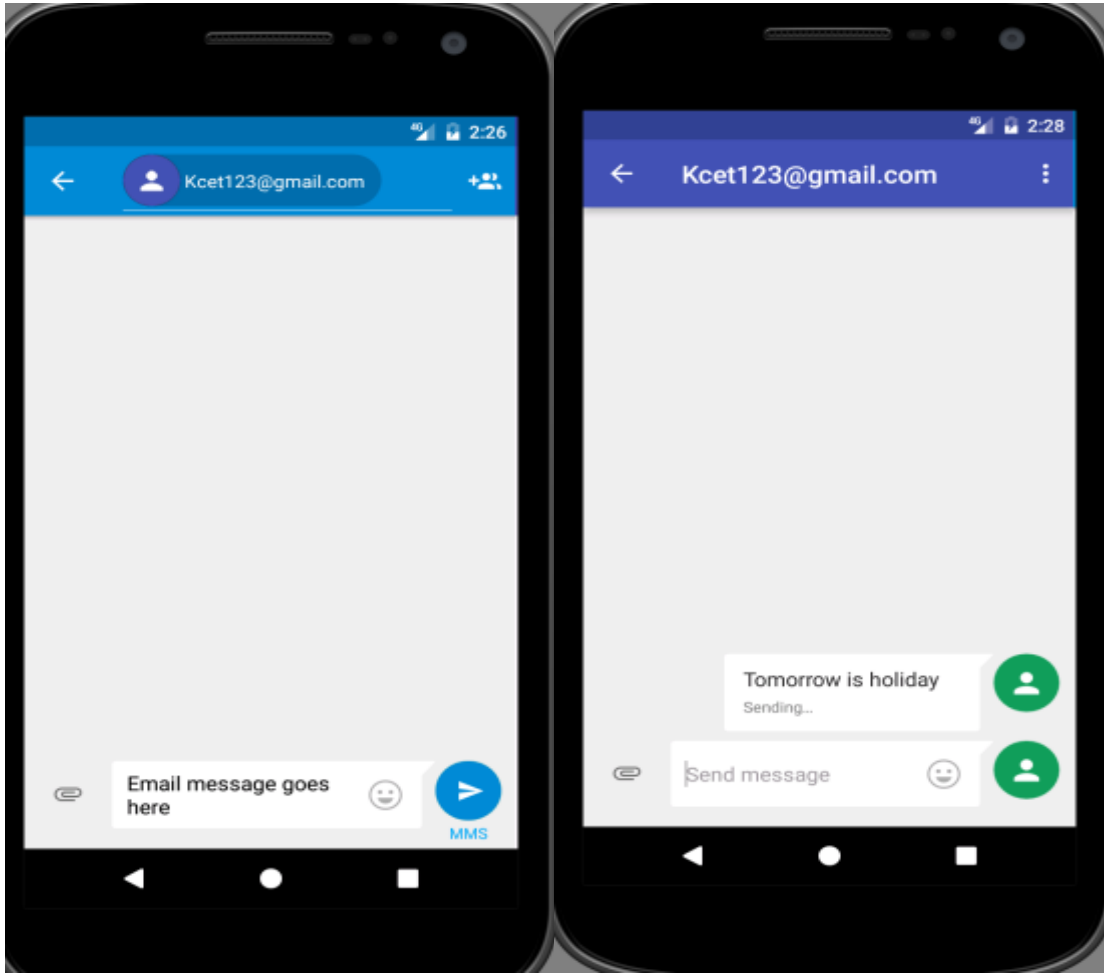
RESULT:

Thus, the program for android application to send an email was executed successfully.

EX. NO. :12	Mini Project 1
DATE :	Write a mobile application that creates alarm clock

AIM:

To develop an android application that creates alarm clock.

ALGORITHM:

1.Create a New Android Project:
- Click New   in the toolbar.
- In the window that appears, open the Android folder, select Android Application Project, and click next.
- Provide the application name and the project name and then finally give the desired package name.
- Choose a launcher icon for your application and then select Blank Activity and then click Next
- Provide the desired Activity name for your project and then click Finish.

