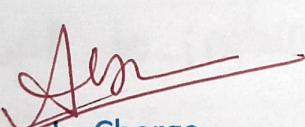


Thadomal Shahani Engineering College

Bandra (W.), Mumbai- 400 050.

© CERTIFICATE ©

Certify that ~~Mr~~/Miss Niyati Savant
of Computer Department, Semester VI with
Roll No. 2103156 has completed a course of the necessary
experiments in the subject Mobile Computing under my
supervision in the **Thadomal Shahani Engineering College**
Laboratory in the year 2023 - 2024


Teacher In-Charge

Head of the Department

Date 20/3/24

Principal

CONTENTS

SR. NO.	EXPERIMENTS	PAGE NO.	DATE	TEACHERS SIGN.
1.	Write an Android Application to draw basic graphical 2D primitives	1 - 7	17/11/24	
2.	WAA to draw Basic graphical 3D primitives	8 - 15	24/11/24	
3.	WAA to design a form with GUI Components	16 - 22	31/11/24	
4.	WAA to design GUI Components using Database	23 - 42	7/12/24	
5	WAA to develop an EMI calculator application	43 - 51	21/12/24	14/12/2024
6	WAA that creates an alert on receiving a message.	52 - 57	28/12/24	20/12/2024
7	WAA to implement Basic calculator	58 - 66	6/13/24	
8	Write a program to demonstrate Cellular Frequency reuse.	67 - 76	13/13/24	
9	Write a program to explain Concept of DSSS	77 - 83	20/13/24	
10	Write a program to implement A3/A5/A8 GSM security Algorithm	84 - 96	20/13/24	

CONTENTS

Experiment - 1

Aim - Android Application to draw Basic graphical 2D primitives

Theory -

Android Studio is a software cum development kit and integrated development (IDE) owned by Google. It is the official software to write and test mobile applications for Android OS. It is used for designing UI and well as Java front-end logic. It uses Java or Kotlin for the logic designing and XML (Extended Markup Language) for designing the user interface. Prominent features of AS include

- 1) Gradle-based build support
- 2) Android-specific refactoring and quick fixes
- 3) Support to build Android Wear apps
- 4) A rich layout editor that allows users to drag and drop UI components, option to preview layouts on multiple screen configurations

After completing the setup of Android Architecture we can create an android application in the studio. A new project for each application has

to be created and contains different types of app modules, source code files, and resource files.

a) Manifest Folder

Contains the `AndroidManifest.xml` for creating our android application. This contains information about our application like Android version, metadata, states package for Kotlin file, and other application components.

b) Java Folder

It contains all the java and Kotlin source code (`.java`) that we create during the app development including other test files. On creating any new project using Kotlin, class file `MainActivity.kt` file is created.

c) Resource (res) Folder

It is the most important folder contains all non-code sources like images, XML layouts, and UI strings.

i) res/drawable folder

Contains different types of images used for development of application

ii) res/layout folder

Contains all XML layout files which we use to define the UI of the application. It contains the `activity_main.xml` file.

iii) res/mipmap folder

Contains launcher.xml files to define icons used to show on home screen. Contains different density types of icons like hdpi, mdpi, xhdpi

iv) res/values folder

Contains number of XML files like strings, dimensions, colors and style definitions.

D) Gradle scripts folder

Gradle means automated build system and it contains a number of files that are used to define a build configuration that can be applied to all modules in our application

Steps to start a project

- 1) Install Android studio
- 2) Select 'Start a new Android Studio project' option
- 3) Name your application in 'Application name' Text box
- 4) Select the minimum SDK to select OS which must be the latest version to run the app. Click on Next
- 5) Choose 'Empty Activity' as Activity to mobile
- 6) Fill Activity name text field
- 7) Click on finish. A default app is created with all default files and now you can start coding

The built-in functions used are -

- 1) Bitmap. CreateBitmap(int width, int height, Bitmap.Config config)
Creates a new bitmap of self-specified details
- 2) Canvas(Canvas, canvas)
Construct a canvas with specified bitmap to draw into
- 3) Paint.setColor(int color)
Set color for subsequent drawing
- 4) Canvas.drawText(String text, float x, float y, Paint paint)
Draws specified text, starting at specified coordinates using specified paint
- 5) Canvas.drawRect(float left, float top, float right, float bottom, Paint paint)
Draws specified rectangle with specified paint
- 6) Canvas.drawCircle(float cx, float cy, float radius, Paint paint)
Draws ~~filled~~ circle with specified paint

Conclusion -

Understood the use of Android's graphics capabilities to create custom drawings by utilizing built-in functions to create visually appealing & interactive UI tailored to specific application requirement.

✓

✗

Experiment 1

Aim: To draw Basic graphical 2D primitives.

Java Program:

```
package com.example.c31_156;

import androidx.appcompat.app.AppCompatActivity;

import android.graphics.Bitmap;
import android.graphics.Canvas;
import android.graphics.Color;
import android.graphics.Paint;
import android.os.Bundle;
import android.widget.ImageView;

public class MainActivity extends AppCompatActivity{
    Bitmap bg;
    ImageView img;

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

        // create bitmap
        bg=Bitmap.createBitmap(720,1280, Bitmap.Config.RGB_565);
        img = findViewById(R.id.imageView);
        img.setImageBitmap(bg);

        // canvas creation
        Canvas canvas= new Canvas(bg);

        // paint object creation
        Paint paint1= new Paint();
        paint1.setColor(Color.MAGENTA);
        paint1.setTextSize(50);
        Paint paint2 = new Paint();
        paint2.setColor(Color.BLUE);
        paint2.setTextSize(50);
        Paint paint3 = new Paint();
        paint3.setColor(Color.GREEN);
        paint3.setTextSize(50);
        Paint paint4 = new Paint();
        paint4.setColor(Color.WHITE);
        paint4.setTextSize(50);
    }
}
```

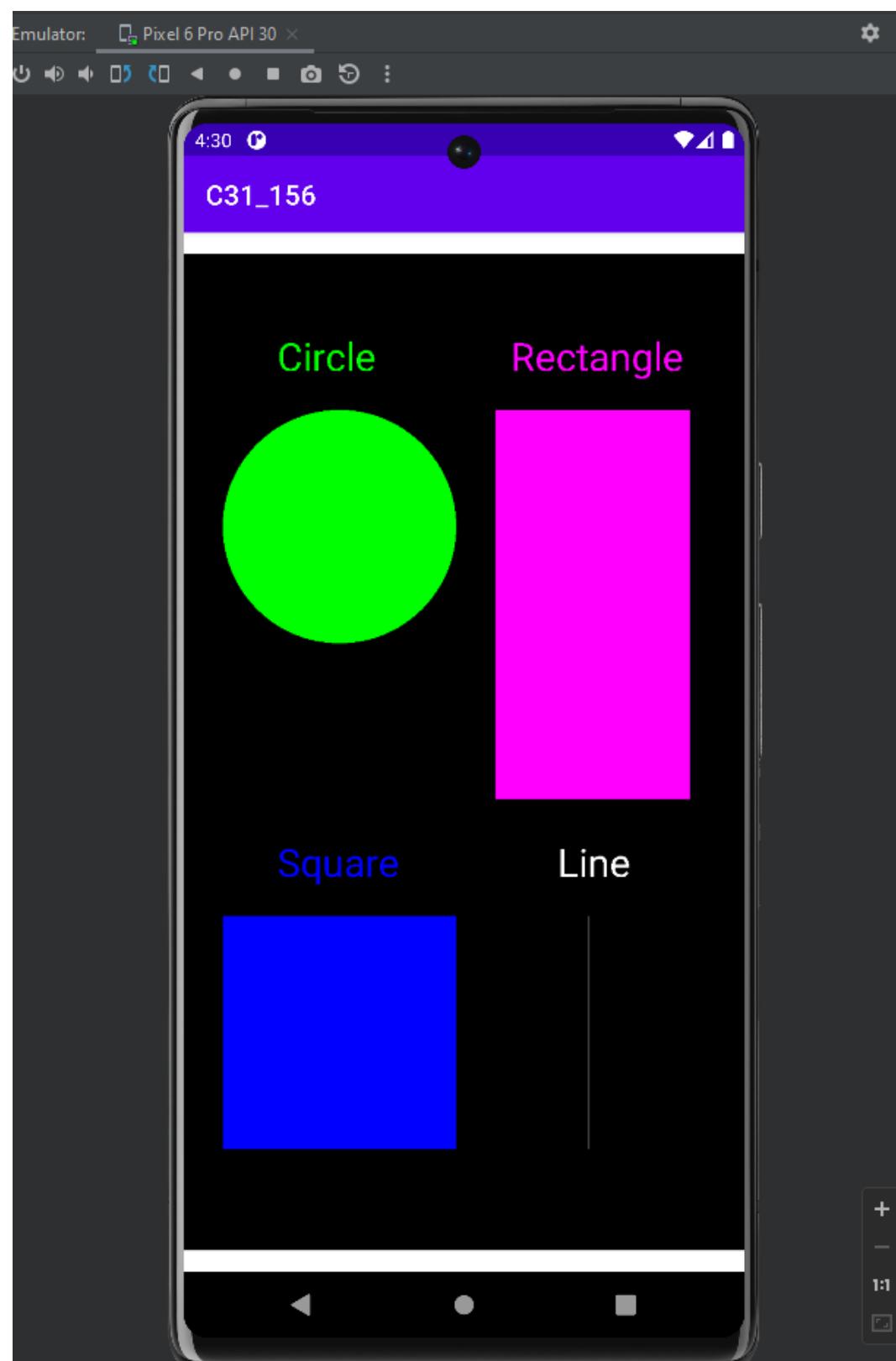
Niyati_Savant_TE_C31_2103156

```
// draw rectangle  
canvas.drawText("Rectangle",420,150,paint1);  
canvas.drawRect(400,200,650,700,paint1);  
  
// draw square  
canvas.drawText("Square",120,800,paint2);  
canvas.drawRect(50,850,350,1150,paint2);  
  
// circle  
canvas.drawText("Circle",120,150,paint3);  
canvas.drawCircle(200,350,150,paint3);  
  
// draw line  
canvas.drawText("Line",480,800,paint4);  
canvas.drawLine(520,850,520,1150,paint4);  
  
}  
}
```

XML program:

```
<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
    xmlns:app="http://schemas.android.com/apk/res-auto"  
    xmlns:tools="http://schemas.android.com/tools"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    tools:context=".MainActivity">  
  
<ImageView  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:id="@+id/imageView"/>  
</LinearLayout>
```

Output:



Aim - Android Application to draw Basic graphical 3D primitives.

Theory

The android program, written in Java creates a simple 3D shape drawing application using canvas & Bitmap classes. The built-in functions are

1. Package & Import:

This program is part of the 'com.example.a3dshop' package. Import necessary classes for Android app development including 'AppCompatActivity', 'Bitmap', 'Canvas', 'Color', 'Paint', 'Bundle', & 'ImageView'.

2. Class and Variables

The main activity class is 'MainActivity', extending 'AppCompatActivity'. Declare Bitmap 'bg' and ImageView 'img' as class variables.

3. OnCreate Method:

Initialize the activity, sets the content view from 'activity_main.xml'. Create a Bitmap 'bg' with dimensions 720×1280 pixels and sets it as the background for an ImageView. Initialize a Canvas object associated with the Bitmap and fills it with a white color. Creates a Paint object ('paint') for drawing shapes, sets color, stroke width & style.

4. Drawing Methods

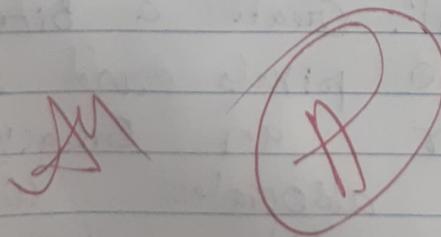
'draw3DCube', 'draw3Dcuboid', 'draw3Dcylinder' and 'draw3Dcone' are private methods for drawing 3D shapes on canvas. Each method takes parameters specifying the position and dimension of the shape, along with a Paint object for styling. The shapes are drawn by combining rectangles, ellipses and connecting lines, giving a simple 3D appearance.

5. Method Usage:

In 'onCreate', these methods are called with specific parameters to draw 3D cube, cuboid, cylinder and cone on the canvas.

Conclusion

This program essentially creates a basic 3D Shapes drawing app in Android, illustrating the use of Canvas, Bitmap, and Paint for graphical rendering.



Experiment 2

Aim: To draw Basic graphical 3D primitives.

Java Program:

```
package com.example.c31_56;

import androidx.appcompat.app.AppCompatActivity;
import android.graphics.Bitmap;
import android.graphics.Canvas;
import android.graphics.Color;
import android.graphics.Paint;
import android.os.Bundle;
import android.widget.ImageView;

public class MainActivity extends AppCompatActivity {

    Bitmap bg;
    ImageView img;

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

        // create bitmap
        bg = Bitmap.createBitmap(720, 1280, Bitmap.Config.RGB_565);

        img = findViewById(R.id.imageView);
        img.setImageBitmap(bg);

        // canvas creation
        Canvas canvas = new Canvas(bg);
```

Niyati_Savant_TE_C31_2103156

```
canvas.drawColor(Color.BLACK);

// paint object creation
Paint paint1 = new Paint();
paint1.setColor(Color.YELLOW);
paint1.setStrokeWidth(5); // Set the line thickness
paint1.setStyle(Paint.Style.STROKE); // Set to stroke (no fill)

// draw 3D cube
draw3DCube(canvas, 100, 200, 250, 350, paint1);

// draw 3D pyramid
Paint paint2 = new Paint();
paint2.setColor(Color.CYAN);
paint2.setStrokeWidth(5);
paint2.setStyle(Paint.Style.STROKE);

draw3DStriaghtPyramid(canvas, 400, 200, 550, 350, paint2);

// draw 3D cuboid
Paint paint3 = new Paint();
paint3.setColor(Color.GREEN);
paint3.setStrokeWidth(5);
paint3.setStyle(Paint.Style.STROKE);

draw3DCuboid(canvas, 100, 500, 250, 700, paint3);

// draw 3D cylinder
Paint paint4 = new Paint();
paint4.setColor(Color.MAGENTA);
paint4.setStrokeWidth(5);
paint4.setStyle(Paint.Style.STROKE);
```

```
draw3DCylinder(canvas, 400, 500, 550, 700, paint4);

}

private void draw3DCube(Canvas canvas, float left, float top, float right, float bottom, Paint paint)
{
    // Draw front face border
    canvas.drawRect(left, top, right, bottom, paint);

    // Draw back face border
    canvas.drawRect(left + 50, top + 50, right + 50, bottom + 50, paint);

    // Draw connecting lines for perspective
    canvas.drawLine(left, top, left + 50, top + 50, paint);
    canvas.drawLine(right, top, right + 50, top + 50, paint);
    canvas.drawLine(left, bottom, left + 50, bottom + 50, paint);
    canvas.drawLine(right, bottom, right + 50, bottom + 50, paint);
}

private void draw3DStriaghtPyramid(Canvas canvas, float left, float top, float right, float bottom,
Paint paint) {
    // Draw base rectangle
    canvas.drawRect(left, bottom - 50, right, bottom, paint);

    // Draw lines connecting apex to base corners
    canvas.drawLine(left, bottom - 50, (left + right) / 2, top, paint);
    canvas.drawLine(right, bottom - 50, (left + right) / 2, top, paint);
    canvas.drawLine(left, bottom - 50, right, bottom - 50, paint);
    canvas.drawLine(left, bottom - 50, (left + right) / 2, top, paint);
    canvas.drawLine(right, bottom - 50, (left + right) / 2, top, paint);
}
```

