

Thadomal Shahani Engineering College

Bandra (W.), Mumbai - 400 050.

© CERTIFICATE ©

Certify that Mr./Miss Shirish Shetty
of ComPS Department, Semester VI with
Roll No. 164 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 2024 - 2025


Teacher In-Charge

Head of the Department

Date 26/3/24

Principal

CONTENTS

SR. NO.	EXPERIMENTS	PAGE NO.	DATE	TEACHERS SIGN.
1.)	Write an Android application to draw Basic graphical 2 P primitives	1-4	16/01/24)
2.)	WAA to draw Basic graphical 3D primitives	5-6	23/1/24	
3.)	WAA to design a form with GUI components	7- 9	30/1/24	
4.)	WAA to design GUI component using DB	11-12	6/2/24	
5.)	WAA to develop an EMI calculator appln	13-15	20/2/24	
6.)	WAA that create an alert on receiving a message.	13-18	27/2/24	
7.)	WAA to implement Basic Calculator using Android.	19-20	5/3/24	Pjain ✓
8.)	Write a program to demonstrate Cellular frequency reuse.	21-22	12/3/24	26/3/24
9.)	Write a program to explain concept of DSSS	23-24	19/3/24	
10.)	Write a program to implement A3/A5/ A5 .gSM security Algorithm	24-25	26/3/24	
11.)	Written Assignment - 1	26-39	1/2/24	
11/2)	Written Assignment - 2	40-47	5/3/24	

Experiment no. 1

Aim - Write an Android Application to draw.
Basic 2D graphical primitives.

Theory :-

Android Studio is official integrated development environment for Google's Android Operating System, built on Jet Brains IntelliJ IDEA software and designed specifically for Android development, it is available for download on windows, macOS and Linux based operating system. It is a replacement for the Eclipse Android Development Tool as primary IDE for native Android application development.

Android development is usually done in programming languages Java, Kotlin and Flutter.

XML files are used as layout / display screen of your android application. You can add any text / image and various other elements.

We can make 2D graphical primitives in Android using Canvas and Paint objects.

Canvas is a class in Android Graphics framework that provides an area for drawing graphics on screen. It serves as a drawing surface where you can create & manipulate graphical element

such as lines, shapes, text, images.

Paint class in Android is used for styling and controlling how the elements are drawn on a canvas. It defines parameters such as color, stroke, width, text size and style that affect the appearance of graphics drawn on the canvas.

Function:- In Android Studio, functions are essential part of app development, allowing developers to break up their code into reusable pieces. They are particularly in Kotlin, the preferred language for Android development. Function in Android Studio are defined using the fun keyword in Kotlin. and can be passed as arguments and even returned from other functions. They help improve code organization, reusability and readability. By creating functions, developers can avoid duplicating code and make their programs more maintainable. Additionally, functions can be used to define and customize the behavior of various app components.

Conclusion:- learnt about basic shapes, how to draw in android studio.

Paint (A)

Experiment 1

Aim: Write an android application (WAA) to draw basic graphical 2D primitives.

Program:

MainActivity.java

```
package com.example.Exp1;

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);
        bg=Bitmap.createBitmap(750,1180, Bitmap.Config.RGB_565); //create bitmap
        img = findViewById(R.id.imageView);
        img.setImageBitmap(bg);
    }
}
```

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

Paint paint1 = new Paint();
paint1.setColor(Color.YELLOW);
paint1.setTextSize(50);

Paint paint2 = new Paint();
paint2.setColor(Color.WHITE);
paint2.setTextSize(50);

Paint paint = new Paint();
paint.setColor(Color.BLUE);
paint.setTextSize(50);

Paint paint3 = new Paint();
paint3.setColor(Color.CYAN);
paint3.setTextSize(50);

canvas.drawText("Rectangle",420,150,paint1);
canvas.drawRect(400,200,650,700,paint1);

canvas.drawText("Square",120,800,paint);
canvas.drawRect(50,850,350,1150,paint);

canvas.drawText("Circle",120,150,paint3);
canvas.drawCircle(200,350,150,paint3);