2.Create a New AVD (Android Virtual Device):
- click Android Virtual Device Manager from the toolbar.
- In the Android Virtual Device Manager panel, click New.
- Fill in the details for the AVD. Give it a name, a platform target, an SD card size, and a skin (HVGA is default).
- Click Create AVD and Select the new AVD from the Android Virtual Device Manager and click Start.

3.Design the graphical layout.

4.Run the application.

5.When the application starts alarm sound will be invoked.

6.Stop alarm button is clicked to stop the alarm.

7.Close the Android project.

PROGRAM CODE:

MainActivity.java

```
package com.example.admin.myapplication;

import android.app.AlarmManager;
import android.app.PendingIntent;
import android.content.Intent;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.widget.TimePicker;
import android.widget.Toast;
import android.widget.ToggleButton;

import java.util.Calendar;
```

```java
public class MainActivityextends AppCompatActivity
{
TimePickeralarmTimePicker;
PendingIntentpendingIntent;
AlarmManageralarmManager;

    @Override
protected void onCreate(Bundle savedInstanceState)
    {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
alarmTimePicker= (TimePicker) findViewById(R.id.timePicker);
alarmManager= (AlarmManager) getSystemService(ALARM_SERVICE);
    }
public void OnToggleClicked(View view)
    {
long time;
if (((ToggleButton) view).isChecked())
        {
Toast.makeText(MainActivity.this, "ALARM ON", Toast.LENGTH_SHORT).show();
        Calendar calendar = Calendar.getInstance();
calendar.set(Calendar.HOUR_OF_DAY, alarmTimePicker.getCurrentHour());
calendar.set(Calendar.MINUTE, alarmTimePicker.getCurrentMinute());
        Intent intent = new Intent(this, AlarmReceiver.class);
pendingIntent= PendingIntent.getBroadcast(this, 0, intent, 0);

        time=(calendar.getTimeInMillis()-(calendar.getTimeInMillis()%60000));
if(System.currentTimeMillis()>time)
        {
if (calendar.AM_PM== 0)
            time = time + (1000*60*60*12);
else
time = time + (1000*60*60*24);
        }
alarmManager.setRepeating(AlarmManager.RTC_WAKEUP, time, 10000, pendingIntent);
    }
else
{
alarmManager.cancel(pendingIntent);
Toast.makeText(MainActivity.this, "ALARM OFF", Toast.LENGTH_SHORT).show();
    }
  }
}
```

AlarmReceiverActivity.java

```java
package com.example.admin.myapplication;

import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.media.Ringtone;
import android.media.RingtoneManager;
import android.net.Uri;
```

```java
import android.widget.Toast;

public class AlarmReceiverextends BroadcastReceiver
{
    @Override
public void onReceive(Context context, Intent intent)
    {
Toast.makeText(context, "Alarm! Wake up! Wake up!", Toast.LENGTH_LONG).show();
     Uri alarmUri = RingtoneManager.getDefaultUri(RingtoneManager.TYPE_ALARM);
if (alarmUri == null)
      {
alarmUri = RingtoneManager.getDefaultUri(RingtoneManager.TYPE_NOTIFICATION);
      }
     Ringtone ringtone = RingtoneManager.getRingtone(context, alarmUri);
ringtone.play();
    }
}
```

activity_main.xml

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

<TimePicker
android:id="@+id/timePicker"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_gravity="center" />

<ToggleButton
android:id="@+id/toggleButton"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_gravity="center"
android:layout_margin="20dp"
android:checked="false"
android:onClick="OnToggleClicked" />

</LinearLayout>
```

AndroidMainfest.Xml

```xml
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
  package="com.example.admin.myapplication">

<application
android:allowBackup="true"
android:icon="@mipmap/ic_launcher"
android:label="@string/app_name"
android:roundIcon="@mipmap/ic_launcher_round"
android:supportsRtl="true"
android:theme="@style/AppTheme">
```

```xml
<activity android:name=".MainActivity">
<intent-filter>
<action android:name="android.intent.action.MAIN" />

<category android:name="android.intent.category.LAUNCHER" />
</intent-filter>
</activity>
<receiver android:name=".AlarmReceiver" >
</receiver>
</application>