Niyati_Savant_TE_C31_2103156

```
private void draw3DCuboid(Canvas canvas, float left, float top, float right, float bottom, Paint
paint) {

    // Draw front face border
    canvas.drawRect(left, top, right, bottom, paint);

    // Draw back face border
    canvas.drawRect(left + 50, top - 50, right + 50, bottom - 50, paint);

    // Draw connecting lines for perspective
    canvas.drawLine(left, top, left + 50, top - 50, paint);
    canvas.drawLine(right, top, right + 50, top - 50, paint);
    canvas.drawLine(left, bottom, left + 50, bottom - 50, paint);
    canvas.drawLine(right, bottom, right + 50, bottom - 50, paint);
}

private void draw3DCylinder(Canvas canvas, float left, float top, float right, float bottom, Paint
paint) {

    // Draw top ellipse
    canvas.drawOval(left, top, right, top + 50, paint);

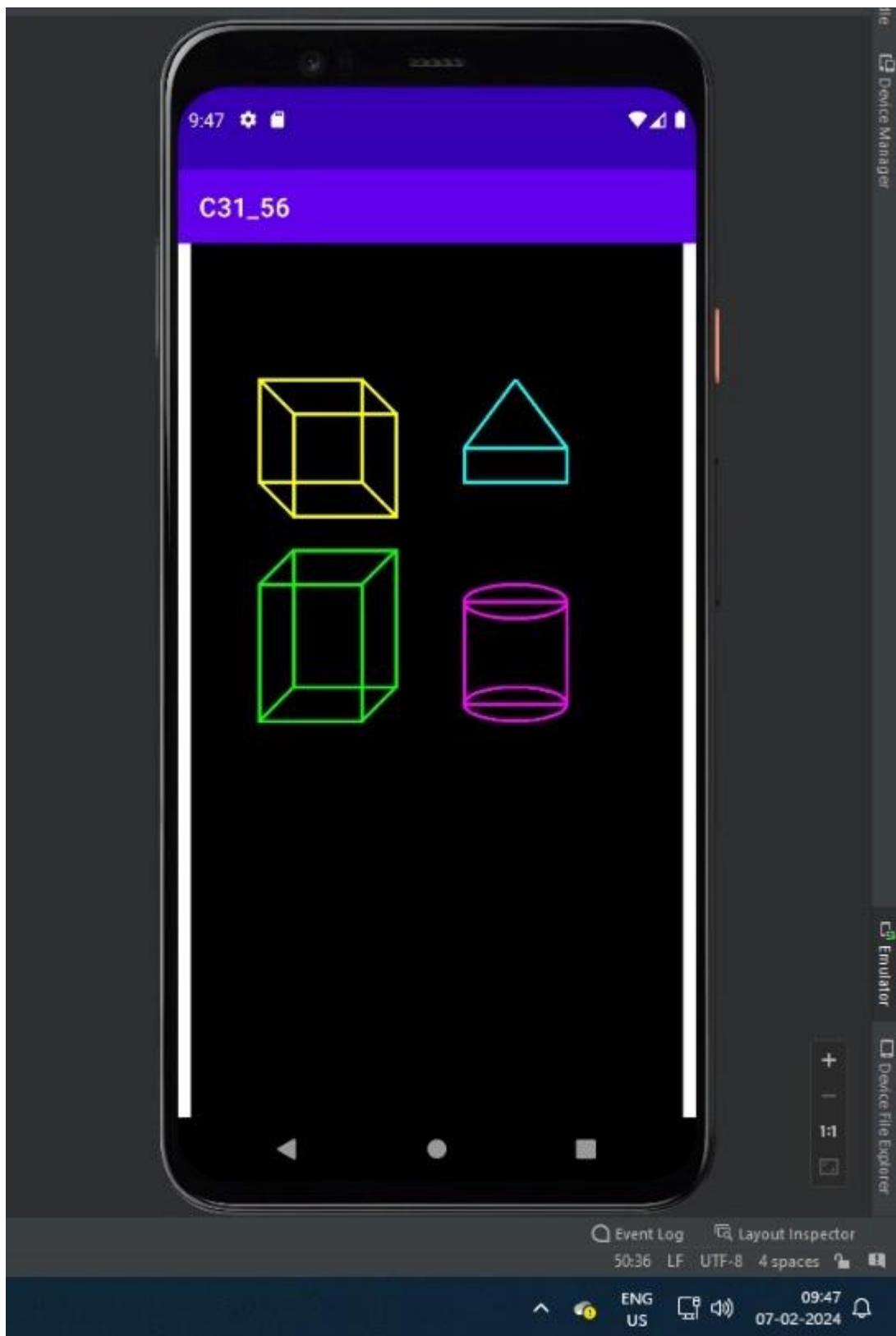
    // Draw bottom ellipse
    canvas.drawOval(left, bottom - 50, right, bottom, paint);

    // Draw connecting lines
    canvas.drawLine(left, top + 25, left, bottom - 25, paint);
    canvas.drawLine(right, top + 25, right, bottom - 25, paint);
    canvas.drawLine(left, top + 25, right, top + 25, paint);
    canvas.drawLine(left, bottom - 25, right, bottom - 25, paint);
}
```

XML program:

```
<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
    xmlns:app="http://schemas.android.com/apk/res-auto"  
    xmlns:tools="http://schemas.android.com/tools"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    tools:context=".MainActivity">  
  
    <ImageView  
        android:layout_width="match_parent"  
        android:layout_height="match_parent"  
        android:id="@+id/imageView"/>  
    </LinearLayout>
```

Output:



2103156 Experiment 3

Aim - Write Android Application to design a Form with GUI components.

Theory -

The Java program defines an Android activity that allows users to input their name, surname, age, and gender through EditText fields and RadioButtons. When submit is clicked, the program retrieves and values entered by user and logs them to console.

The following following functions are used -

1) onCreate(Bundle savedInstanceState)

This method is called when the activity is first created. It initializes the activity layout, retrieves references to the views (EditText fields, RadioGroup, RadioButtons, and Submit Button), and sets an onClickListener on the submit button.

2) findViewById(int id)

This method is used to find a view by its ID in the current layout. It is used to initialize all views by retrieving references to them.

3) setOnClickListener(View.OnClickListener listener)

Set an OnClickListener on Submit Button

It takes an instance of View.OnClickListener as a parameter. The OnClickListener defines the actions to be performed when button is clicked.

4) onClick(View view)

Called when Submit Button is clicked. It retrieves values entered by user from EditText fields and RadioButtons. It then logs these values to console for demonstration.

5) getText().toString()

Called to retrieve the string entered in an Edit Text field.

6) isChecked()

Used to check whether a RadioButton is checked or not.

7) RadioGroup

Used to group RadioButtons Together. It ensures that only 1 button in the group can be selected at a time.

8) RadioButton

Represents the choices in the form of a radio Button i.e. can select only one from each group.

Conclusion - I created a form which accepts Name, gender, college year, email and phone number from student.

Experiment 3

Aim: To design a Form with GUI Components .

Java Program:

```
package com.example.niyati_exp3_c31_2103156;

import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.RadioButton;
import android.widget.RadioGroup;
import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {

    private EditText editTextName, editTextSurname, editTextAge;
    private RadioGroup radioGroupGender;
    private RadioButton radioButtonMale, radioButtonFemale;
    private Button buttonSubmit;

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

        editTextName = findViewById(R.id.editTextName);
        editTextSurname = findViewById(R.id.editTextSurname);
        editTextAge = findViewById(R.id.editTextAge);
        radioGroupGender = findViewById(R.id.radioGroupGender);
        radioButtonMale = findViewById(R.id.radioButtonMale);
        radioButtonFemale = findViewById(R.id.radioButtonFemale);
```

Niyati_Savant_TE_C31_2103156

```
buttonSubmit = findViewById(R.id.buttonSubmit);

buttonSubmit.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        // Handle submit button click, you can retrieve values from the fields here
        String name = editTextName.getText().toString();
        String surname = editTextSurname.getText().toString();
        int age = Integer.parseInt(editTextAge.getText().toString());
        String gender = radioButtonMale.isChecked() ? "Male" : "Female";

        // Perform further actions or validation as needed
        // For simplicity, let's just log the values for demonstration
        System.out.println("Name: " + name + ", Surname: " + surname + ", Age: " + age + ",
Gender: " + gender);
    }
});
```

XML program:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="16dp"
    tools:context=".MainActivity">

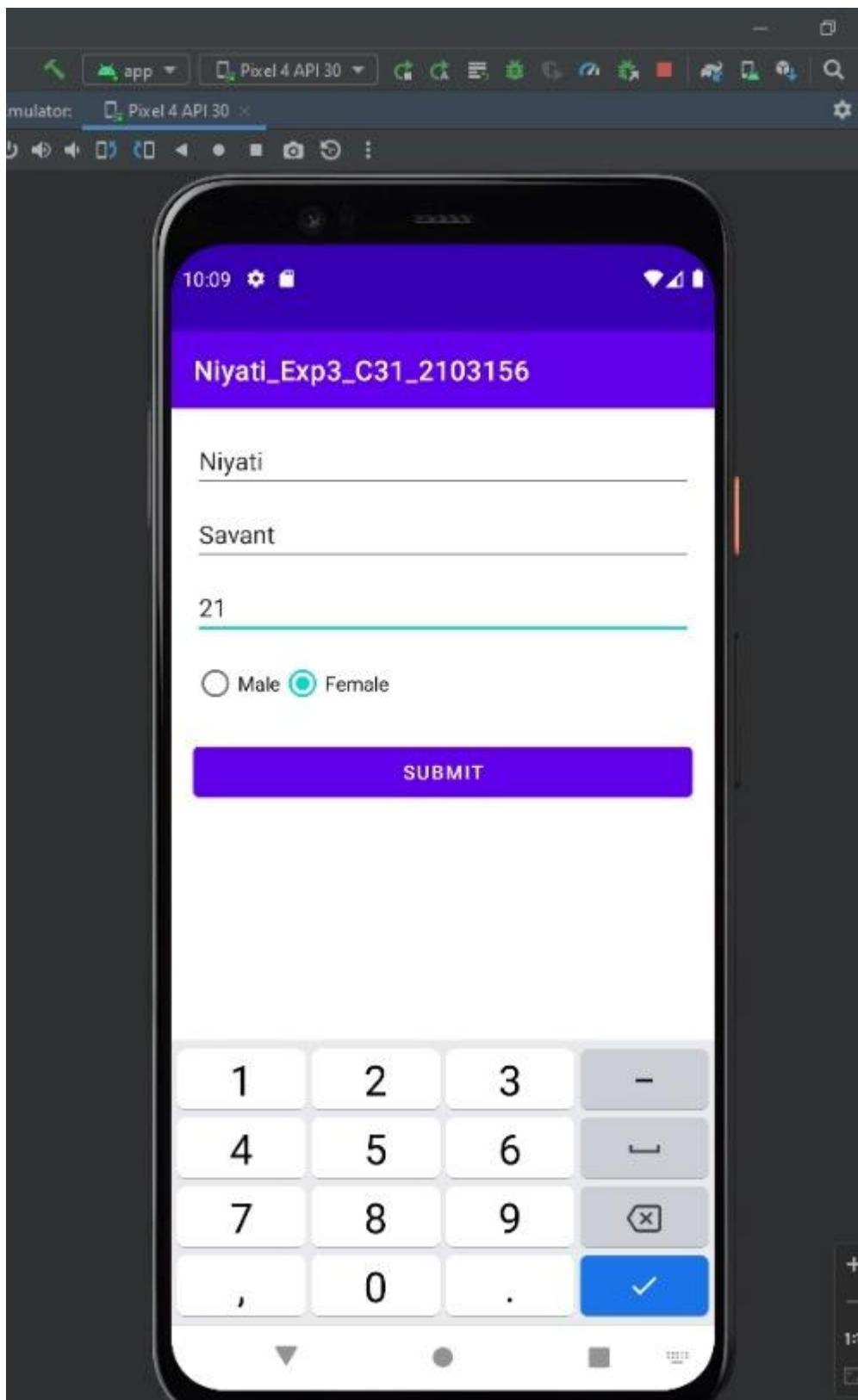
    <EditText
        android:id="@+id/editTextName"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Name"/>

    <EditText
        android:id="@+id/editTextSurname"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Surname"
        android:layout_marginTop="8dp"/>

    <EditText
        android:id="@+id/editTextAge"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Age"
        android:inputType="number"
        android:layout_marginTop="8dp"/>
```

```
<RadioGroup  
    android:id="@+id/radioGroupGender"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:orientation="horizontal"  
    android:layout_marginTop="8dp">  
  
<RadioButton  
    android:id="@+id radioButtonMale"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="Male"/>  
  
<RadioButton  
    android:id="@+id radioButtonFemale"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="Female"/>  
</RadioGroup>  
  
<Button  
    android:id="@+id/buttonSubmit"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:text="Submit"  
    android:layout_marginTop="16dp"/>  
  
</LinearLayout>
```

Output:



Aim : Write an Android Application to design GUI Components using Database

Theory:

Database in Android:

In Android, we make use of a different version of SQL databases. The database used is SQLite the most preferred one. It is the backbone whether its used directly or via some third party wrapper. It is a relational database containing tables, indexes. Schema is built to create our own tables.

Android has features available to handle changing database schemas which mostly depend on using SQLiteOpenHelper class and is designed to get rid of below 2 problems.

- i) When app runs for first time - When there is no database and so table, index must be created
- ii) When app is upgraded to newer schema - we have option to alter the database schema from older edition to match needs of new app. It has ..

i) Constructor

That takes Context (eg an Activity), name of database, optional Cursor factory and integer representing version of database schema you are using

onCreate (SQLiteDatabase db)

It is called when there is no database and app needs one. It passes an object, pointing to a newly created database that we can populate with tables and initial data.

onUpgrade (SQLiteDatabase db, int oldV, int newV)

It is called when schema version we need does not match schema version of database and finds best way to convert database from old to new.

The DBManager class is used for CRUD operations.

First the database connection is opened using ~~getWritableDatabase()~~ getReadableDatabase(). Connection is closed using dbHelper.close()

To insert values, we use ContentValues. put (DbHelper.DESC); and database.insert (DbHelper.Table_Name, null, value).

Similarly to update we use database.update (Data, Table_name, contentValues, DBHelper._ID + "=" + id, null);

Similarly to delete, we need to pass id of record to be deleted.

~~The Cursor represents result set of query.~~

Once query is fetched call to cursor.moveToFirst() made that allows us to test whether query returned an empty set and moves cursor to final result.

Conclusion - I have connected front end to an

SQL database and input data is locally stored

Experiment 4

Aim: To design GUI Components using Database.

Java Program:

```
package com.example.database_operations;
import androidx.appcompat.app.AppCompatActivity;
import android.content.Context;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.RadioGroup;
import android.widget ScrollView;
import android.widget.Toast;
import com.google.android.material.textfield.TextInputEditText;
import com.google.android.material.textfield.TextInputLayout;
public class FirstActivity extends AppCompatActivity {
    Context context;
    DatabaseHelper databaseHelper;
    ScrollView svRegistrationForm;
    16
    TextInputLayout tilName, tilEmail, tilContactNo;
    TextInputEditText tietName, tietEmail, tietContactNo;
    RadioGroup rgGender;
    Button btnSignUp, btnViewUsers;
    String name, email, contactNo, gender;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_first);
        context = FirstActivity.this;
        databaseHelper = new DatabaseHelper(context);
```

Niyati_Savant_TE_C31_2103156

```
svRegistrationForm = findViewById(R.id.sv_registration_form);
tilName = findViewById(R.id.til_name);
tilEmail = findViewById(R.id.til_email);
tilContactNo = findViewById(R.id.til_contact_no);
tietName = findViewById(R.id.tiet_name);
tietEmail = findViewById(R.id.tiet_email);
tietContactNo = findViewById(R.id.tiet_contact_no);
rgGender = findViewById(R.id.rg_gender);
btnSignUp = findViewById(R.id.btn_insert_data);
btnViewUsers = findViewById(R.id.btn_view_users);
rgGender.setOnCheckedChangeListener(new
RadioGroup.OnCheckedChangeListener() {
    @Override
    public void onCheckedChanged(RadioGroup group, int checkedId) {
        if(checkedId == R.id.rb_male) {
            gender = "Male";
        } else {
            gender = "Female";
        }
    }
});
btnSignUp.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if(validateForm()) {
            name = tietName.getText().toString();
            email = tietEmail.getText().toString();
            contactNo = tietContactNo.getText().toString();
            boolean insertResult = databaseHelper.insertUserDetails(email, name,
            contactNo,
            gender);
            if(insertResult) {
```

Niyati_Savant_TE_C31_2103156

```
Toast.makeText(context, "Data inserted successfully",
Toast.LENGTH_SHORT).show();
} else {
17
Toast.makeText(context, "Data not inserted",
Toast.LENGTH_SHORT).show();
}
Intent intent = new Intent(context, FirstActivity.class);
startActivity(intent);
}
}
});
btnViewUsers.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
Intent intent = new Intent(context, SecondActivity.class);
startActivity(intent);
}
});
}
public boolean validateForm() {
if(tietName.getText().toString().isEmpty()) {
tilName.setError("Enter Name");
focusOnView(tilName);
return false;
} else if(tietEmail.getText().toString().isEmpty()) {
tilEmail.setError("Enter Email");
focusOnView(tilEmail);
return false;
} else if(tietContactNo.getText().toString().isEmpty()) {
tilContactNo.setError("Enter Contact no");
focusOnView(tilContactNo);
}
}
```

Niyati_Savant_TE_C31_2103156

```
return false;
} else if(rgGender.getCheckedRadioButtonId() == -1) {
    Toast.makeText(context, "Select Gender", Toast.LENGTH_SHORT).show();
    focusOnView(rgGender);
    return false;
} else {
    return true;
}
}

public void focusOnView(final View view) {
    svRegistrationForm.post(new Runnable() {
        @Override
        public void run() {
            svRegistrationForm.smoothScrollTo(0, view.getTop()-50);
        }
    });
}
}
```

```
package com.example.database_operations;
import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;
import android.content.Context;
import android.content.Intent;
import android.database.Cursor;
import android.os.Bundle;
import android.view.View;
import android.widget.ListView;
import android.widget.TextView;
import java.util.ArrayList;
public class SecondActivity extends AppCompatActivity {
    Context context;
    DatabaseHelper databaseHelper;
    ArrayList<ModelForUsers> userList = new ArrayList();
    RecyclerView rvUserDetails;
    RecyclerView.Adapter adapter;
    TextView tvMessage;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_second);
        context = SecondActivity.this;
        databaseHelper = new DatabaseHelper(context);
        rvUserDetails = findViewById(R.id.rv_user_details);
        tvMessage = findViewById(R.id.tv_message);
        Cursor cursor = databaseHelper.getAllUserDetails();
        if(cursor.getCount() == 0) {
            tvMessage.setVisibility(View.VISIBLE);
        } else {
```

Niyati_Savant_TE_C31_2103156