//draw line
canvas.drawText("Triangle",430,800,paint2);
canvas.drawLine(520,850,460,1150,paint2);
canvas.drawLine(520,850,580,1150,paint2);
canvas.drawLine(460,1150,580,1150,paint2);
```

```
}
```

```
}
```

activity_main.xml

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

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

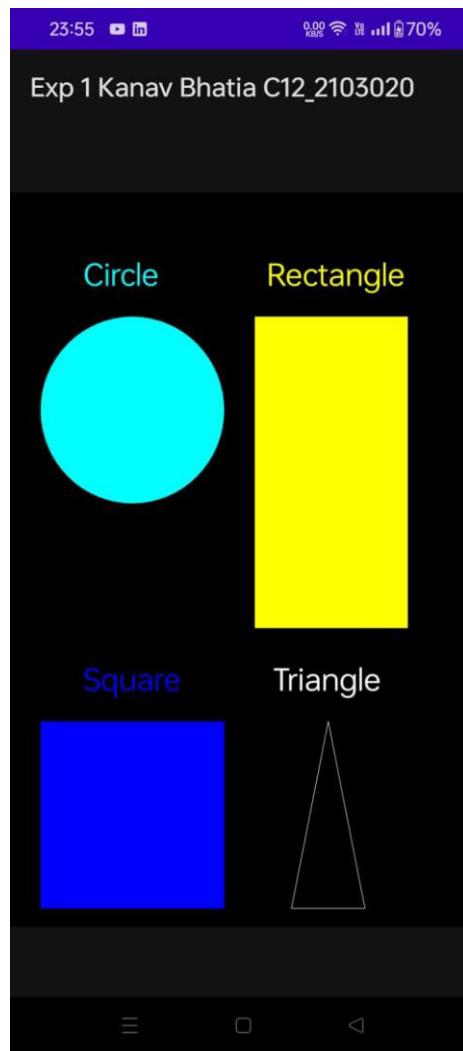
    <ImageView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:id="@+id/imageView"/>

</LinearLayout>
```

AndroidManifest.xml

```
android:label="Exp 1 Kanav Bhatia C12_2103020" (rest all code remains the same..)
```

Output:-



Exp - 2

Aim :- WAP to draw Basic graphical 3D primitives

Theory :-

Function used :-

- 1.) `findViewById` :- a method in Android that is used to retrieve the view with the specified resource ID (R.id.imageView in this case) from the layout.
- 2.) `setStrokeWidth()` :- a method of the paint class in Android, used to set the width of the stroke (outline) for drawing shapes.
- 3.) `setStyle()` :- a method of paint class in Android, used the style of the point, which determines how the fill or stroke shape.
- 4.) `draw3Dcave()` : it is a self defined function that uses `drawLine()` and `drawRect()` to create a 3D cave on the canvas.
- 5.) `draw2Dcuboid()` : It is self defined function that uses `drawRect()` and `drawLine()` to create a 3D cuboid on the canvas.

6.) draw 3D cylinder() :-

It is a self defined function that uses ellipse and line to create a 3D cylinder on the canvas.

7.) Draw 3D cone() :-

The self defined method uses ellipse and lines to draw a 3D cone on the canvas.

Conclusion :-

These method and their usage are essential for customizing the appearance of drawn shapes in an Android App. Though their experiment, we have understood how to draw 3D graphical primitives.

~~Final AT~~

Experiment 2

SHIRISH SHETTY C32 2103164

Aim: Write an android application (WAA) to draw basic graphical 3D primitives.