</manifest>
```
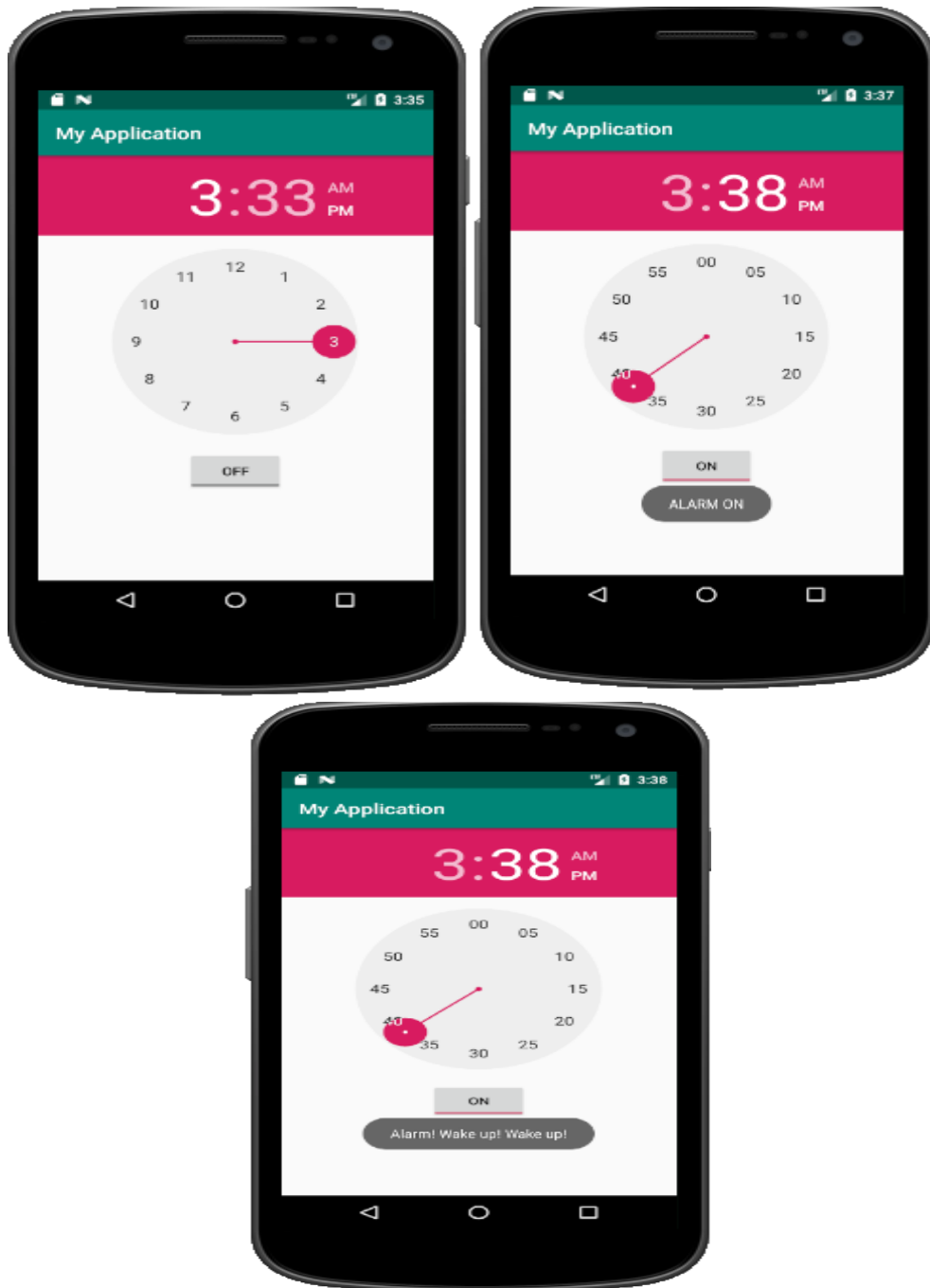
OUTPUT:

RESULT:

Thus, the program for android application for an alarm was executed successfully.

EX. NO. :12	**Mini Project2**
DATE :	**Develop a native calculator application**

AIM:

      To develop a calculator android application.

ALGORITHM:

1.   Create a New Android Project:
   - Click New in the toolbar.
   - In the window that appears, open the Android folder, select Android Application Project, and click next.
   - Provide the application name and the project name and then finally give the desired package name.
   - Choose a launcher icon for your application and then select Blank Activity and then click Next
   - Provide the desired Activity name for your project and then click Finish.
2.   Create a New AVD (Android Virtual Device):
   - click Android Virtual Device Manager from the toolbar.
   - In the Android Virtual Device Manager panel, click New.
   - Fill in the details for the AVD. Give it a name, a platform target, an SD card size, and a skin (HVGA is default).
   - Click Create AVD and Select the new AVD from the Android Virtual Device Manager and click Start.
3.   Run the application.
4.   Provide any two input numbers.
5.   Choose any arithmetic operations of your choice and the output gets displayed on the display screen of the calculator application.
6.   Close the Android project.

PROGRAM CODE:

MainActivity.java