```
tvMessage.setVisibility(View.GONE);
while (cursor.moveToFirst()) {
    ModelForUsers user = new ModelForUsers();
    user.setId(cursor.getString(0));
    user.setName(cursor.getString(1));
    user.setEmail(cursor.getString(2));
    user.setContactNo(cursor.getString(3));
    user.setGender(cursor.getString(4));
    userList.add(user);
}
rvUserDetails.setHasFixedSize(true);
rvUserDetails.setLayoutManager(new LinearLayoutManager(this));
adapter = new UserRecyclerViewAdapter(context, userList)
rvUserDetails.setAdapter(adapter);
}
```

Program (DatabaseHelper.java): -

```
package com.example.database_operations;
import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.MatrixCursor;
import android.database.SQLException;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
import android.util.Log;
import androidx.annotation.Nullable;
import java.util.ArrayList;
public class DatabaseHelper extends SQLiteOpenHelper {
    public static final String DATABASE_NAME = "Music App";
```

Niyati_Savant_TE_C31_2103156

```
public static final String TABLE_NAME = "Users";
// COL_0 will be user Id which is AUTOINCREMENT
public static final String COL_1 = "Email";
public static final String COL_2 = "Name";
public static final String COL_3= "Contact_No";
public static final String COL_4 = "Gender";
public DatabaseHelper(@Nullable Context context) {
    super(context, DATABASE_NAME, null, 1);
}
@Override
public void onCreate(SQLiteDatabase db) {
    db.execSQL("create table Users (Id INTEGER PRIMARY KEY
    AUTOINCREMENT,Name TEXT, Email TEXT,Contact_No TEXT, Gender TEXT)");
}
@Override
public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
    db.execSQL("DROP TABLE IF EXISTS Users");
    onCreate(db);
}
// method for inserting data of a single user
public boolean insertUserDetails(String email, String name, String contactNo,
String gender) {
    SQLiteDatabase db = this.getWritableDatabase();
    ContentValues contentValues = new ContentValues();
    contentValues.put(COL_1, email);
    contentValues.put(COL_2, name);
    contentValues.put(COL_3, contactNo);
    contentValues.put(COL_4, gender);
    long insertResult = db.insert(TABLE_NAME, null, contentValues);
    if(insertResult == -1) {
        return false;
    } else {
```

Niyati_Savant_TE_C31_2103156

```
return true;
}
}

// method for get data of all users

public Cursor getAllUserDetails() {
    SQLiteDatabase db = this.getWritableDatabase();
    Cursor cursor = db.rawQuery("SELECT * FROM " + TABLE_NAME, null);
    return cursor;
}
}
```

Program (ModelForUsers.java): -

```
package com.example.database_operations;

public class ModelForUsers {

    String id, name, email, contactNo, gender;

    public String getId() {
        return id;
    }

    public void setId(String id) {
        this.id = id;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getEmail() {
        return email;
    }

    public void setEmail(String email) {
        this.email = email;
    }
}
```

Niyati_Savant_TE_C31_2103156

```
public String getContactNo() {  
    return contactNo;  
}  
  
public void setContactNo(String contactNo) {  
    this.contactNo = contactNo;  
}  
  
public String getGender() {  
    return gender;  
}  
  
public void setGender(String gender) {  
    this.gender = gender;  
}
```

Program (UserRecyclerViewAdapter.java): -

```
package com.example.database_operations;  
  
import android.content.Context;  
  
import android.view.LayoutInflater;  
  
import android.view.View;  
  
import android.view.ViewGroup;  
  
import android.widget.TextView;  
  
import androidx.annotation.NonNull;  
  
import androidx.recyclerview.widget.RecyclerView;  
  
import java.util.ArrayList;  
  
public class UserRecyclerViewAdapter extends  
RecyclerView.Adapter<UserRecyclerViewAdapter.UserRecyclerViewHolder> {  
  
    Context context;  
  
    ArrayList<ModelForUsers> usersList;  
  
    public UserRecyclerViewAdapter(Context context, ArrayList<ModelForUsers>  
usersList) {  
  
        this.context = context;  
  
        this.usersList = usersList;  
    }
```

Niyati_Savant_TE_C31_2103156

```
@NonNull  
@Override  
public UserRecyclerViewHolder onCreateViewHolder(@NonNull ViewGroup  
parent, int viewType) {  
    LayoutInflater layoutInflater = LayoutInflater.from(parent.getContext());  
    View view = layoutInflater.inflate(R.layout.row_users, parent, false);  
    UserRecyclerViewHolder userRecyclerViewHolder = new  
    UserRecyclerViewHolder(view);  
    return userRecyclerViewHolder;  
}  
  
@Override  
public void onBindViewHolder(@NonNull UserRecyclerViewHolder  
userRecyclerViewHolder, int position) {  
    userRecyclerViewHolder.tvId.setText("Id: " + usersList.get(position).getId());  
    userRecyclerViewHolder.tvName.setText("Name: " +  
    usersList.get(position).getName());  
    userRecyclerViewHolder.tvEmail.setText("Email: " +  
    usersList.get(position).getEmail());  
    userRecyclerViewHolder.tvContactNo.setText("Contact No: " +  
    usersList.get(position).getContactNo());  
    userRecyclerViewHolder.tvGender.setText("Gender: " +  
    usersList.get(position).getGender());  
}  
  
@Override  
public int getItemCount() {  
    return usersList.size();  
}  
  
class UserRecyclerViewHolder extends RecyclerView.ViewHolder {  
    TextView tvId, tvName, tvEmail, tvContactNo, tvGender;  
    public UserRecyclerViewHolder(@NonNull View itemView) {  
        super(itemView);  
        tvId = itemView.findViewById(R.id.tv_id);
```

```
tvName = itemView.findViewById(R.id.tv_name);
tvEmail = itemView.findViewById(R.id.tv_email);
tvContactNo = itemView.findViewById(R.id.tv_contact_no);
tvGender = itemView.findViewById(R.id.tv_gender);
}}
```

XML program:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".FirstActivity">
    <ScrollView
        android:id="@+id/sv_registration_form"
        android:layout_width="match_parent"
        android:layout_height="wrap_content">
        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:layout_margin="10dp"
            android:orientation="vertical">
            <com.google.android.material.textfield.TextInputLayout
                android:id="@+id/til_name"
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:layout_marginTop="20dp">
                <com.google.android.material.textfield.TextInputEditText
                    android:id="@+id/tiet_name"
                    android:layout_width="match_parent"
```

Niyati_Savant_TE_C31_2103156

```
    android:layout_height="wrap_content"
    android:hint="Name" />
</com.google.android.material.textfield.TextInputLayout>
<com.google.android.material.textfield.TextInputLayout
    android:id="@+id/til_email"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_marginTop="20dp">
    <com.google.android.material.textfield.TextInputEditText
        android:id="@+id/tiet_email"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Email"
        android:inputType="textEmailAddress"/>
</com.google.android.material.textfield.TextInputLayout>
<com.google.android.material.textfield.TextInputLayout
    android:id="@+id/til_contact_no"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_marginTop="20dp">
    <com.google.android.material.textfield.TextInputEditText
        android:id="@+id/tiet_contact_no"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Contact No."
        android:inputType="number"
        android:maxLength="10"/>
</com.google.android.material.textfield.TextInputLayout>
<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Gender"
```

Niyati_Savant_TE_C31_2103156

```
    android:textSize="18sp"
    android:layout_marginTop="20dp"/>
<RadioGroup
    android:id="@+id/rg_gender"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" >
    <RadioButton
        android:id="@+id/rb_male"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Male"
        android:textSize="15sp"/>
    <RadioButton
        android:id="@+id/rb_female"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Female"
        android:textSize="15sp"/>
</RadioGroup>
<Button
    android:id="@+id	btn_insert_data"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Insert Data"
    android:textAllCaps="false"
    android:layout_gravity="center"
    android:layout_marginTop="20dp"/>
<Button
    android:id="@+id	btn_view_users"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="View Users"
```

Niyati_Savant_TE_C31_2103156

```
    android:textAllCaps="false"
    android:layout_gravity="center"
    android:layout_marginTop="20dp"/>
</LinearLayout>
</ScrollView>
</LinearLayout>

Program (activity_second.xml <?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".SecondActivity"
    android:background="#DADADA"
    android:orientation="vertical">
<TextView
    android:id="@+id/tv_message"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="#fffff"
    android:layout_margin="20dp"
    android:text="No user details"
    android:textSize="30dp"
    android:gravity="center"
    android:visibility="gone"/>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    android:layout_margin="20dp">
<androidx.recyclerview.widget.RecyclerView
```

Niyati_Savant_TE_C31_2103156

```
    android:id="@+id/rv_user_details"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" />
</LinearLayout>
</LinearLayout>
```

Program (row_users.xml): -

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="10dp"
    android:background="#ffffffff"
    android:padding="10dp">

    <TextView
        android:id="@+id/tv_id"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textSize="20sp"
        android:textColor="#000000"
        android:text="id"/>

    <TextView
        android:id="@+id/tv_name"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textSize="20sp"
        android:textColor="#000000"
        android:text="name"/>

    <TextView
        android:id="@+id/tv_email"
        android:layout_width="match_parent"
```

Niyati_Savant_TE_C31_2103156

```
    android:layout_height="wrap_content"
    android:textSize="20sp"
    android:textColor="#000000"
    android:layout_marginTop="5dp"
    android:text="email"/>
<TextView
    android:id="@+id/tv_contact_no"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:textSize="20sp"
    android:textColor="#000000"
    android:layout_marginTop="5dp"
    android:text="contact"/>
<TextView
    android:id="@+id/tv_gender"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:textSize="20sp"
    android:textColor="#000000"
    android:layout_marginTop="5dp"
    android:text="gender"/>
</LinearLayout>
```

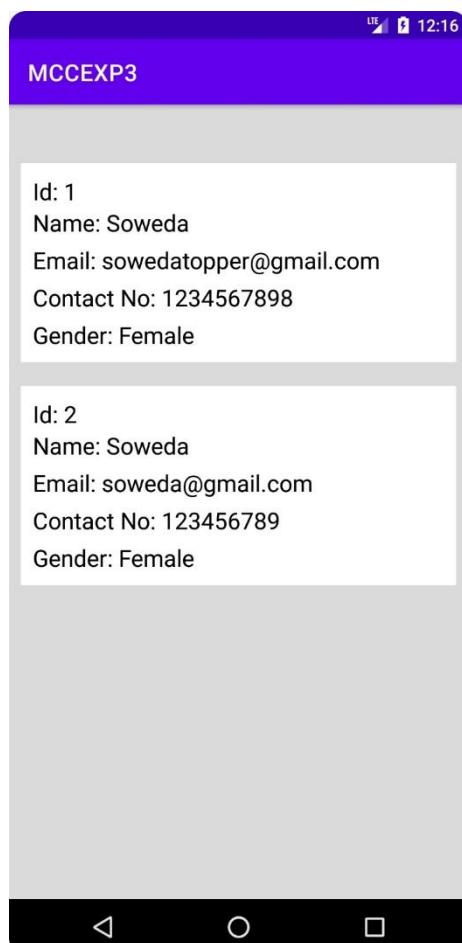
Program (AndroidManifest.xml): -

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.database_operations">
    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
```

Niyati_Savant_TE_C31_2103156

```
android:theme="@style/Theme.Database_Operations">
<activity
    android:name=".FirstActivity"
    android:exported="true">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
<activity
    android:name=".SecondActivity"
    android:exported="true">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
</application>
</manifest>
```

Output:



Experiment 5

Aim : Write Android application to design ~~GUI~~ components using Database EMI Calculator

Theory-

This Java Program represents an Android Application for calculating the Equated Monthly Installment (EMI) for a loan based on the principal amount, interest rate, and loan tenure.

In the onCreate method, various views are initialized by finding them using their respective ids in my layout xml file. These initialized views include EditText fields to enter principal amount (P), interest rate (R) and loan tenure (T) in years. Another text field is used to display result. Finally the emiCalcBtn is initialized to trigger EMI calculation.

OnClickListener for calculate Button is set on the emiCalcBtn using setOnClickListener method that defines what should happen when calculate Button is clicked.

onClick Method is inside the OnClickListener. The values entered by user in EditText fields for principal amount, interest rate, and loan tenure are retrieved as strings (st1, st2, st3)

2103156

TextUtils is used to check if any fields are empty. If so and error message is set and so can is requested on that field. If all fields are filled, string values are passed to `String types(p, isy)`. The necessary calculations for EMI are performed using methods.

- 1) calcPrinc - Return Principal amount
- 2) calInt : Calculate monthly interest rate
- 3) calMonth: Converts loan tenure from years to months
- 4) calDvdnt : calculate ~~final~~ divisor for calculation
- 5) calFinalDvdnt : calculate final dividend for calculation
- 6) calDivisor - Calculate divisor for EMI calculation
- 7) calEmi - Find EMI using the formula

$$\text{EMI} = \frac{(\text{Principal} * \text{Rate} * \text{Dvdnt})}{(\text{Dvdnt} - 1)}$$

Conclusion :

I have built an application that calculates EMI by taking principle amount, rate of interest % and time in months and $\text{Dvdnt} = (1 + x)^n$

MM A

Experiment 5

Aim: To develop an EMI Calculator application.

Java Program:

```
package com.example.emi_exp5;
import android.os.Bundle;
import androidx.appcompat.app.AppCompatActivity;
import android.text.TextUtils;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
public class MainActivity extends AppCompatActivity {
    Button emiCalcBtn;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        final EditText P = (EditText) findViewById(R.id.principal);
        final EditText I = (EditText) findViewById(R.id.interest);
        final EditText Y = (EditText) findViewById(R.id.years);
        final EditText result = (EditText) findViewById(R.id.emi);
        emiCalcBtn = (Button) findViewById(R.id.btn_calculate2);
        emiCalcBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String st1 = P.getText().toString();
                String st2 = I.getText().toString();
                String st3 = Y.getText().toString();
                if (TextUtils.isEmpty(st1)) {
                    P.setError("Enter Principal Amount");
                    P.requestFocus();
                }
                return;
            }
        });
    }
}
```

Niyati_Savant_TE_C31_2103156

```
        }

        if (TextUtils.isEmpty(st2)) {
            I.setError("Enter Interest Rate");
            I.requestFocus();
            return;
        }

        if (TextUtils.isEmpty(st3)) {
            Y.setError("Enter Years");
            Y.requestFocus();
            return;
        }

        float p = Float.parseFloat(st1);
        float i = Float.parseFloat(st2);
        float y = Float.parseFloat(st3);

        float Principal = calPric(p);
        float Rate = calInt(i);
        float Months = calMonth(y);
        float Dvdnt = calDvdnt( Rate, Months);
        float FD = calFinalDvdnt (Principal, Rate, Dvdnt);
        float D = calDivide(Dvdnt);
        float emi = calEmi(FD, D);
        result.setText(String.valueOf(emi));
    }
});
```

}

```
public float calPric(float p) {
    return (float) (p);
}

public float calInt(float i) {
    return (float) (i/12/100);
}

public float calMonth(float y) {
```

Niyati_Savant_TE_C31_2103156

```
return (float) (y * 12);  
}  
  
public float calDvdnt(float Rate, float Months) {  
    return (float) (Math.pow(1+Rate, Months));  
}  
  
public float calFinalDvdnt(float Principal, float Rate, float Dvdnt) {  
    return (float) (Principal * Rate * Dvdnt);  
}  
  
public float calDivider(float Dvdnt) {  
    return (float) (Dvdnt-1);  
}  
  
public float calEmi(float FD, Float D) {  
    return (float) (FD/D);  
}  
}
```

XML program:

```
<androidx.coordinatorlayout.widget.CoordinatorLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    xmlns:tools="http://schemas.android.com/tools"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    tools:context="com.example.emi_exp5.MainActivity"
    android:layout_height="match_parent"
    android:background="@color/black"
    android:backgroundTint="@color/black">
    <androidx.core.widget.NestedScrollView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        app:layout_behavior="@string/appbar_scrolling_view_behavior">
        <LinearLayout
            android:layout_width="fill_parent"
            android:layout_height="match_parent"
            android:layout_marginTop="?attr/actionBarSize"
            android:orientation="vertical"
            android:paddingLeft="20dp"
            android:paddingRight="20dp"
            android:paddingTop="10dp">
            <com.google.android.material.textfield.TextInputLayout
                android:id="@+id/input_layout_principal"
                android:layout_width="match_parent"
                android:layout_height="wrap_content">
                <EditText
                    android:id="@+id/principal"
                    android:layout_width="match_parent"
                    android:layout_height="wrap_content"
                    android:singleLine="true"/>
            
```