Program:

MainActivity.java

```
package com.example.Exp2;

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

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Bitmap bg=Bitmap.createBitmap(720,1280, Bitmap.Config.ARGB_8888);
        ImageView img=findViewById(R.id.imageView);
        img.setImageBitmap(bg);

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

SHIRISH SHETTY C322103164

```
    canvas.drawLine(90,900,400,800,paint);
    canvas.drawLine(90,900,50,1100,paint);
    canvas.drawLine(90,900,140,1100,paint);
    canvas.drawLine(140,1100,50,1100,paint);

    canvas.drawLine(140,1100,450,1000,paint);

    canvas.drawLine(450,1000,400,800,paint);

    canvas.drawLine(400,800,360,1000,paint);
    canvas.drawLine(360,1000,450,1000,paint);

    canvas.drawLine(360,1000,50,1100,paint);

}

}
```

activity_main.xml

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

    <ImageView android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:id="@+id/imageView" />          12
</RelativeLayout>
```

```
Paint paint1 = new Paint();
SHIRISH SHETTY C32 2103164
paint1.setColor(Color.RED);

paint1.setTextSize(50);

Paint paint2 = new Paint();

paint2.setColor(Color.BLUE);

paint2.setTextSize(50);

Paint paint = new Paint();

paint.setColor(Color.BLACK);

paint.setTextSize(50);

canvas.drawLine(0,200,300,200,paint);

canvas.drawLine(300,200,300,600,paint);

canvas.drawLine(300,600,0,600,paint);

canvas.drawLine(0,600,0,200,paint);

canvas.drawLine(100,50,400,50,paint);

canvas.drawLine(400,50,400,450,paint);

canvas.drawLine(400,450,100,450,paint);

canvas.drawLine(100,450,100,50,paint);

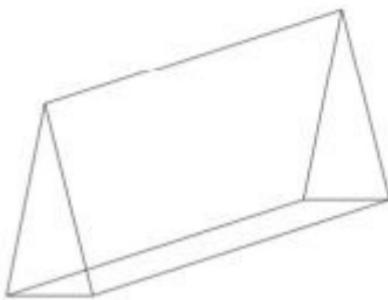
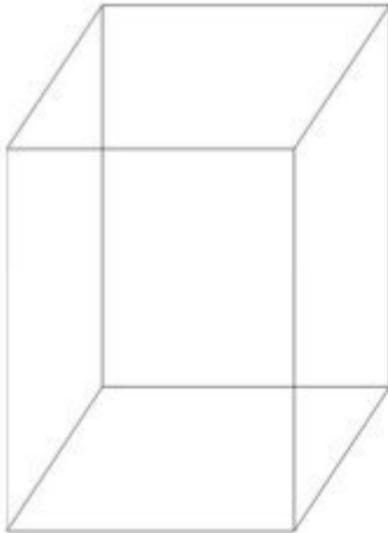
canvas.drawLine(100,50,0,200,paint);

canvas.drawLine(400,50,300,200,paint);

canvas.drawLine(300,600,400,450,paint);

canvas.drawLine(100,450,0,600,paint]3
```

SHIRISH SHETTY C32 2103164



Experiment no. 1

Aim - Write an Android Application to draw.
Basic 2D graphical primitives.

Theory :-

Android Studio is official integrated development environment for Google's Android Operating System, built on Jet Brains IntelliJ IDEA software and designed specifically for Android development, it is available for download on windows, macOS and Linux based operating system. It is a replacement for the Eclipse Android Development Tool as primary IDE for native Android application development.

Android development is usually done in programming languages Java, Kotlin and Flutter.

XML files are used as layout / display screen of your android application. You can add any text / image and various other elements.

We can make 2D graphical primitives in Android using Canvas and Paint objects.

Canvas is a class in Android Graphics framework that provides an area for drawing graphics on screen. It serves as a drawing surface where you can create & manipulate graphical element

such as lines, shapes, text, images.

Paint class in Android is used for styling and controlling how the elements are drawn on a canvas. It defines parameters such as color, stroke, width, text size and style that affect the appearance of graphics drawn on the canvas.

Function:- In Android Studio, functions are essential part of app development, allowing developers to break up their code into reusable pieces. They are particularly in Kotlin, the preferred language for Android development. Function in Android Studio are defined using the fun keyword in Kotlin. and can be passed as arguments and even returned from other functions. They help improve code organization, reusability and readability. By creating functions, developers can avoid duplicating code and make their programs more maintainable. Additionally, functions can be used to define and customize the behavior of various app components.