```
package com.example.calculator_two;
import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.EditText;

public class MainActivity extends Activity implements OnClickListener {
        private Button nine, eig, sev, six, fiv, four, thr, two, one, zero, dot,
                        plus, mins, div, mul, eq, cl;
        private EditText et;
```

```java
        private String s = "0";
        private int result = 0;
        private char lO = ' ';

        protected void onCreate(Bundle savedInstanceState) {
                // TODO Auto-generated method stub
                super.onCreate(savedInstanceState);
                setContentView(R.layout.activity_main);
                nine = (Button) findViewById(R.id.b9);
                eig = (Button) findViewById(R.id.b8);
                sev = (Button) findViewById(R.id.b7);
                six = (Button) findViewById(R.id.b6);
                fiv = (Button) findViewById(R.id.b5);
                four = (Button) findViewById(R.id.b4);
                thr = (Button) findViewById(R.id.b3);
                two = (Button) findViewById(R.id.b2);
                one = (Button) findViewById(R.id.b1);
                zero = (Button) findViewById(R.id.b0);
                dot = (Button) findViewById(R.id.bd);
                plus = (Button) findViewById(R.id.bpl);
                mins = (Button) findViewById(R.id.bmin);
                div = (Button) findViewById(R.id.bdiv);
                mul = (Button) findViewById(R.id.bmul);
                eq = (Button) findViewById(R.id.beq);
                cl = (Button) findViewById(R.id.bcl);
                et = (EditText) findViewById(R.id.tv);
                zero.setOnClickListener(this);
                nine.setOnClickListener(this);
                eig.setOnClickListener(this);
                sev.setOnClickListener(this);
                six.setOnClickListener(this);
                fiv.setOnClickListener(this);
                four.setOnClickListener(this);
                thr.setOnClickListener(this);
                two.setOnClickListener(this);
                one.setOnClickListener(this);
                dot.setOnClickListener(this);
                plus.setOnClickListener(this);
                mins.setOnClickListener(this);
                div.setOnClickListener(this);
                mul.setOnClickListener(this);
                eq.setOnClickListener(this);
                cl.setOnClickListener(this);
                et.setOnClickListener(this);
        }

        public void onClick(View v) {
```

```java
switch (v.getId()) {
case R.id.b0:
case R.id.b1:
case R.id.b2:
case R.id.b3:
case R.id.b4:
case R.id.b5:
case R.id.b6:
case R.id.b7:
case R.id.b8:
case R.id.b9:

        String inDigit = ((Button) v).getText().toString();
        if (s.equals("0")) {
                s = inDigit;
        } else {
                s += inDigit;
        }
        et.setText(s);
        if (lO == '=') {
                result = 0;
                lO = ' ';
        }
        break;
case R.id.bpl:
        compute();
        lO = '+';
        break;
case R.id.bmin:
        compute();
        lO = '-';
        break;
case R.id.bdiv:
        compute();
        lO = '/';
        break;
case R.id.bmul:
        compute();
        lO = '*';
        break;
case R.id.beq:
        compute();
        lO = '=';
        break;
case R.id.bcl:
        result = 0;
        s = "0";
```