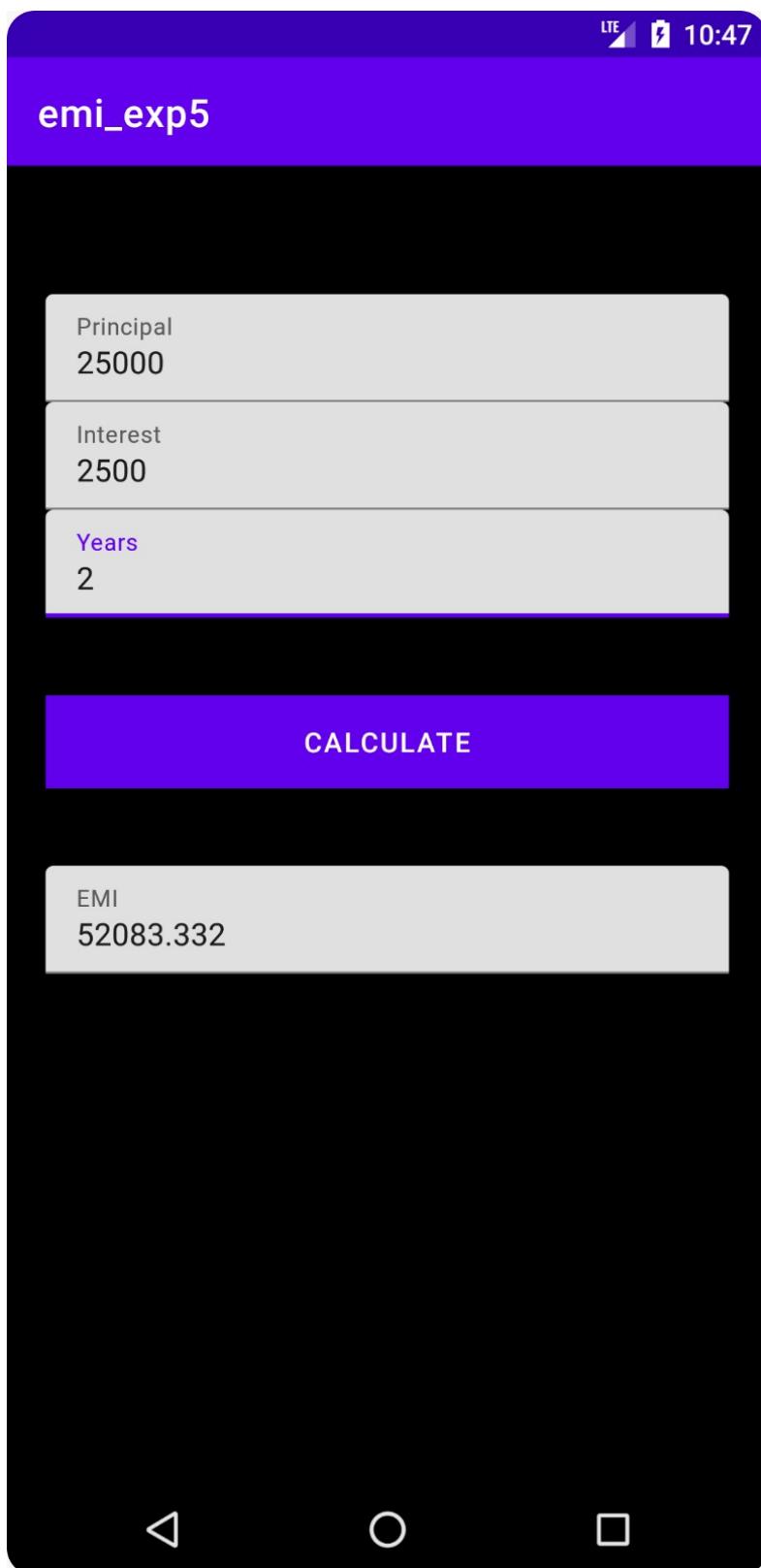
Niyati_Savant_TE_C31_2103156

```
    android:inputType="number"
    android:digits="0123456789."
    android:hint="Principal" />
</com.google.android.material.textfield.TextInputLayout>
<com.google.android.material.textfield.TextInputLayout
    android:id="@+id/input_layout_interest"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">
    <EditText android:id="@+id/interest"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:singleLine="true"
        android:inputType="number"
        android:digits="0123456789."
        android:hint="Interest" />
</com.google.android.material.textfield.TextInputLayout>
<com.google.android.material.textfield.TextInputLayout
    android:id="@+id/input_layout_tenure"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">
    <EditText
        android:id="@+id/years"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:inputType="number"
        android:digits="0123456789."
        android:hint="Years" />
</com.google.android.material.textfield.TextInputLayout>
<Button android:id="@+id/btn_calculate2"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="Calculate"
```

Niyati_Savant_TE_C31_2103156

```
        android:background="#000000"
        android:layout_marginTop="40dp"
        android:textColor="#FFFFFF"/>
<com.google.android.material.textfield.TextInputLayout
    android:id="@+id/input_layout_emi"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="40dp">
    <EditText android:id="@+id/emi"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:maxEms="0"
        android:inputType="number"
        android:hint="EMI" />
</com.google.android.material.textfield.TextInputLayout>
</LinearLayout>
</androidx.core.widget.NestedScrollView>
</androidx.coordinatorlayout.widget.CoordinatorLayout>
```

Output:



Experiment 6

Aim - Write an Android Application to Create an alert on receiving a message.

Theory -

A push notification is notification that appears in the notification bar of the android phone and used to alert user about something in the app or any information the app might want to communicate to the user.

The Java program creates a simple Android app that sends a push notification when a button is clicked.

- It defines a layout with an EditText field for entering a message and a Button to trigger notification.
- In MainActivity class, it initializes the button and EditText field in the onCreate method.
- It sets an OnClickListener for the button to handle notification creation when clicked.
- Inside the onClick method, it retrieves message from the Edit Text field.
- It creates an Intent to launch the MainActivity when notification is clicked and creates a PendingIntent from the Intent.
- It constructs a Notification Channel for devices running specified version and constructs Notification object with provided message and PendingIntent.
- It retrieves Notification Manager system service.
- It notifies user with constructed notification.

- Further, it sets the Auto-Cancel flag to automatically dismiss notification when clicked
- It also clears EditText field after sending the notification

The 'OnClickListener' in Android is an interface used to handle click events on views such as buttons. It requires implementing 'onClick' method where you define actions to be taken when view is clicked.

The 'setOnItemClickListener' is used to get an instance of 'OnClickListener' on the 'btnNotify' button. When clicked it invokes 'onClick' method and triggers notification creation process. In-built functions used -

- 1) findViewById() - Finds a view by its Id within current layout and used to initialize 'btnNotify' and 'etMessage'.
- 2) getSystemService() - Retrieve system service with specified name.
- 3) setContentTitle() - Sets title of notification.
- 4) setContentText() - Sets text content of notification.
- 5) setSmallIcon() - Sets the small icon to be displayed in notification bar.
- 6) setContentIntent() - Sets PendingIntent to be sent when notification is clicked.
- 7) setAutoCancel() - Sets whether notification should be automatically canceled by user click.
- 8) notify() - Notifies user with notification.

Conclusion: A push notification is implemented using notification class.

Experiment 6

Aim: To create an alert on receiving a message.

Java Program:

```
package com.example.messagealert;
import androidx.annotation.RequiresApi;
import androidx.appcompat.app.AppCompatActivity;
import android.app.Notification;
import android.app.NotificationChannel;
import android.app.NotificationManager;
import android.app.PendingIntent;
import android.content.Context;
import android.content.Intent;
import android.os.Build;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
public class MainActivity extends AppCompatActivity {
    Button btnNotify;
    EditText etMessage;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        btnNotify = findViewById(R.id.btn_notify);
        etMessage = findViewById(R.id.et_message);
        btnNotify.setOnClickListener(new View.OnClickListener() {
            @RequiresApi(api = Build.VERSION_CODES.O)
            @Override
            public void onClick(View v) {
                String message = etMessage.getText().toString();
            }
        });
    }
}
```

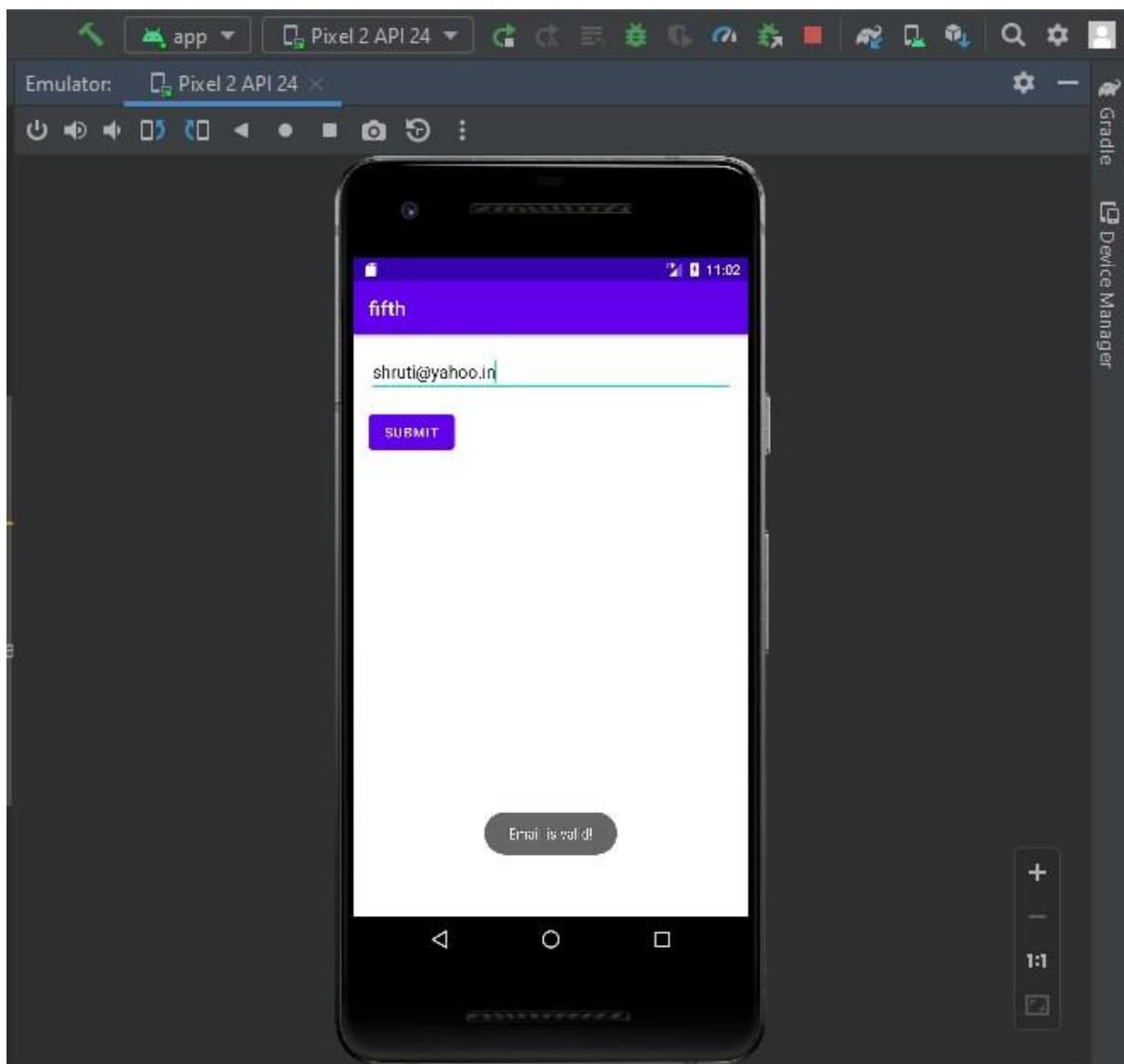
Niyati_Savant_TE_C31_2103156

```
Intent intent = new Intent(MainActivity.this, MainActivity.class);
PendingIntent pendingIntent = PendingIntent.getActivity(MainActivity.this,
0, intent, 0);
//Notification notification = new
Notification.Builder(MainActivity.this).setContentTitle("New
Message").setContentText(etMessage.getText().toString()).setSmallIcon(R.mipmap.ic
_launcher).setContentIntent(pendingIntent).build();
NotificationManager mNotificationManager = (NotificationManager)
getSystemService(Context.NOTIFICATION_SERVICE);
String id = "243";
int importance = NotificationManager.IMPORTANCE_LOW;
CharSequence name="";
NotificationChannel mChannel = new NotificationChannel(id,
name,importance);
mChannel.enableLights(true);
mNotificationManager.createNotificationChannel(mChannel);
Notification notification = new Notification.Builder(MainActivity.this ,
id).setContentTitle("Title").setContentText(etMessage.getText().toString()).setSmallI
con(R.mipmap.ic_launcher).setContentIntent(pendingIntent).build();
mNotificationManager.notify(Integer.parseInt(id), notification);
notification.flags |= Notification.FLAG_AUTO_CANCEL;
mNotificationManager.notify(0, notification);
etMessage.setText("");
}
});
}
}
```

XML program:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".MainActivity">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Message"
        android:textSize="30sp" />
    <EditText
        android:id="@+id/et_message"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:singleLine="true"
        android:textColor="#000000"
        android:textSize="30sp" />
    <Button
        android:id="@+id/btn_notify"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:layout_margin="30dp"
        android:text="Notify"
        android:textSize="30sp" />
</LinearLayout>
```

Output:



Aim : Write Android Application to implement Basic calculator using Android

Theory :

The Java program represents an Android application that performs basic arithmetic operations (addition, subtraction, multiplication, and division) on two numbers entered by user and displays the result.

a] Initialization of Views

In the 'onCreate' method, various views are initialized by finding them using their respective IDs in the layout XML file. These include

- 'etFirstNumber', 'etSecondNumber' - Edit Text fields where user can input first and second numbers
- 'tvResult' - TextView where result is displayed
- 'btnAdd', 'btnSubtract', 'btnMultiply', 'btnDivide', 'btnClear' - Buttons for performing addition, subtraction, division, multiplication and clearing input fields and result.

b] OnClicklistener for Buttons

On each operation button ('btnAdd', 'btnSubtract', 'btnMultiply', 'btnDivide'), an 'onClickListener' is set to perform corresponding arithmetic operation when clicked similarly, on the clear button ('btnClear'), an 'onClickListener' is set to clear the input fields and

result when clicked.

c) onClick Method

Inside this method of each OnClick Listener values entered by user in EditText fields for 1st and 2nd numbers are received as strings. which are converted to integers using Integer.parseInt() for arithmetic operations, the operation is performed and result is calculated. It is displayed in 'Result' using setText().

d) Clear Button Functionality

The clear button ('btnClear') onClick method clears the text in input fields ('etFirstNumber', 'etSecondNumber') and sets result TextView ('tvResult') to an empty string.

Conclusion - Using GridLayout, I have arranged the keypad for calculation and calculation logic has been implemented in Main Activity.java

AM **A**

Experiment 7

Aim: To implement Basic Calculator using Android.

Java Program:

```
package com.example.calculator_exp7;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import androidx.appcompat.app.AppCompatActivity;
public class MainActivity extends AppCompatActivity {
    EditText etFirstNumber, etSecondNumber;
    TextView tvResult;
    Button btnAdd, btnSubtract, btnMultiply, btnDivide, btnClear;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        etFirstNumber = findViewById(R.id.et_first_number);
        etSecondNumber = findViewById(R.id.et_second_number);
        tvResult = findViewById(R.id.tv_result);
        btnAdd = findViewById(R.id.btn_add);
        btnSubtract = findViewById(R.id.btn_subtract);
        btnMultiply = findViewById(R.id.btn_multiply);
        btnDivide = findViewById(R.id.btn_divide);
        btnClear = findViewById(R.id.btn_clear);
        btnAdd.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                int firstNumber = Integer.parseInt(etFirstNumber.getText().toString());
                int secondNumber =

```

Niyati_Savant_TE_C31_2103156

```
Integer.parseInt(etSecondNumber.getText().toString());
int result = firstNumber + secondNumber;
tvResult.setText(result+" ");
}

});

btnSubtract.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
int firstNumber = Integer.parseInt(etFirstNumber.getText().toString());
int secondNumber =
Integer.parseInt(etSecondNumber.getText().toString());
int result = firstNumber - secondNumber;
tvResult.setText(result+" ");
}
});

btnMultiply.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
int firstNumber = Integer.parseInt(etFirstNumber.getText().toString());
int secondNumber =
Integer.parseInt(etSecondNumber.getText().toString());
int result = firstNumber * secondNumber;
tvResult.setText(result+" ");
}
});

btnDivide.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
int firstNumber = Integer.parseInt(etFirstNumber.getText().toString());
int secondNumber =
Integer.parseInt(etSecondNumber.getText().toString());
int result = firstNumber / secondNumber;
tvResult.setText(result+" ");
}
});
```

Niyati_Savant_TE_C31_2103156

```
tvResult.setText(result+" ");
}

});

btnClear.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
etFirstNumber.setText("");
etSecondNumber.setText("");
tvResult.setText("");
}
});
```

XML program:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns: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/et_first_number"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:ems="10"
        android:inputType="number"
        android:layout_marginTop="50dp"
        android:hint="First Number"/>
    <EditText
        android:id="@+id/et_second_number"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:ems="10"
        android:inputType="number"
        android:layout_marginTop="50dp"
        android:hint="Second Number"/>
    <TextView
        android:id="@+id/tv_result"
        android:layout_width="match_parent"
        android:layout_height="30dp"
        android:textSize="20sp"
        android:textColor="#000000"
        android:text=""
        android:layout_marginTop="50dp"
```