Conclusion:- learnt about basic shapes, how to draw in android studio.

Paint (A)

Experiment 1

Aim: Write an android application (WAA) to draw basic graphical 2D primitives.

Program:

MainActivity.java

```
package com.example.Exp1;

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);
        bg=Bitmap.createBitmap(750,1180, Bitmap.Config.RGB_565); //create bitmap
        img = findViewById(R.id.imageView);
        img.setImageBitmap(bg);
    }
}
```

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

Paint paint1 = new Paint();
paint1.setColor(Color.YELLOW);
paint1.setTextSize(50);

Paint paint2 = new Paint();
paint2.setColor(Color.WHITE);
paint2.setTextSize(50);

Paint paint = new Paint();
paint.setColor(Color.BLUE);
paint.setTextSize(50);

Paint paint3 = new Paint();
paint3.setColor(Color.CYAN);
paint3.setTextSize(50);

canvas.drawText("Rectangle",420,150,paint1);
canvas.drawRect(400,200,650,700,paint1);

canvas.drawText("Square",120,800,paint);
canvas.drawRect(50,850,350,1150,paint);

canvas.drawText("Circle",120,150,paint3);
canvas.drawCircle(200,350,150,paint3);

//draw line
canvas.drawText("Triangle",430,800,paint2);
canvas.drawLine(520,850,460,1150,paint2);
canvas.drawLine(520,850,580,1150,paint2);
canvas.drawLine(460,1150,580,1150,paint2);
```

```
}
```

```
}
```

activity_main.xml

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

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

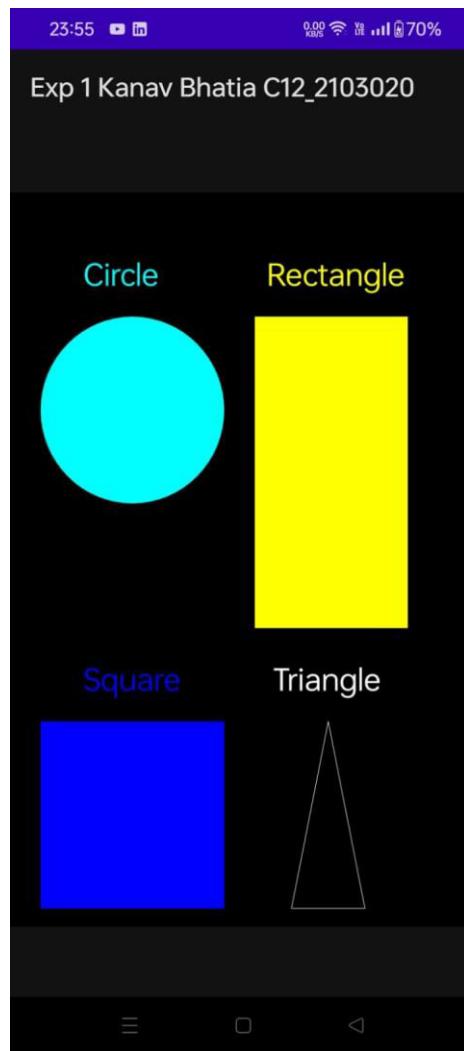
    <ImageView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:id="@+id/imageView"/>

</LinearLayout>
```

AndroidManifest.xml

```
android:label="Exp 1 Kanav Bhatia C12_2103020" (rest all code remains the same..)
```

Output:-



Experiment No - 4

Aim:- WAP to design GUI components using DataBase.

Theory :-

Step-by-Step implementation :-

Step-1: Create a new project

Step-2: - Add permission to access storage in and old manifest xml file.

Step-3: - Work on Activity main.xml file (Section) are UI component that user interact with to trigger actions. They play a crucial role in user interface, serving as entry point for various functionalities within an app.

<EditText> is a versatile UI component for capturing user input. It allows user to enter text, numbers, date types making it a fundamental element for form & data input scenarios.

Step-4: Creating a new java class for performing SQLite operation.

Step 5:- Working with Main Activity java file.