```
                        lO = ' ';
                        et.setText("0");
                        break;
                }
        }

        private void compute() {
                int inNum = Integer.parseInt(s);
                s = "0";
                if (lO == ' ') {
                        result = inNum;
                } else if (lO == '+') {
                        result += inNum;
                } else if (lO == '-') {
                        result -= inNum;
                } else if (lO == '*') {
                        result *= inNum;
                } else if (lO == '/') {
                        result /= inNum;
                } else if (lO == '=') {
                        // Keep the result for the next operation
                }
                et.setText(String.valueOf(result));
        }
}
```

activity_main.xml

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

<EditText
android:id="@+id/tv"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:hint="enter no. here"
android:textSize="30dp" />

<LinearLayout
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_marginTop="100dp"
android:orientation="horizontal"
android:weightSum="4">
```

```xml
<Button
android:id="@+id/b9"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_weight="1"
android:text="9"
android:textColor="#ff0000" />

<Button
android:id="@+id/b8"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_weight="1"
android:text="8"
android:textColor="#ff0000" />

<Button
android:id="@+id/b7"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_weight="1"
android:text="7"
android:textColor="#ff0000" />

<Button
android:id="@+id/bpl"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_weight="1"
android:text="+"
android:textColor="#ff0000" />
</LinearLayout>

<LinearLayout
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:orientation="horizontal"
android:weightSum="4">

<Button
android:id="@+id/b6"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_weight="1"
android:text="6"
android:textColor="#ff0000" />
```

```xml
<Button
android:id="@+id/b5"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_weight="1"
android:text="5"
android:textColor="#ff0000" />

<Button
android:id="@+id/b4"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_weight="1"
android:text="4"
android:textColor="#ff0000" />

<Button
android:id="@+id/bmin"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_weight="1"
android:text="-"
android:textColor="#ff0000" /></LinearLayout>

<LinearLayout
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:orientation="horizontal"
android:weightSum="4">

<Button
android:id="@+id/b3"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_weight="1"
android:text="3"
android:textColor="#ff0000" />

<Button
android:id="@+id/b2"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_weight="1"
android:text="2"
android:textColor="#ff0000" />
```

```xml
<Button
android:id="@+id/b1"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_weight="1"
android:text="1"
android:textColor="#ff0000" />

<Button
android:id="@+id/bmul"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_weight="1"
android:text="*"
android:textColor="#ff0000" />
</LinearLayout>

<LinearLayout
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:orientation="horizontal"
android:weightSum="5">

<Button
android:id="@+id/bd"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_weight="1"
android:text="."
android:textColor="#ff0000" />

<Button
android:id="@+id/b0"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_weight="1"
android:text="0"
android:textColor="#ff0000" />

<Button
android:id="@+id/bcl"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_weight="1"
```

```
android:text="Clr"
android:textColor="#ff0000" />


<Button
android:id="@+id/beq"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_weight="1"
android:text="="
android:textColor="#ff0000" />

<Button
android:id="@+id/bdiv"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_weight="1"
android:text="/"
android:textColor="#ff0000" />
</LinearLayout>

</LinearLayout>
```