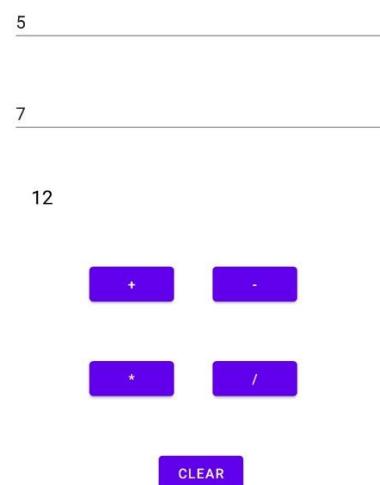
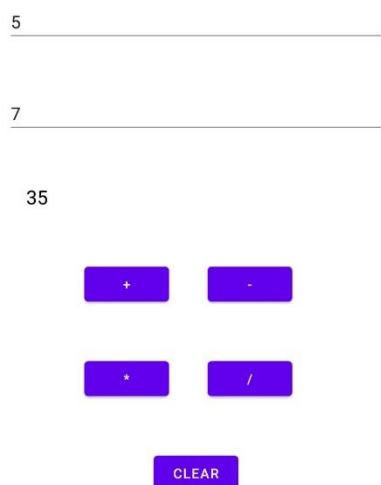
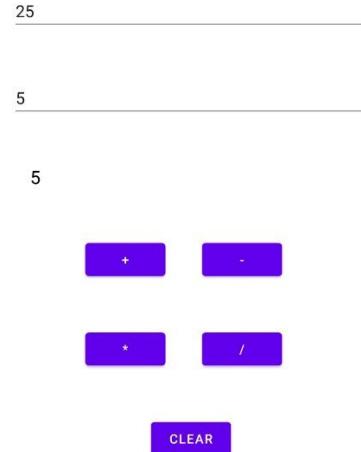
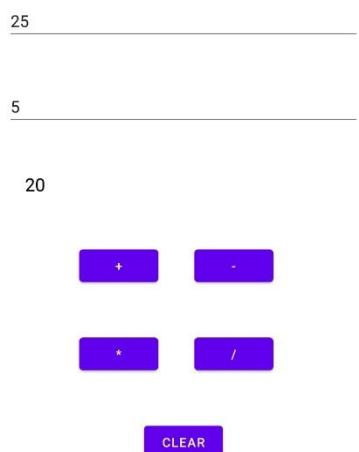
Niyati_Savant_TE_C31_2103156

```
    android:paddingLeft="20dp"/>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="50dp"
    android:layout_marginRight="50dp"
    android:layout_marginLeft="50dp">
    <Button
        android:id="@+id	btn_add"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="+"
        android:layout_marginLeft="30dp"
        android:layout_marginRight="20dp"/>
    <Button
        android:id="@+id	btn_subtract"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="-"
        android:layout_marginLeft="20dp"
        android:layout_marginRight="20dp"/>
</LinearLayout>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="50dp"
    android:layout_marginRight="50dp"
    android:layout_marginLeft="50dp">
    <Button
        android:id="@+id	btn_multiply"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
```

Niyati_Savant_TE_C31_2103156

```
    android:text="*"  
    android:layout_marginLeft="30dp"  
    android:layout_marginRight="20dp"/>  
  
    <Button  
        android:id="@+id	btn_divide"  
        android:layout_width="wrap_content"  
        android:layout_height="wrap_content"  
        android:text="/"  
        android:layout_marginRight="20dp"  
        android:layout_marginLeft="20dp"/>  
    </LinearLayout>  
  
    <Button  
        android:id="@+id	btn_clear"  
        android:layout_width="wrap_content"  
        android:layout_height="wrap_content"  
        android:text="clear"  
        android:layout_marginTop="50dp"  
        android:layout_gravity="center"/>  
    </LinearLayout>
```

Output:



Aim - Write a program to demonstrate cellular Frequency reuse

Theory

Frequency reuse is a scheme in which allocation and reuse of channels throughout a coverage region is done. Each cellular base station is allocated a group of radio channels or Frequency sub-bands to be used within a small geographic area called cell of hexagonal shape.

The process of selecting and allocating the frequency sub-bands for all cellular base station within a system is called Frequency reuse. The cells of same frequency are chosen such that there is sufficient distance for minimum interference i.e. no two neighbors use the same set of frequency. A set of several cells are further grouped into clusters and cells of same cluster have different frequency.

If S is total number of duplex channels available, and N is the number of cells (cluster cells) and k is channels allocated to each cell ($k < S$)

$$S = Nk$$

$$\text{or } k = \frac{S}{N}$$

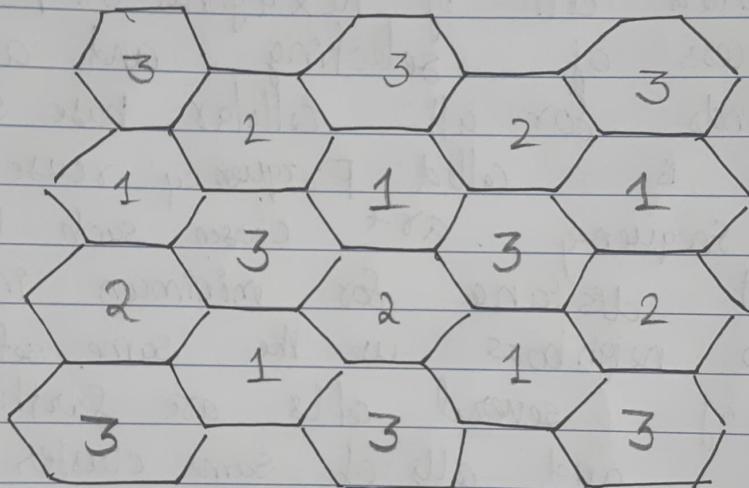
The frequency reuse factor is $\frac{1}{N}$

If cluster size is reduced and cell size remains constant more clusters are needed to cover area ie capacity is more.

If cluster size is large, the ratio between cell radii and d (distance between co-cells) is small

Eg -

For $N=3$, frequency reuse factor is $1/3$



The value of N must satisfy the condition

$$N = i^2 + ij + j^2 \quad \therefore i, j = 0, 1, 2, 3 \dots$$

ie possible values are 1, 3, 4, 7, 9, 12, 13, 16, 19

Within cellular system of cell is repeated m times

capacity C ps

$$C = M \frac{kN}{S}$$

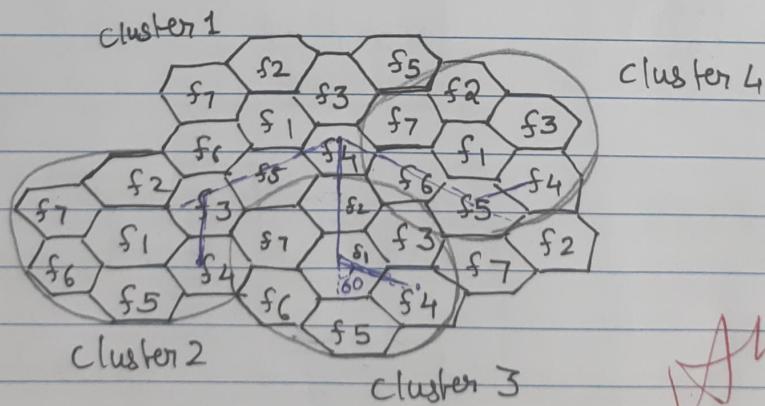
Co-channel cells are those that use same set of frequencies and thus must be separated by 'd' to avoid interference.

$$d = \gamma^* (3N)^{1/2} \quad . \quad \gamma : \text{cell radius}$$

To find nearest co-channel neighbour of a cell

- Move i cells through center of successive cells
- Turn 60° in anticlockwise direction
- Move j cells forward through centers of successive cells

For $N=7$ i.e. $i=2$ and $j=1$



AH

Conclusion - I implemented Frequency reuse and learnt about clusters, co-channel cells and other related concepts.

EXPERIMENT 8

Code:

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import java.util.*;
import java.lang.Math;

class Hexagon {

    private JPanel parent;
    private int x;
    private int y;
    private int length;
    private Color color;
    private String tags;

    public Hexagon(JPanel parent, int x, int y, int length, Color color, String tags) {
        this.parent = parent;
        this.x = x;
        this.y = y;
        this.length = length;
        this.color = color;
        this.tags = tags;
        drawHex();
    }

    private void drawHex() {
        int start_x = this.x;
        int start_y = this.y;
        int angle = 60;
        ArrayList<int[]> coords = new ArrayList<>();
    }
}
```

Niyati_Savant_TE_C31_2103156

```
for (int i = 0; i < 6; i++) {  
    int end_x = start_x + (int) (this.length * Math.cos(Math.toRadians(angle * i)));  
    int end_y = start_y + (int) (this.length * Math.sin(Math.toRadians(angle * i)));  
    coords.add(new int[] {start_x, start_y});  
    start_x = end_x;  
    start_y = end_y;  
}  
  
int[] xPoints = new int[6];  
int[] yPoints = new int[6];  
for (int i = 0; i < 6; i++) {  
    xPoints[i] = coords.get(i)[0];  
    yPoints[i] = coords.get(i)[1];  
}  
this.parent.getGraphics().setColor(this.color);  
this.parent.getGraphics().fillPolygon(xPoints, yPoints, 6);  
this.parent.getGraphics().setColor(Color.BLACK);  
this.parent.getGraphics().drawPolygon(xPoints, yPoints, 6);  
}  
}  
  
class FrequencyReuse extends JFrame {  
    private static final int CANVAS_WIDTH = 800;  
    private static final int CANVAS_HEIGHT = 650;  
    private static final int[] TOP_LEFT = {20, 20};  
    private static final int[] BOTTOM_LEFT = {790, 560};  
    private static final int[] TOP_RIGHT = {780, 20};  
    private static final int[] BOTTOM_RIGHT = {780, 560};  
    private int curr_angle;  
    private boolean first_click;  
    private boolean reset;  
    private int edge_len;  
    private int cluster_size;
```

Niyati_Savant_TE_C31_2103156

```
private ArrayList<Integer> reuse_list;
private boolean all_selected;
private int curr_count;
private ArrayList<Hexagon> hexagons;
private ArrayList<int[]> co_cell_endp;
private JPanel canvas;
private JTextArea textbox;

public FrequencyReuse(int cluster_size, int columns, int rows, int edge_len) {
    this.curr_angle = 330;
    this.first_click = true;
    this.reset = false;
    this.edge_len = edge_len;
    this.cluster_size = cluster_size;
    this.reuse_list = new ArrayList<>();
    this.all_selected = false;
    this.curr_count = 0;
    this.hexagons = new ArrayList<>();
    this.co_cell_endp = new ArrayList<>();

    this.canvas = new JPanel() {
        @Override
        protected void paintComponent(Graphics g) {
            super.paintComponent(g);
            createGrid(16, 10);
        }
    };
    this.canvas.setPreferredSize(new Dimension(CANVAS_WIDTH, CANVAS_HEIGHT));
    this.canvas.setBackground(new Color(77, 208, 225));
    this.canvas.addMouseListener(new MouseAdapter() {
        @Override
        public void mouseClicked(MouseEvent e) {
```

```
        callBack(e);
    }
});

this.canvas.setFocusable(true);

this.canvas.getInputMap().put(KeyStroke.getKeyStroke(KeyEvent.VK_R,
InputEvent.SHIFT_DOWN_MASK), "reset");

this.canvas.getActionMap().put("reset", new AbstractAction() {

    @Override

    public void actionPerformed(ActionEvent e) {

        resets(e);
    }
});

this.setTitle("Niyati's Frequency reuse");
this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
this.getContentPane().add(this.canvas);
this.pack();
this.setVisible(true);

}

private void createGrid(int cols, int rows) {

    int size = this.edge_len;
    for (int c = 0; c < cols; c++) {

        int offset = (c % 2 == 0) ? 0 : (int) (size * Math.sqrt(3) / 2);
        for (int r = 0; r < rows; r++) {

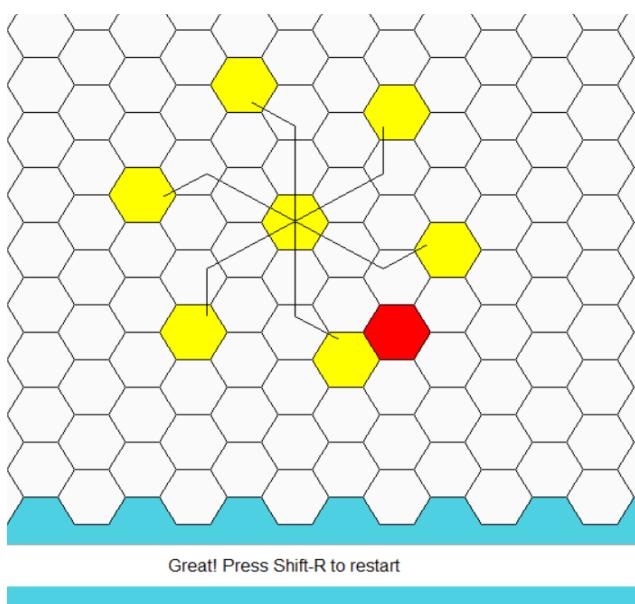
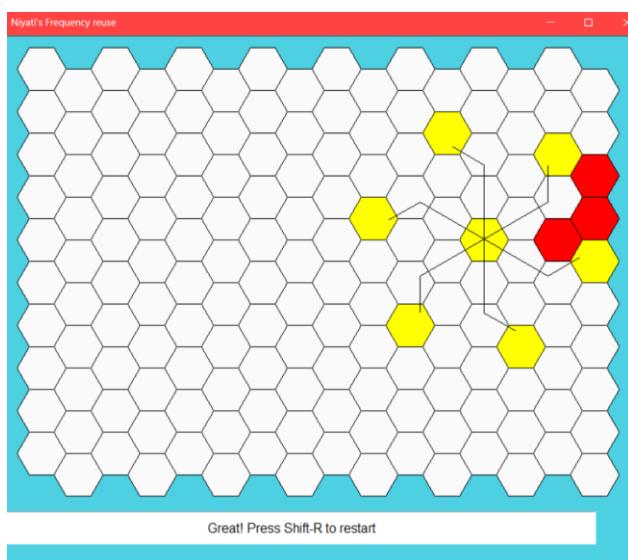
            int x = c * (int) (this.edge_len * 1.5) + 50;
            int y = r * (int) (this.edge_len * Math.sqrt(3)) + offset + 15;
            Hexagon hx = new Hexagon(this.canvas, x, y, this.edge_len, new Color(250, 250, 250), r +
            "," + c);
            this.hexagons.add(hx);
        }
    }
}
```

```
private void callBack(MouseEvent evt) {  
    int selected_hex_id = 1;  
    Hexagon hexagon = this.hexagons.get(selected_hex_id - 1);  
    int s_x = hexagon.x;  
    int s_y = hexagon.y;  
    int[] approx_center = new int[]{s_x + 15, s_y + 25};  
  
    if (this.first_click) {  
        this.first_click = false;  
        this.co_cell_endp.add(approx_center);  
        this.textbox.setText("Now, select another hexagon such that it should be a co-cell of the  
original hexagon.");  
        hexagon.drawHex();  
    } else {  
    }  
}  
  
private void resets(ActionEvent evt) {  
    this.first_click = true;  
    this.curr_angle = 330;  
    this.curr_count = 0;  
    this.co_cell_endp.clear();  
    this.reuse_list.clear();  
    for (Hexagon hexagon : this.hexagons) {  
        hexagon.drawHex(); // Redraw each hexagon with its original color  
    }  
    this.textbox.setText("Select a Hexagon");  
}  
  
public static void main(String[] args) {  
    System.out.println("Niyati's code for Frequency Reuse");
```

```
Scanner scanner = new Scanner(System.in);
System.out.println("Enter i & j values. common (i,j) values are:");
System.out.println("(1,0), (1,1), (2,0), (2,1), (3,0), (2,2)");
System.out.print("Enter i: ");
int i = scanner.nextInt();
System.out.print("Enter j: ");
int j = scanner.nextInt();
int N;
if (i == 0 && j == 0) {
    throw new IllegalArgumentException("i & j both cannot be zero");
} else if (j > i) {
    throw new IllegalArgumentException("value of j cannot be greater than i");
} else {
    N = (int) (Math.pow(i, 2) + i * j + Math.pow(j, 2));
    System.out.println("N is " + N);
}
new FrequencyReuse(N, 16, 10, 30);
}
```

Output:

```
Niyati's code for Frequency Reuse
Enter i & j values. common (i,j) values are:
(1,0), (1,1), (2,0), (2,1), (3,0), (2,2)
Enter i: 2
Enter j: 1
N is 7
```



Nilesh Savant
Niyati Savant
TE - C31

2103156

Experiment 9

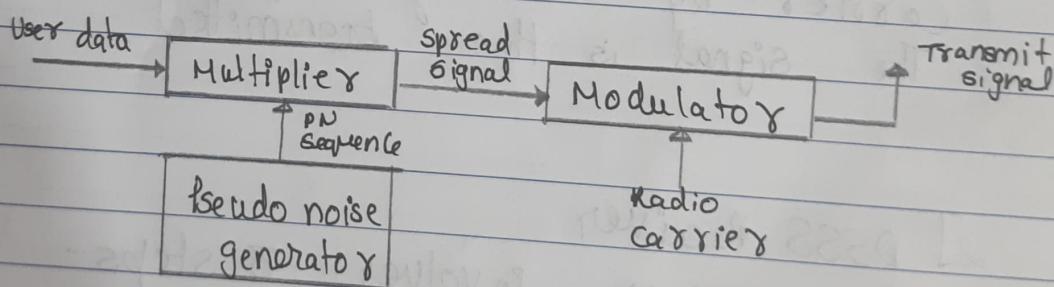
Aim - Write a Program to explain the concept of DSSS

Theory -

Spread spectrum is an important form of encoding for wireless communications. It is a wideband technology where frequency of transmitted signal is deliberately spread in the frequency domain so that the resultant signal has much greater bandwidth than original signal.

DSSS - Direct Sequence Spread Spectrum

1] DSSS Transmitter



The Transmitter has 2 steps -
i) Spreading the signal

Achieved by modulating user data with a digital code sequence of higher bit rate. It is typically a pseudorandom binary code also called

Niyati Savant
TEC31
2103156

PN ("pseudo-noise") or chipping sequence. It can be simply done by XORing user bit stream with chipping sequence. The time period of a single bit in PN code is called chip and bit rate of the code is chip rate.

If t_b is bit duration of user data and t_c is duration of one chip, the spreading factor, determines signal bandwidth

$$S = \frac{t_b}{t_c}$$

If 'w' is bandwidth of original, resulting needs (S.W) bandwidth.

ii) Radio modulation

The spread signal is modulated with radio carrier by shifting signal to carrier frequency. The signal is then transmitted.

2] DSSS Receiver

The receiver involves 3 steps -

i) Demodulation

Demodulation of received signal is achieved by using same carrier as the transmitter, reversing the modulation process. Bandwidth of resultant is almost same as that of original spread signal.

i) Correlator

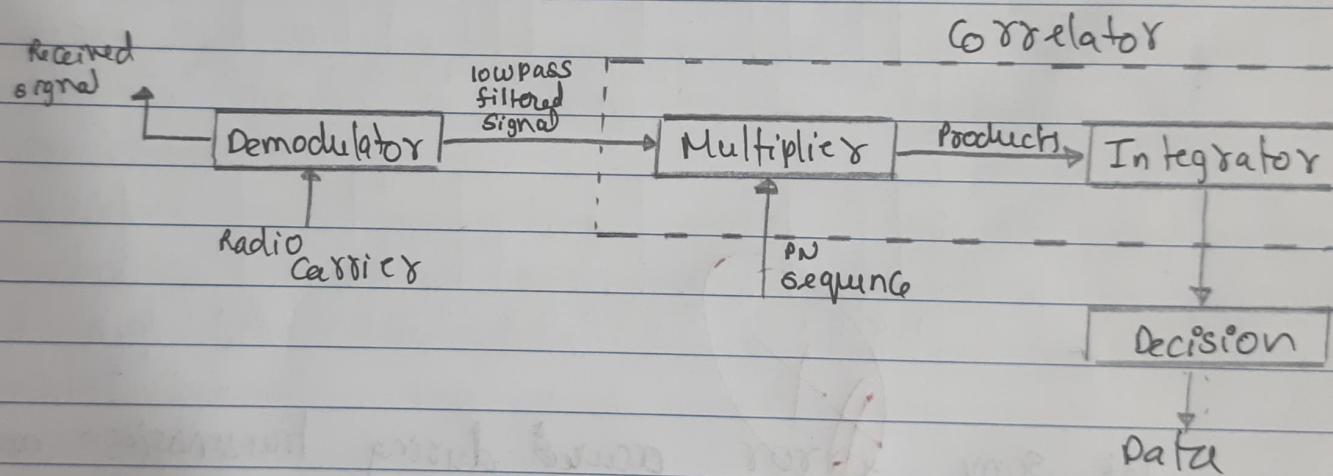
The receiver uses same pseudo random sequence as the transmitter, so the chip sequence must be precisely synchronized as the receiver calculates product of chip with incoming signal (XOR). During a bit period, an integrator adds them.

iii) Decision Unit

It decides if sum represents binary 0 or 1 as per sum.

If $0 < \text{sum} < 4$ - bit is 0

$7 < \text{sum} < 11$ - bit is 1



Example :

At Transmitter:

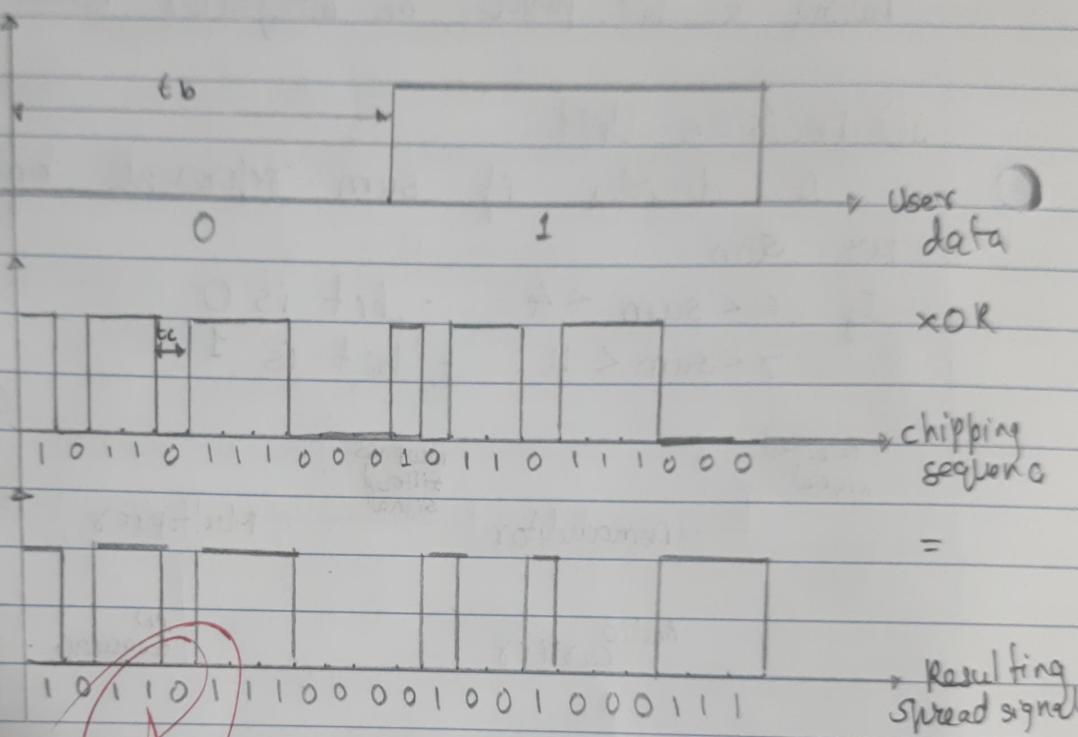
User data : 0 1

Chip : 1 0 1 1 0 1 1 1 0 0 0 (11-chip Barker code)

XOR process:

Niyati Savan
TE - C31
2103156

	0	1
XOR	10110111000	10110111000
Spread Signal	10110111000	01001000111



Let some ~~error~~ occurred during transmission and received signal was -

~~101011010000110100011~~ $\{$ - : reversed bits $\}$

XOR 10110110001011011000

Result 0001101000110111111

Sum	3	10
Decision Unit	0	1

Conclusion - I implemented DSSS code using Barker code and input signal

EXPERIMENT 9

Code:

```
import java.lang.*;
import java.util.*;

public class DSSS {
    public static String getString(char a) {
        if (a == '1') {
            return "1111111111";
        }
        else {
            return "0000000000";
        }
    }

    // Method for performing XOR
    public static String getXOR(String x, String y)
    {
        StringBuilder z = new StringBuilder();
        for (int i = 0; i < x.length(); i++) {
            if ((x.charAt(i) == '1' && y.charAt(i) == '1') || (x.charAt(i) == '0' && y.charAt(i) == '0')) {
                z.append("0");
            }
            else if ((x.charAt(i) == '0' && y.charAt(i) == '1') || (x.charAt(i) == '1' && y.charAt(i) == '0')) {
                z.append("1");
            }
        }
        System.out.println(x + " XOR with " + y + " is " + z.toString());
        return z.toString();
    }
}
```

```
public static int getNoOfOnes(String a) {  
    int count = 0;  
    for (int i=0; i<a.length(); i++) {  
        if (a.charAt(i) == '1') {  
            count = count + 1;  
        }  
    }  
    return count;  
}  
  
public static void main(String[] args) {  
    Scanner reader = new Scanner(System.in);  
    System.out.println("Niyati's code for DSSS ");  
    System.out.print("Enter the input string: ");  
    String inputString = reader.next();  
  
    String Barker_code = "10110111000";  
  
    StringBuilder eOutput = new StringBuilder();  
    for (int i=0; i<inputString.length(); i++) {  
        String a = getString(inputString.charAt(i));  
        eOutput.append(getXOR(a, Barker_code));  
    }  
    System.out.println("Encrypted message: " + eOutput);  
  
    ArrayList<String> enStrings = new ArrayList<>();  
    for (int i=0; i<eOutput.length(); i = i + 11) {  
        enStrings.add(eOutput.substring(i, i + 11));  
    }  
}
```

```
StringBuilder dOutput = new StringBuilder();
for (String enString: enStrings) {
    String a=getXOR(enString, barker_code);
    if (getNoOfOnes(a) > 7) {
        dOutput.append("1");
    }
    else if (getNoOfOnes(a) < 3) {
        dOutput.append("0");
    }
}
System.out.println("Decrypted message: " + dOutput);
}
```

}

Output:

```
javac DSSS.java } ; if ($?) { java DSSS }
Niyati's code for DSSS
Enter the input string: 01
0000000000 XOR with 10110111000 is 10110111000
1111111111 XOR with 10110111000 is 01001000111
Encrypted message: 1011011100001001000111
10110111000 XOR with 10110111000 is 00000000000
01001000111 XOR with 10110111000 is 11111111111
Decrypted message: 01
```

Experiment 10

Aim - Write program to implement SAS/AS/A8 GSM security Algorithm.

Theory -

GSM offers several security services using confidential information stored in AUC and SIM. The SIM stores personal, secret data and is protected with a PIN against unauthorized use. The security services offered by GSM are -

- Access Control and Authentication

Includes authentication of valid SIM user. User needs a secret PIN to access a SIM. GSM also authenticates subscriber through a challenge-response mechanism.

- Confidentiality

Achieved by encrypting data over air interface. After authentication, MS and BTS apply encryption to voice, data and signalling information. It is not end-to-end but exists between MS and BTS only.

- Anonymity

Subscriber identity is always hidden over air interface.

To ensure confidentiality, TMSI is used and VLR may change it any time.

The 3 security algorithms. -

A3 : Authentication

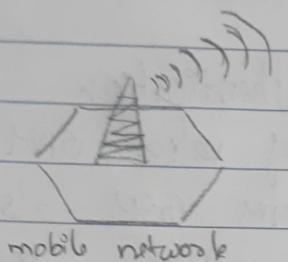
A5 : Encryption

A8 : Generation of cipher key

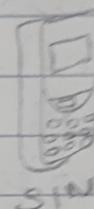
A3, A8 are located on the SIM and in AUC and A5 is implemented in VLR and is thus common for all service providers unlike the other two.

Authentication

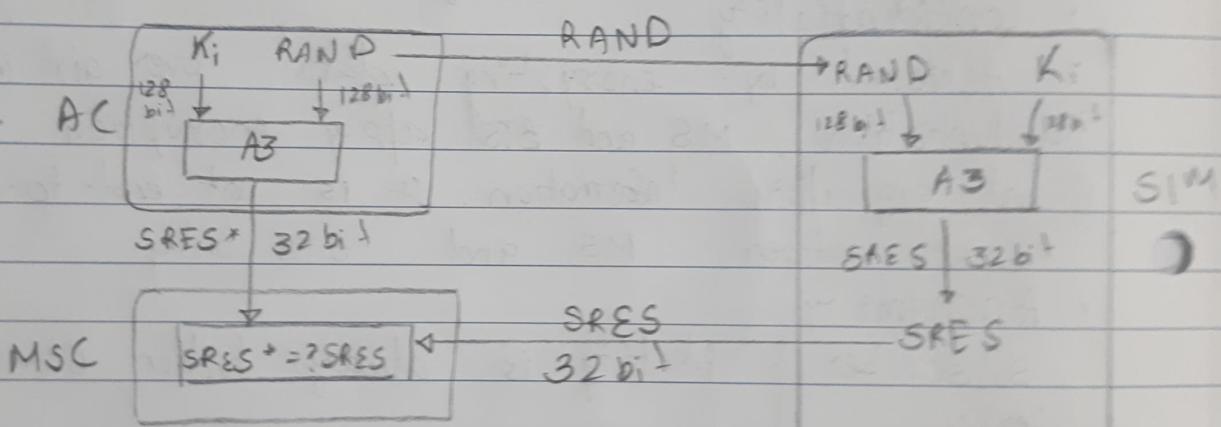
Before accessing any GPRS service, user must be authenticated. It is based on SIM that stores individual authentication key Ki, user identification IMSI and A3. It uses challenge-response method.



mobile network



SIM



Ki : Individual subscriber key

SRES : Signed response

- 1) AC - access control generates 128 bit random number RAND as a challenge.
- 2) VLR sends it to MS
- 3) MS computes SRES as per RAND with A3

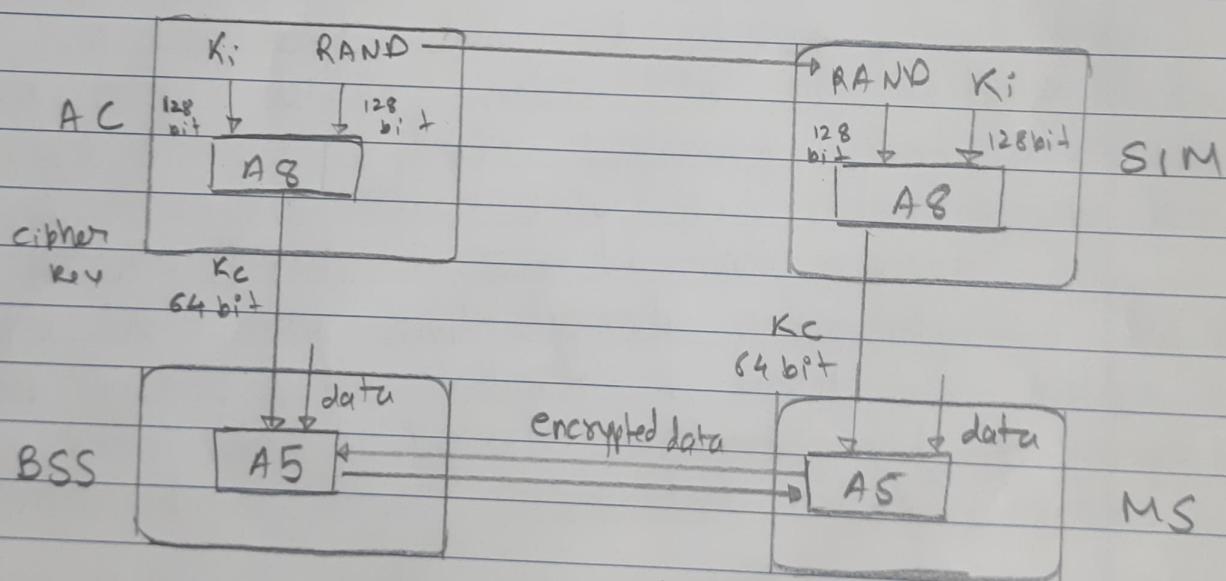
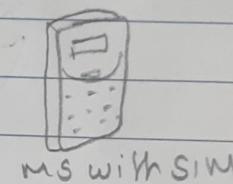
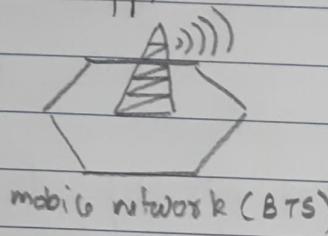
using key (K_i)

- 4) MS sends SRES to MSC
- 5) AC also calculates SRES
- 6) MSC compares the SRES values received by AC and MS and subscriber is accepted if these match.

Encryption

For privacy, all message with user-related information are encrypted over air interface.

Once authentication is done, MS and BSS can initiate encryption



The SIM and AC generate the 64 bit cipher key

TRN
Niyati Savant
2103156
TE - C31

- K_c using K_i and RAND and applying A8.
- The MS and BTS can now encrypt and decrypt data with A5 and R_c .
- The 64 bit K_c is only strong to protect against simple eavesdropping.

Conclusion-

I implemented GSM security algorithms and learnt about how message is encrypted and authenticated.

AM

EXPERIMENT 10

A3 ALGORITHM

Code :