How to View and Locate SQLite Data Base in android studio then;

Step-1: Open android studio project which has SQLite database connection

Step 2:- Connect a device

Step 3:- Search for Device File Explorer in android studio.

Step 4:- Search application package name.

Step 5:- Download the DataBase.

Step 6:- Download SQLite browser.

Step 7:- Search saved database file.

Step 8:- View saved data in tables.

Conclusion :- Hence, we had successfully created a form and stored data in database.

Rajiv (AT)

Program:

MainActivity.java

```
package com.example.Exp4;

import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {

    private EditText courseNameEdt, courseTracksEdt, courseDurationEdt,
courseDescriptionEdt;
    private Button addCourseBtn;
    private DBHandler dbHandler;

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

        courseNameEdt = findViewById(R.id.idEdtCourseName);
        courseTracksEdt = findViewById(R.id.idEdtCourseTracks);

        courseDurationEdt = findViewById(R.id.idEdtCourseDuration);
```

```
courseDescriptionEdt = findViewById(R.id.idEdtCourseDescription);
addCourseBtn = findViewById(R.id.idBtnAddCourse);

dbHandler = new DBHandler(MainActivity.this);

addCourseBtn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {

        String courseName = courseNameEdt.getText().toString();
        String courseTracks = courseTracksEdt.getText().toString();
        String courseDuration = courseDurationEdt.getText().toString();
        String courseDescription = courseDescriptionEdt.getText().toString();

        if (courseName.isEmpty() && courseTracks.isEmpty() &&
courseDuration.isEmpty() && courseDescription.isEmpty()) {
            Toast.makeText(MainActivity.this, "Please enter all the data..",
Toast.LENGTH_SHORT).show();
            return;
        }

        dbHandler.addNewCourse(courseName, courseDuration,
courseDescription, courseTracks);

        Toast.makeText(MainActivity.this, "Course has been added.",
Toast.LENGTH_SHORT).show();

        courseNameEdt.setText("");
        courseDurationEdt.setText("");
        courseTracksEdt.setText("");
    }
});
```

```
        courseDescriptionEdt.setText("");
    }
});
}
}
```

DBHandler.java

```
package com.example.Exp4;

import android.content.ContentValues;
import android.content.Context;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;

public class DBHandler extends SQLiteOpenHelper {

    private static final String DB_NAME = "kanavDB";

    private static final int DB_VERSION = 1;

    private static final String TABLE_NAME = "mycourses";

    private static final String ID_COL = "id";

    private static final String NAME_COL = "name";

    private static final String DURATION_COL = "duration";

    private static final String DESCRIPTION_COL = "description";
```

```
private static final String TRACKS_COL = "tracks";\n\npublic DBHandler(Context context) {\n    super(context, DB_NAME, null, DB_VERSION);\n}\n\n@Override\npublic void onCreate(SQLiteDatabase db) {\n    String query = "CREATE TABLE " + TABLE_NAME + "("\n        + ID_COL + " INTEGER PRIMARY KEY AUTOINCREMENT, "\n        + NAME_COL + " TEXT,"\n        + DURATION_COL + " TEXT,"\n        + DESCRIPTION_COL + " TEXT,"\n        + TRACKS_COL + " TEXT);"\n\n    db.execSQL(query);\n}\n\npublic void addNewCourse(String courseName, String courseDuration, String\ncourseDescription, String courseTracks) {\n\n    SQLiteDatabase db = this.getWritableDatabase();\n\n    ContentValues values = new ContentValues();\n\n    values.put(NAME_COL, courseName);\n    values.put(DURATION_COL, courseDuration);\n    values.put(DESCRIPTION_COL, courseDescription);\n}
```

```
values.put(TRACKS_COL, courseTracks);

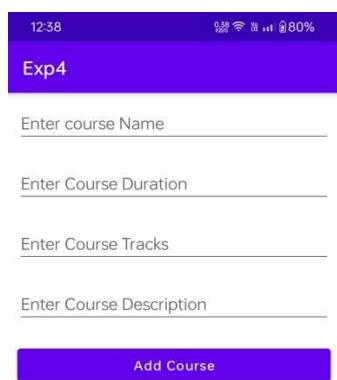
db.insert(TABLE_NAME, null, values);
db.close();
}

@Override
public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
    db.execSQL("DROP TABLE IF EXISTS " + TABLE_NAME);
    onCreate(db);
}
}
```