OUTPUT:
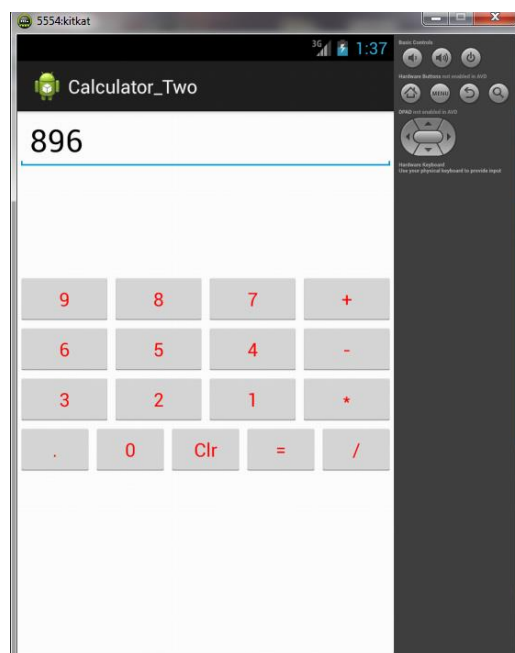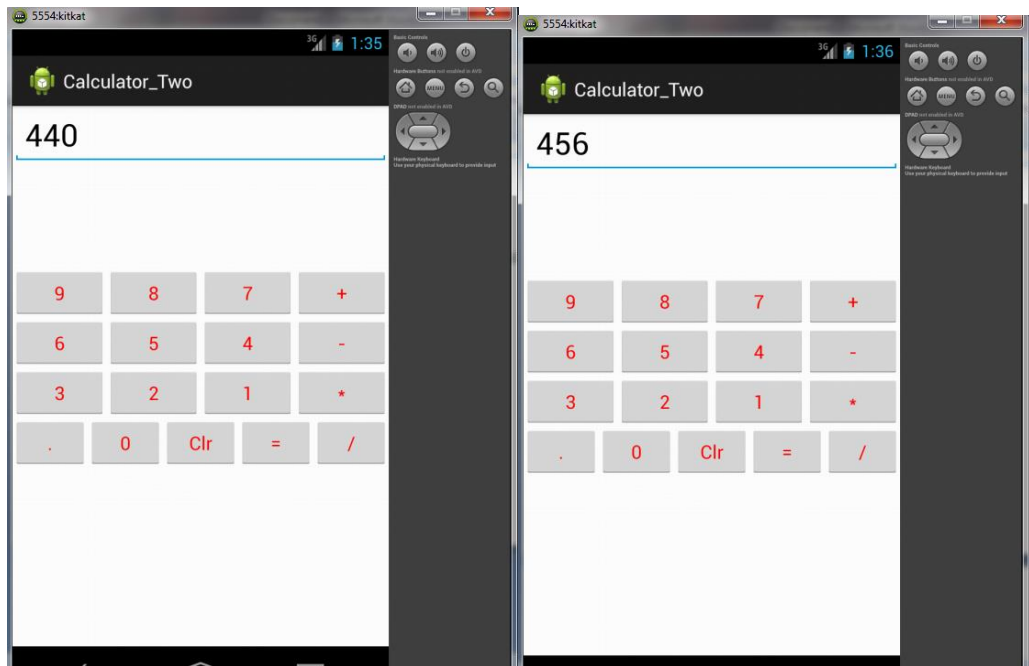
RESULT:

Thus, the program for android based calculator application was executed successfully.

VIVA QUESTIONS & ANSWERS

1) What is Android?

It is an open-sourced operating system that is used primarily on mobile devices, such as cell phones and tablets. It is a Linux kernel-based system that's been equipped with rich components that allows developers to create and run apps that can perform both basic and advanced functions.

2) What Is the Google Android SDK?

The Google Android SDK is a toolset that developers need in order to write apps on Android enabled devices. It contains a graphical interface that emulates an Android driven handheld environment, allowing them to test and debug their codes.

3) What is the Android Architecture?

Android Architecture is made up of 4 key components:
– Linux Kernel
– Libraries
– Android Framework
– Android Applications

4) Describe the Android Framework.

The Android Framework is an important aspect of the Android Architecture. Here you can find all the classes and methods that developers would need in order to write applications on the Android environment.

5) What is AAPT?

AAPT is short for Android Asset Packaging Tool. This tool provides developers with the ability to deal with zip-compatible archives, which includes creating, extracting as well as viewing its contents.

6) What is the importance of having an emulator within the Android environment?

The emulator lets developers "play" around an interface that acts as if it were an actual mobile device. They can write and test codes, and even debug. Emulators are a safe place for testing codes especially if it is in the early design phase.

7) What is the use of an activityCreator?

An activityCreator is the first step towards the creation of a new Android project. It is made up of a shell script that will be used to create new file system structure necessary for writing codes within the Android IDE.

8) Describe Activities.

Activities are what you refer to as the window to a user interface. Just as you create windows in order to display output or to ask for an input in the form of dialog boxes, activities play the same role, though it may not always be in the form of a user interface.

9) What are Intents?

Intents displays notification messages to the user from within the Android enabled device. It can be used to alert the user of a particular state that occurred. Users can be made to respond to intents.

10) Differentiate Activities from Services.