```
import java.lang.Math;

public class A3 {
    static int[] GenerateBits() {
        int[] a = new int[16];
        for (int i = 0; i < 16; i++) {
            double rand = Math.random();
            if (rand >= 0.5) {
                a[i] = 1;
            } else {
                a[i] = 0;
            }
        }
        return a;
    }

    static int[] XOR(int[] a, int[] b) {
        int[] temp = new int[16];
        for (int i = 0; i < 16; i++) {
            if (a[i] == 1 && b[i] == 1 || a[i] == 0 && b[i] == 0) {
                temp[i] = 0;
            } else {
                temp[i] = 1;
            }
        }
        System.out.print(temp[i]);
    }
}
```

```
return temp;  
}  
  
public static void main(String[] args) {  
    System.out.println("Niyati's code for A3 algorithm");  
    int[] a;  
    System.out.println("Generating the random number");  
    a = GenerateBits();  
    for (int i=0; i<16; i++) {  
        System.out.print(a[i]);  
    }  
  
    int[] b;  
    System.out.println("\n\nGenerating the key identification number");  
    b = GenerateBits();  
    for (int i = 0; i <16; i++) {  
        System.out.print(b[i]);  
    }  
  
    int[] c;  
    System.out.println("\n\nGenerating the Barker code");  
    c = GenerateBits();  
    for (int i = 0; i <16; i++) {  
        System.out.print(c[i]);  
    }  
  
    System.out.println("\n\nChecking for mobile network: ");  
    int[] z;  
    System.out.println("\n XOR of key identification number and the random number");  
    z = XOR(a, b);  
    System.out.println("\n\nXOR of the above number and the Barker code");  
    z = XOR(z, c);
```

Niyati_Savant_TE_C31_2103156

```
System.out.println("\n\nChecking for SIM :");
int[] z1;
System.out.println("\nXOR of key identification number and the random number");
z1 = XOR(a, b);
System.out.println("\n\nXOR of the above number and the barker code");
z1 = XOR(z1, c);

int flag = 0;
for (int i=0; i<16; i++) {
    if (z[i] != z1[i]) {
        flag = 1;
        break;
    }
}
if (flag == 1) {
    System.out.print("\n\nSigned responses do not match, therefore authentication failed");
}
else{
    System.out.print("\n\nSigned responses match, therefore authentication passed");
}
}
```

Output:

```
PS C:\Engineering\3rd Year\Sem VI\PRACTICALS\MCC> cd "c:\javac A3.java } ; if ($?) { java A3 }
Niyati's code for A3 algorithm
Generating the random number
1111000001100010

Generating the key identification number
1001111001110100

Generating the barker code
011111010010001

Checking for mobile network:

XOR of key identification number and the random number
0110111000010110

XOR of the above number and the barker code
0001000010000111
```

Checking for SIM :

XOR of key identification number and the random number
0110111000010110

XOR of the above number and the barker code
0001000010000111

Signed responses match, therefore authentication passed

A5 ALGORITHM

Code

```
public class A5_algo {  
    public static final int SIZEX = 19;  
    public static final int SIZEY = 22;  
    public static final int SIZEZ = 23;  
  
    public static int maj(int x, int y, int z) {  
        int m;  
        if (x == 0) {  
            if (y == 0 || z == 0) {  
                m = 0;  
            } else {  
                m = 1;  
            }  
        } else {  
            if (y == 1 || z == 1) {  
                m = 1;  
            } else {  
                m = 0;  
            }  
        }  
        return m;  
    }  
  
    public static void main(String[] args) {  
        int[] x = {1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1};  
        int[] y = {1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1};  
        int[] z = {1, 1, 1, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 0, 0, 0};  
  
        System.out.println("Niyati - code for A5 algorithm");  
    }  
}
```

```
int m;
for (int i = 0; i < 9; i++) {
    m = maj(x[8], y[10], z[10]);

    System.out.print("X: ");
    if (x[8] == m) {
        int t = x[13] ^ x[16] ^ x[17] ^ x[18];
        for (int j = SIZEX - 1; j >= 0; j--) {
            if (j == 0) {
                x[j] = t;
            } else {
                x[j] = x[j - 1];
            }
        }
    }
    for (int j = 0; j < SIZEX; j++) {
        System.out.print(x[j]);
    }

    System.out.print("\nY: ");
    if (y[10] == m) {
        int t = y[20] ^ y[21];
        for (int j = SIZEY - 1; j >= 0; j--) {
            if (j == 0) {
                y[j] = t;
            } else {
                y[j] = y[j - 1];
            }
        }
    }
    for (int j = 0; j < SIZEY; j++) {
        System.out.print(y[j]);
    }
}
```

```
}

System.out.print("\nZ: ");

if (z[10] == m) {

    int t = z[7] ^ z[20] ^ z[21] ^ z[22];

    for (int j = SIZEZ - 1; j >= 0; j--) {

        if (j == 0) {

            z[j] = t;

        } else {

            z[j] = z[j - 1];

        }

    }

    for (int j = 0; j < SIZEZ; j++) {

        System.out.print(z[j]);

    }

    if (i < 8) {

        int key = x[SIZEX - 1] ^ y[SIZEY - 1] ^ z[SIZEZ - 1];

        System.out.printf("\nKeystream bit = %d ^ %d ^ %d = %d\n", x[SIZEX - 1], y[SIZEY - 1], z[SIZEZ - 1], key);

        System.out.println();

    }

}

}
```

Output:

```
Niyati - code for A5 algorithm
X: 010101010101010101010
Y: 1100110011001100110011
Z: 11110000111100001111000
Keystream bit = 0 ^ 1 ^ 0 = 1

X: 0010101010101010101
Y: 0110011001100110011001
Z: 11110000111100001111000
Keystream bit = 1 ^ 1 ^ 0 = 0

X: 0001010101010101010
Y: 1011001100110011001100
Z: 01111000011110000111100
Keystream bit = 0 ^ 0 ^ 0 = 0

X: 0001010101010101010
Y: 0101100110011001100110
Z: 10111100001111000011110
Keystream bit = 0 ^ 0 ^ 0 = 0

X: 0000101010101010101
Y: 1010110011001100110011
Z: 10111100001111000011110
Keystream bit = 1 ^ 1 ^ 0 = 0
```

```
X: 0000101010101010101  
Y: 1010110011001100110011  
Z: 10111100001111000011110  
Keystream bit = 1 ^ 1 ^ 0 = 0  
  
X: 00000101010101010  
Y: 1010110011001100110011  
Z: 01011110000111100001111  
Keystream bit = 0 ^ 1 ^ 1 = 0  
  
X: 00000010101010101  
Y: 0101011001100110011001  
Z: 10101111000011110000111  
Keystream bit = 1 ^ 1 ^ 1 = 1  
  
X: 00000001010101010  
Y: 1010101100110011001100  
Z: 10101111000011110000111  
Keystream bit = 0 ^ 0 ^ 1 = 1  
  