Activities can be closed, or terminated anytime the user wishes. On the other hand, services are designed to run behind the scenes, and can act independently. Most services run continuously, regardless of whether there are certain or no activities being executed.

11) What items are important in every Android project?

These are the essential items that are present each time an Android project is created:
– AndroidManifest.xml
– build.xml
– bin/
– src/
– res/
– assets/

12) What is the importance of XML-based layouts?

The use of XML-based layouts provides a consistent and somewhat standard means of setting GUI definition format. In common practice, layout details are placed in XML files while other items are placed in source files.

13) What are containers?

Containers, as the name itself implies, holds objects and widgets together, depending on which specific items are needed and in what particular arrangement that is wanted. Containers may hold labels, fields, buttons, or even child containers, as examples.

14) What is Orientation?

Orientation, which can be set using setOrientation(), dictates if the LinearLayout is represented as a row or as a column. Values are set as either HORIZONTAL or VERTICAL.

15) What is the importance of Android in the mobile market?

Developers can write and register apps that will specifically run under the Android environment. This means that every mobile device that is Android enabled will be able to support and run these apps. With the growing popularity of Android mobile devices, developers can take advantage of this trend by creating and uploading their apps on the Android Market for distribution to anyone who wants to download it.

16) What do you think are some disadvantages of Android?

Given that Android is an open-source platform, and the fact that different Android operating systems have been released on different mobile devices, there's no clear cut policy to how applications can adapt with various OS versions and upgrades. One app that runs on this particular version of Android OS may or may not run on another version. Another disadvantage is that since mobile devices such as phones and tabs come in different sizes and forms, it poses a challenge for developers to create apps that can adjust correctly to the right screen size and other varying features and specs.

17) What is adb?

Adb is short for Android Debug Bridge. It allows developers the power to execute remote shell commands. Its basic function is to allow and control communication towards and from the emulator port.

18) What are the four essential states of an activity?

– Active – if the activity is at the foreground
– Paused – if the activity is at the background and still visible
– Stopped – if the activity is not visible and therefore is hidden or obscured by another activity
– Destroyed – when the activity process is killed or completed terminated

19) What is ANR?

ANR is short for Application Not Responding. This is actually a dialog that appears to the user whenever an application have been unresponsive for a long period of time.

20) Which elements can occur only once and must be present?

Among the different elements, the and elements must be present and can occur only once. The rest are optional, and can occur as many times as needed.

21) How are escape characters used as attribute?

Escape characters are preceded by double backslashes. For example, a newline character is created using '\\n'

22) What is the importance of settings permissions in app development?

Permissions allow certain restrictions to be imposed primarily to protect data and code. Without these, codes could be compromised, resulting to defects in functionality.

23) What is the function of an intent filter?

Because every component needs to indicate which intents they can respond to, intent filters are used to filter out intents that these components are willing to receive. One or more intent filters are possible, depending on the services and activities that is going to make use of it.

24) Enumerate the three key loops when monitoring an activity

– Entire lifetime – activity happens between onCreate and onDestroy
– Visible lifetime – activity happens between onStart and onStop
– Foreground lifetime – activity happens between onResume and onPause

25) When is the onStop() method invoked?

A call to onStop method happens when an activity is no longer visible to the user, either because another activity has taken over or if in front of that activity.

26) Is there a case wherein other qualifiers in multiple resources take precedence over locale?

Yes, there are actually instances wherein some qualifiers can take precedence over locale. There are two known exceptions, which are the MCC (mobile country code) and MNC (mobile network code) qualifiers.

27) What are the different states wherein a process is based?

There are 4 possible states:
– foreground activity
– visible activity
– background activity
– empty process

28) How can the ANR be prevented?

One technique that prevents the Android system from concluding a code that has been responsive for a long period of time is to create a child thread. Within the child thread, most of the actual workings of the codes can be placed, so that the main thread runs with minimal periods of unresponsive times.

29) What role does Dalvik play in Android development?

Dalvik serves as a virtual machine, and it is where every Android application runs. Through Dalvik, a device is able to execute multiple virtual machines efficiently through better memory management.

30) What is the AndroidManifest.xml?

This file is essential in every application. It is declared in the root directory and contains information about the application that the Android system must know before the codes can be executed.

31) What is the proper way of setting up an Android-powered device for app development?

The following are steps to be followed prior to actual application development in an Android-powered device:
-Declare your application as "debuggable" in your Android Manifest.
-Turn on "USB Debugging" on your device.
-Set up your system to detect your device.

32) Enumerate the steps in creating a bounded service through AIDL.

1. create the .aidl file, which defines the programming interface
2. implement the interface, which involves extending the inner abstract Stub class as well as implanting its methods.
3. expose the interface, which involves implementing the service to the clients.

33) What is the importance of Default Resources?

When default resources, which contain default strings and files, are not present, an error will occur and the app will not run. Resources are placed in specially named subdirectories under the project res/ directory.

34) When dealing with multiple resources, which one takes precedence?

Assuming that all of these multiple resources are able to match the configuration of a device, the 'locale' qualifier almost always takes the highest precedence over the others.

35) When does ANR occur?

The ANR dialog is displayed to the user based on two possible conditions. One is when there is no response to an input event within 5 seconds, and the other is when a broadcast receiver is not done executing within 10 seconds.

36) What is AIDL?

AIDL, or Android Interface Definition Language, handles the interface requirements between a client and a service so both can communicate at the same level through interprocess communication or IPC. This process involves breaking down objects into primitives that Android can understand. This part is required simply because a process cannot access the memory of the other process.

37) What data types are supported by AIDL?

AIDL has support for the following data types:
-string
-charSequence

-List
-Map
-all native Java data types like int,long, char and Boolean

38) What is a Fragment?

A fragment is a part or portion of an activity. It is modular in a sense that you can move around or combine with other fragments in a single activity. Fragments are also reusable.

39) What is a visible activity?

A visible activity is one that sits behind a foreground dialog. It is actually visible to the user, but not necessarily being in the foreground itself.

40) When is the best time to kill a foreground activity?

The foreground activity, being the most important among the other states, is only killed or terminated as a last resort, especially if it is already consuming too much memory. When a memory paging state has been reach by a foreground activity, then it is killed so that the user interface can retain its responsiveness to the user.

41) Is it possible to use or add a fragment without using a user interface?

Yes, it is possible to do that, such as when you want to create a background behavior for a particular activity. You can do this by using add(Fragment,string) method to add a fragment from the activity.

42) How do you remove icons and widgets from the main screen of the Android device?

To remove an icon or shortcut, press and hold that icon. You then drag it downwards to the lower part of the screen where a remove button appears.

43) What are the core components under the Android application architecture?

There are 5 key components under the Android application architecture:
– services
– intent
– resource externalization
– notifications
– content providers

44) What composes a typical Android application project?

A project under Android development, upon compilation, becomes an .apk file. This apk file format is actually made up of the AndroidManifest.xml file, application code, resource files, and other related files.

45) What is a Sticky Intent?

A Sticky Intent is a broadcast from sendStickyBroadcast() method such that the intent floats around even after the broadcast, allowing others to collect data from it.

46) Do all mobile phones support the latest Android operating system?

Some Android-powered phone allows you to upgrade to the higher Android operating system version. However, not all upgrades would allow you to get the latest version. It depends largely on the capability and specs of the phone, whether it can support the newer features available under the latest Android version.

47) What is portable wi-fi hotspot?

Portable Wi-Fi Hotspot allows you to share your mobile internet connection to other wireless device. For example, using your Android-powered phone as a Wi-Fi Hotspot, you can use your laptop to connect to the Internet using that access point.

48) What is an action?

In Android development, an action is what the intent sender wants to do or expected to get as a response. Most application functionality is based on the intended action.

49) What is the difference between a regular bitmap and a nine-patch image?

In general, a Nine-patch image allows resizing that can be used as background or other image size requirements for the target device. The Nine-patch refers to the way you can resize the image: 4 corners that are unscaled, 4 edges that are scaled in 1 axis, and the middle one that can be scaled into both axes.

50) What language is supported by Android for application development?

The main language supported is Java programming language. Java is the most popular language for app development, which makes it ideal even for new Android developers to quickly learn to create and deploy applications in the Android environment.