X: 00000000101010101  
Y: 1010101100110011001100  
Z: 01010111100001111000011
```

Q1 Explain in detail with merits and demerits

a) Snooping TCP

- Snooping TCP works completely transparently and leaves the TCP end-to-end connection intact. It overcomes the some of the I-TCP drawbacks.

Working -

- Correspondent host sends to mobile host via happens via wired TCP Access point buffers packet ~~H sent~~ sent by host
 - Access point snoops on packet in both directions to reorganize acknowledgements
 - Once host receives packet, the ack. and also passes through access point.
 - If access point does not get any ack from mobile host within certain time, it retransmits the packet from its buffer, performing faster retransmission.
 - Since time-out reflects only the delay of one hop and processing time, it can be much shorter.
 - Mobile host may send duplicate acknowledgements for the same packet to indicate a packet loss. So retransmission after these duplicate acknowledgements is unnecessary.
- Mobile host transmits packet to correspondent host.

AH A

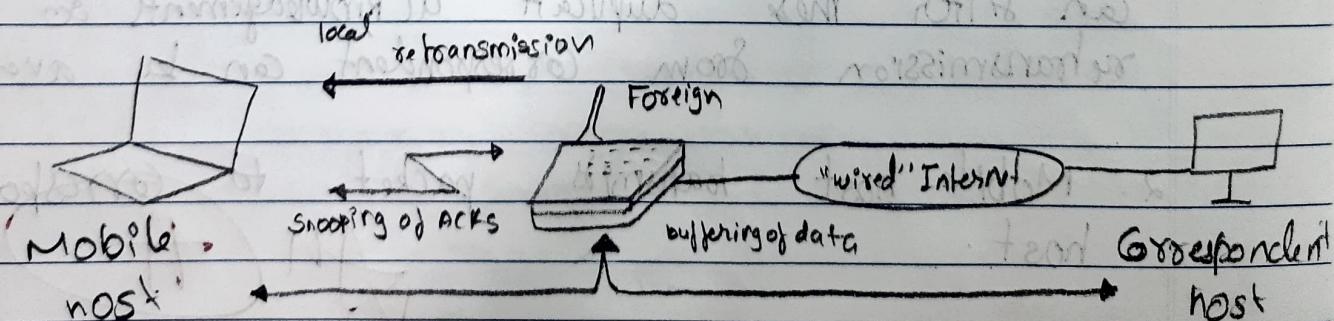
- When a mobile sends packet to correspondent foreign agent keeps track of sequence numbers of these packets
- When foreign agent detects gap in sequence numbers, i.e. packet loss, it sends negative ack or NACK to mobile
- Once mobile host receives NACK, it can retransmit missing packet immediately
- Reordering of packets is done automatically at correspondent node.

Advantages -

- Preserve end-to-end semantics
- Enhancements are done in foreign agent and correspondent host need not be changed
- No need of handover if mobile moves to another foreign agent

Disadvantages -

- Use NACK between foreign agent and mobile host assumes additional mechanisms on mobile
- Fails on end-to-end encryption
- Does not isolate behaviour of wireless link from wired



b] Mobile TCP

M-TCP deals with lengthy and/or frequent disconnections. It aims to improve overall throughput, lower delay, maintain end-to-end TCP semantics, provide efficient handover.

The connection is split up into 2 parts - Correspondent node and supervisory host that communicate via unmodified standard TCP and that return supervisory (SN) and mobile (MH) is via modified special TCP.

• Working -

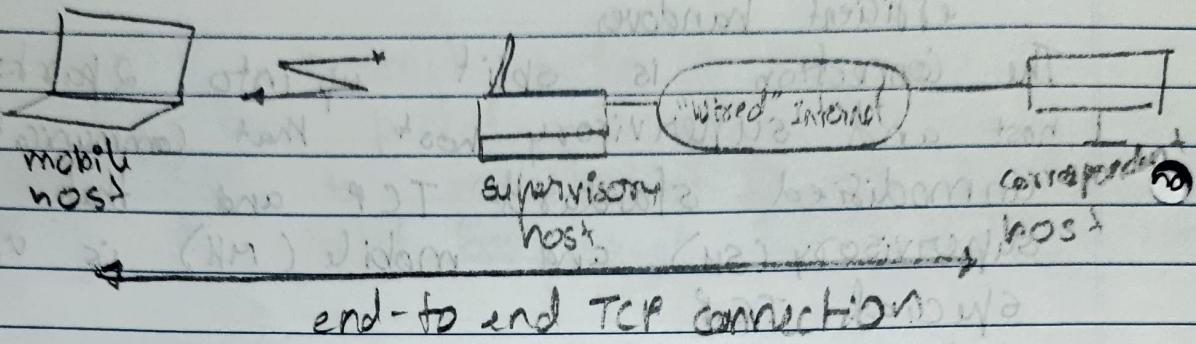
Packets are sent to MH by correspondent. If loss on wireless link, original sender will retransmit. SN monitors all packets sent to MH and packets are acknowledged by MH via ACK. After set time, if SN gets no ACK, it assumes MH is disconnected. SN sets sender window size to 0. to check it and it forces sender to go into persistent mode where independent of receiver's period of disconnected state, sender state won't change. Once SN detects connectivity again, sender size is reset to old value.

• Advantages

- Maintain end-to-end semantics.
- Avoid unnecessary retransmission
- Not necessary to forward all data to new SN

Disadvantages

- Losses on wireless propagated to wired link
- Needs new network elements like bandwidth manager



C) I-TCP (Indirect TCP)

- TCP performs poorly together with wireless links and within fixed network, TCP cannot be changed
- I-TCP separates TCP connection into 2 parts.
- Fixed part - between mobile support router (access point) and fixed host over fixed network
- Wireless part - between MH (mobile host) and its access point.

Segmenting the connection between mobile host and correspondent host is done by agent.

- i) Correspondent sends packet to mobile host.
- As correspondent is fixed, it sends TCP packet via standard TCP.

A CA access point receives the packet, sends ACK to fixed host and it buffers the packet & forwards it to mobile host with wireless TCP. In case of transmission error, access point will retransmit local retransmission.

- Mobile host transmits packet to fixed host
 - If sends packet, a CA access point receives it and sends ACK to mobile host. If lost, mobile host notices the packet loss even faster and retransmits.

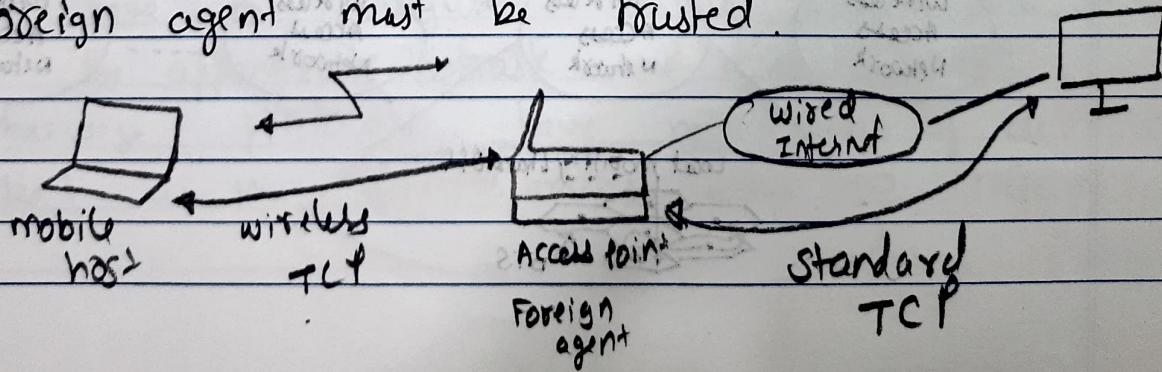
- Mobile host moves to new location
 - It registers new foreign agent and upon registration old agent forwards all buffered packets to new agent. With the data, access point sockets must also migrate to new agent.

Advantages

- No changes in TCP of wired
- Errors don't propagate to fixed
- Simple to control
- Fast retransmission

Disadvantages

- Losses end-to-end semantics
- Higher handover latency
- Foreign agent must be trusted



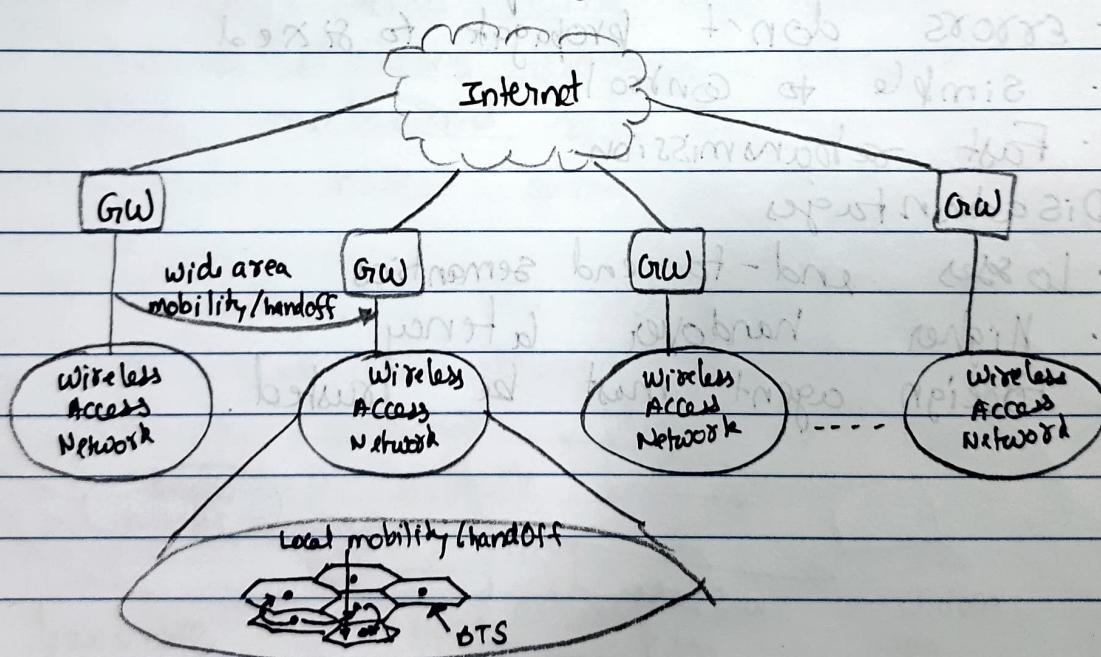
Q2] Explain Cellular (Mobile) IP.

→ Mobile IP exhibits problem when there is large no. of mobile devices changing network frequently and moving very fast. In such cases, a high load on home agents and on the network is generated by registration and binding update messages. Mobile IP is basically designed only for macro level mobility and relatively slow moving hosts.

CIP or Cellular IP is a new robust, simpler, flexible protocol for highly mobile hosts. It supports local mobility and can accommodate large number of users by separating idle hosts from active hosts.

The architecture has 3 main components

- Cellular IP gateway (GW)
- Cellular IP node or base station (BS)
- Cellular IP mobile host (MH)



CIP network has several interconnected BSs. These communicate with mobile hosts (MNs) via wireless interface and also route IP packets inside cellular network. The base stations are built on regular IP forwarding engines but the routing is cellular IP routing. CIP gateway router connects cellular IP and regular internet. Gateway mobility is managed by Mobile IP.

Q3 Explain MIPv6.

- MIPv6 or Mobile IPv6 is a protocol developed as a subset of IPv6 for mobility.
- First IP mobility protocol, Mobile IP was developed for IPv4 which solved the TCP/IP layer 3 mobility problem.
- MIPv6 is an update of Mobile IP standard designed to authenticate mobile devices using IPv6 address.
- In traditional IP routing, IP addresses represent a topology and routing mechanisms rely on assumption that each network node has a same point of attachment to Internet.
- In such scheme, if you disconnect a device and want to reconnect through a different network, you have to configure device with a new IP address, and the appropriate netmask and default router.
- Otherwise protocols have no means of delivering packets as the address address have no necessary information.

Mobility IPv6 allows the node to transparently maintain connections by identifying device through home address.

Q4 Short note on HAWAII

→ HAWAII or Handoff-Aware Wireless Access Internet Infrastructure, tries to keep micro-mobility support as transparent as possible for both home agent and MN.

Working

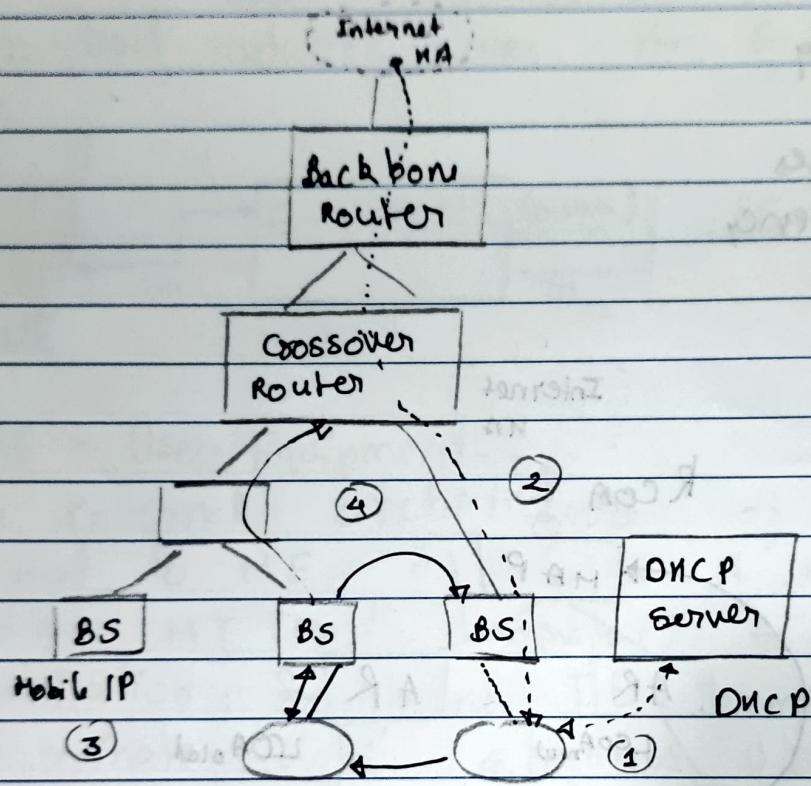
- 1: On entering on HAWAII domain, a mobile node obtains a co-located CoA
- 2: MN registers with the HA
- 3: When MN moves another cell inside the foreign domain, MN sends registration request to new base station
- 4: The base station interprets registration request and sends out a handoff update message which reconfigures all routers on paths from old and new base station to cross over router. When routing has been reconfigured successfully, base station sends registration reply to MN again.

Advantages

- security
- Transparency

Disadvantages.

- Raise OMCP security issue
- Decentralized security - critical functionality
- Unstructured Authentication
- Need PKI or AAA infrastructure



Q5 Explain HMIPv6

→ Hierarchical Mobile IPv6 a.k.a HMIPv6 provides micro mobility support by installing a mobility anchor point (MAP). MAP is an entity which is responsible for certain domain and acts as a local HA within this domain for visiting MNs.

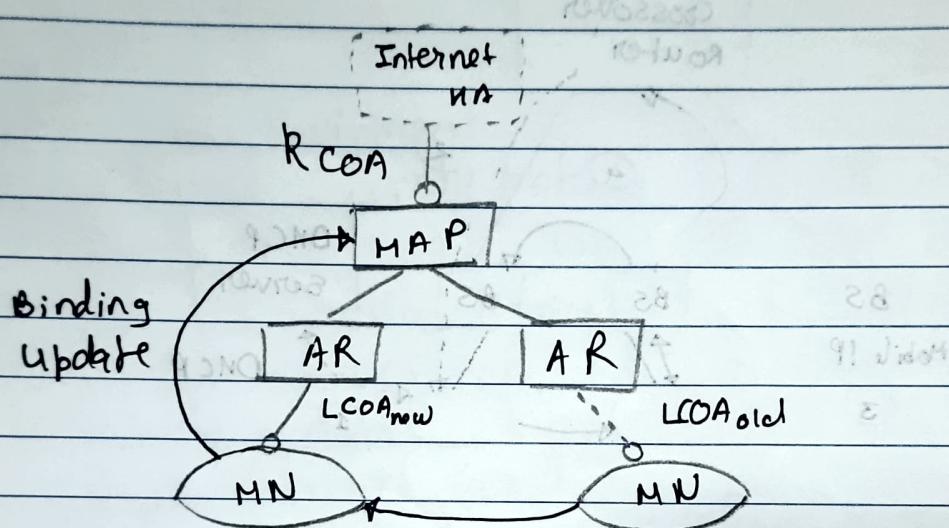
The MAP receives all packets on behalf of MN, encapsulates and forwards them directly to MN's current address LCOA (link COA)

Advantages -

- Security
- Efficiency

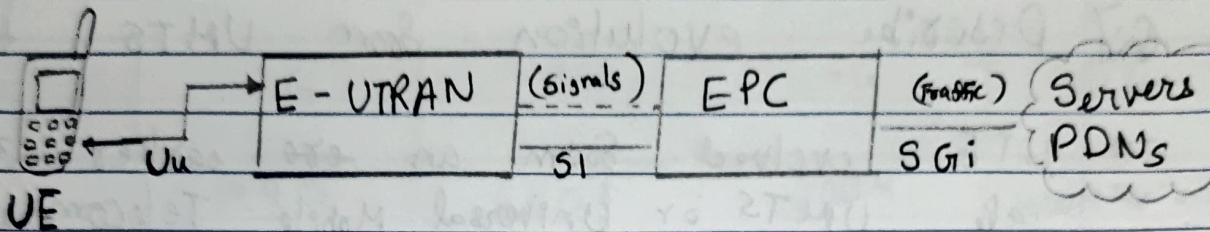
Disadvantages

- Transparency
- Security



Q1] Draw and Explain SAE architecture

→ SAE or System Architecture Evolution is a new network architecture designed to simplify LTE networks. It establishes a flat architecture similar to other IP based communications networks. The high-level network has the following components:



1) UE - User Equipment

The internal architecture of UE for LTE is identical to ME of GSM. It has the following modules - MT (mobile terminator) to handle all the communication functions, TE (Terminal Equipment) to terminate data streams, UICC (Universal Integrated Circuit Card), runs USIM application

2) E - UTRAN : Evolved UMTS Terrestrial Radio Access Network

It handles radio communication between mobile and evolved packet core and it has just one component, evolved base stations ie eNB or eNode B. Each eNB is a base station that controls mobiles in one or more cells. The base station that is communicating with a

A
A

mobile is called PS, bearing eNB

③ EPC - Evolved Packet Core

It provides means to communicate with packet data networks in outside world like internet.

It is connected to E-UTRAN via S1 interface and outside world via Gb interface.

Q7 Describe evolution from UMTS to LTE

→ LTE evolved from an earlier 3GPP system - called UMTS or Universal Mobile Telecommunication System which evolved UMTS from GSM. Even related specifications were formally called evolved UMTS terrestrial radio access (E-UTRA) and evolved UMTS. First version of LTE was documented in Release 8 of the 3GPP specifications.

There was no global standard for wireless broadband until the advent of LTE. The goal with LTE was to merge a fragmented market and offer a more efficient network for network operators. Main reasons behind evolution from UMTS to LTE :-

- There was a need to ensure continuity of competitiveness of the 3G system for the future.
- Users growing demands for higher data rates and quality of service.
- There was a need for packet switch.

optimised system

- Continued demand for cost reduction (CAPEX and OPEX).
- Low complexity
- For avoiding unnecessary fragmentation of technologies for paired and unpaired band operation.

Q3. Compare mobile generations (1G, 2G, 3G, 4G, 5G).

Parameter	1G	2G	3G	4G	5G
Introduced in	1980	1993	2001	2009	currently under development
Technology	AMPS	D-AMPS, GSM	UMTS-136	LTE, WiMAX	OMA and NOMA
Multiplexing	FDMA	TDMA	CDMA	CDMA	CDMA
Switching	Circuit	Circuit and packet	Packet	All packet	All packet
Speed	2.4 to 14.4 kbps	14.4 kbps	3.1 mbps	100 mbps	>10Gbps
Bandwidth	Analog	25MHz	25MHz	100MHz	60GHz
Band type	Narrow	Narrow	Wide	Ultra wide	Extremely high
Operating frequencies	800 MHz	GSM: 900 MHz CDMA: 800MHz	2100 MHz	2600MHz	3 To 40 GHz

[Q4] What are self-organizing networks?

- • SON or Self Organizing networks can configure all of its configuration by itself and makes itself ready for service by just adding an eNB and connecting power and switching on.
- It is an automation technology designed to make the planning, configuration, management, optimization and healing of mobile radio access network simpler and faster. Thus, it is like a 'Plug-and-Play'.
- Normally, when system operator constructs a network it must do the following steps -

- i) Network Planning
- ii) ~~Hardware~~ Bring hardware (like eNB) to locations determined
- iii) Hardware installation
- iv) Basic configuration
- v) Optimizing parameters

The main goal of SON is to automate large portions of human efforts involved in these processes.

- As per location of optimization algorithms, SON can be divided in 3 main architectures:

1) Centralized SON

Algorithms are stored and executed from OAM system. Here, SON functionality resides in small number of locations, at a high level in architecture.

This is easy to deploy but does not support simple and quick optimization cases.

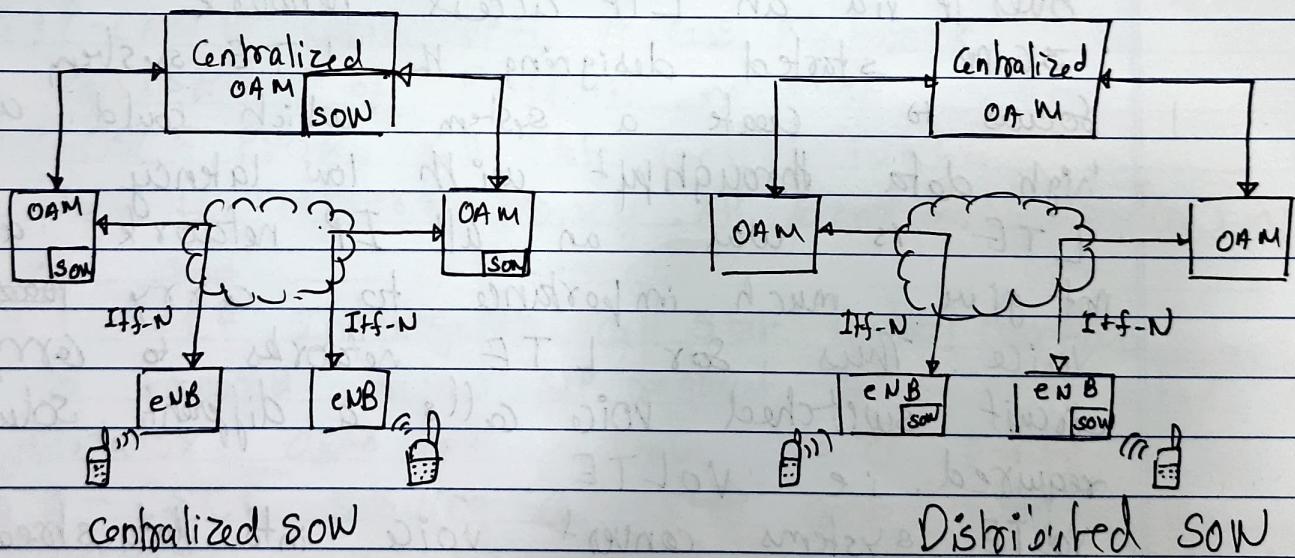
To implement this, existing (Iff-N) interface must be extended

2) Distributed SOW

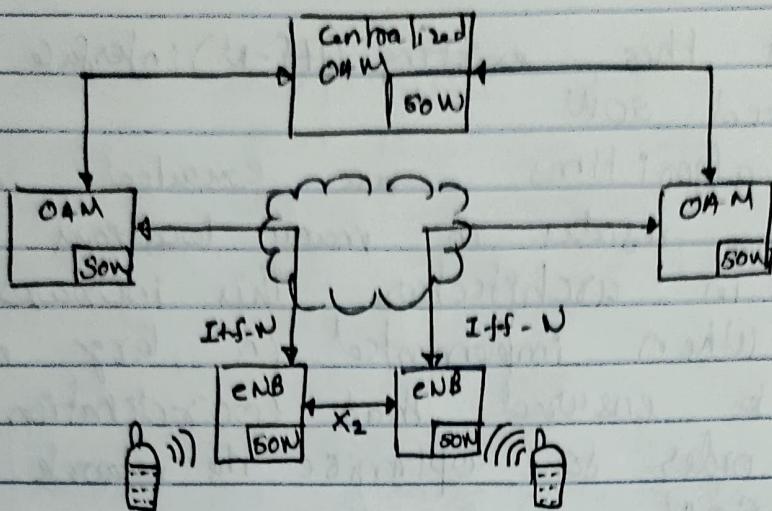
The algorithms are executed in eNBi. SON functionality resides in many locations at a relatively low level in architecture. This increases deployment efforts. When implemented in large number of nodes it must be ensured that coordination exists between them in order to optimise the network.

3) Hybrid SOW

In this architecture, part of the optimization algorithms are executed in OAM system, while others are executed in eNB. Simple and quick optimization schemes are implemented in eNB and complex ones in OAM to give flexibility to support different kinds of optimization.



(P.T.O)



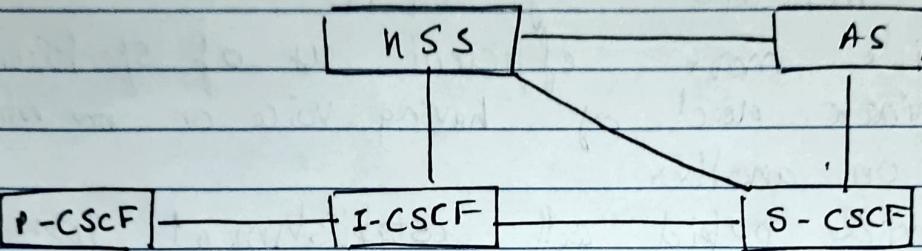
Hybrid SOW

Q5 Explain VOLTE in detail.

- VOLTE i.e. Voice over Long Term Evolution is a digital packet voice service that is delivered over IP via an LTE access network.
- 3GPP started designing the LTE system with focus to create a system which could achieve high data throughput with low latency.
- LTE was an all IP network and did not give much importance to carry traditional voice. Thus for LTE networks to carry traditional circuit switched voice calls a different solution was required. i.e. VOLTE.
- VOLTE systems convert voice into data stream, which is then transmitted using data connection. It is based on IMS (IP multimedia system).
- IMS is an architectural framework for

delivering multimedia communication services like voice, video, text messaging over IP networks.

- VOLTE enables system to be integrated with the suite of other applications for LTE.
- To make VOLTE implementation easy and cost effective to operators, a cut down version of IMS network that reduced number of entities required and simplified the interconnectivity, was defined.



The entities of reduced IMS network used for VoLTE -

- i) IP-CAN IP, Connectivity Access Network consists of EUTRAN and MME.
- ii) P-CSCF, Proxy Call State Control Function User to network proxy. All to and from SIP signaling runs via this
- iii) I-CSCF, Interrogating Call State Control Function To forward initial SIP request to S-CSCF, when initiator does not know exact S-CSCF
- iv) S-CSCF, Serving Call State Control Function performs a variety of actions within overall sys having multiple interfaces

C.P.T.O

v) AS, Application server

Application server to handle voice as an application

vi) HSS, Home Subscriber Server

Main subscriber database used within IMS. Provides details of subscribers to other entities within IMS, enabling users to be granted access

• Some benefits of VoLTE

i) Provide more efficient use of spectrum

ii) Eliminate need of having voice on one network and data on another

iii) Provide rapid call establishment time

iv) Can be deployed in parallel with video calls over 4G

v) Increases handset battery life by 40%.