AndroidManifest.xml

```
android:label="Exp 4 Kanav Bhatia C12_2103020"
```

Output:-





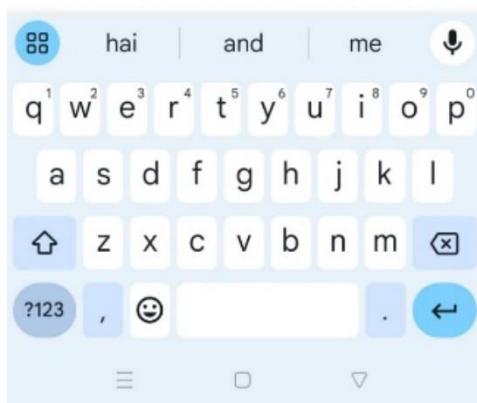
CSS

4 months

AES, DES

Cryptography

Add Course





Enter course Name _____

Enter Course Duration _____

Enter Course Tracks _____

Enter Course Description _____

Add Course

Course has been added.

☰ ☐ ◀

kanavDB_1.db ► mycourses						
Tables (3)		id	name	duration	description	tracks
▶	android_metadata	1	spcc	2hrs	lexical analysis	chp1
▶	mycourses	2	CSS	4 months	Cryptography	AES,DES
▶	sqlite_sequence					

Experiment 5

Aim :- Write a program to develop an EMI calculator application

Theory :-

In this article, I will show how to create a simple EMI calculator, Android application using Android studio.

Requirements :-

- (i) Android studio
- (ii) little bit XML and Java knowledge
- (iii) Android emulator

Step to be followed :-

Step 1 :- Open android studio and start a new android studio project

Step 2 - You can choose your app name & location

Step 3 - Add the path

Step 4 :- Finish the setup

Formula :-

$$E = \frac{P \cdot R \cdot (1+r)^n}{(1+r)^n - 1}$$

Step 5 :- Add the necessary code in main.xml

~~<LinearLayout>~~: is a view group that arranges the children element linearly either vertically or horizontally, based on the specified oriented attribute

~~<EditText>~~ : represent a text input field where the user can enter information

~~<Button>~~ represent a clickable button

Conclusion :-

Hence we were able to make EMI calculator in Android studio from scratch.

Final (A)

Code :

activity_main.xml :

```
<?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"
    android:padding="16dp">

    <EditText
        android:id="@+id/principalAmount"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Principal Amount"
        android:inputType="numberDecimal"/>

    <EditText
        android:id="@+id/interestRate"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Annual Interest Rate (%)"
        android:inputType="numberDecimal"
        android:layout_marginTop="8dp"/>

    <EditText
        android:id="@+id/loanTenure"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Loan Tenure (Years)"
        android:inputType="number"
        android:layout_marginTop="8dp"/>

    <Button
        android:id="@+id/calculateEMI"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Calculate EMI"
        android:layout_gravity="center_horizontal"
        android:layout_marginTop="16dp"/>

    <TextView
        android:id="@+id/emiResult"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="18sp"
        android:layout_gravity="center_horizontal"
        android:layout_marginTop="24dp"/>

</LinearLayout>
```

MainActivity.java :

```

package com.example.myapplication5;

import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import java.text.DecimalFormat;

public class MainActivity extends AppCompatActivity {

    private EditText principalAmount, interestRate, loanTenure;
    private Button calculateEMI;
    private TextView emiResult;

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

        principalAmount = findViewById(R.id.principalAmount);
        interestRate = findViewById(R.id.interestRate);
        loanTenure = findViewById(R.id.loanTenure);
        calculateEMI = findViewById(R.id.calculateEMI);
        emiResult = findViewById(R.id.emiResult);

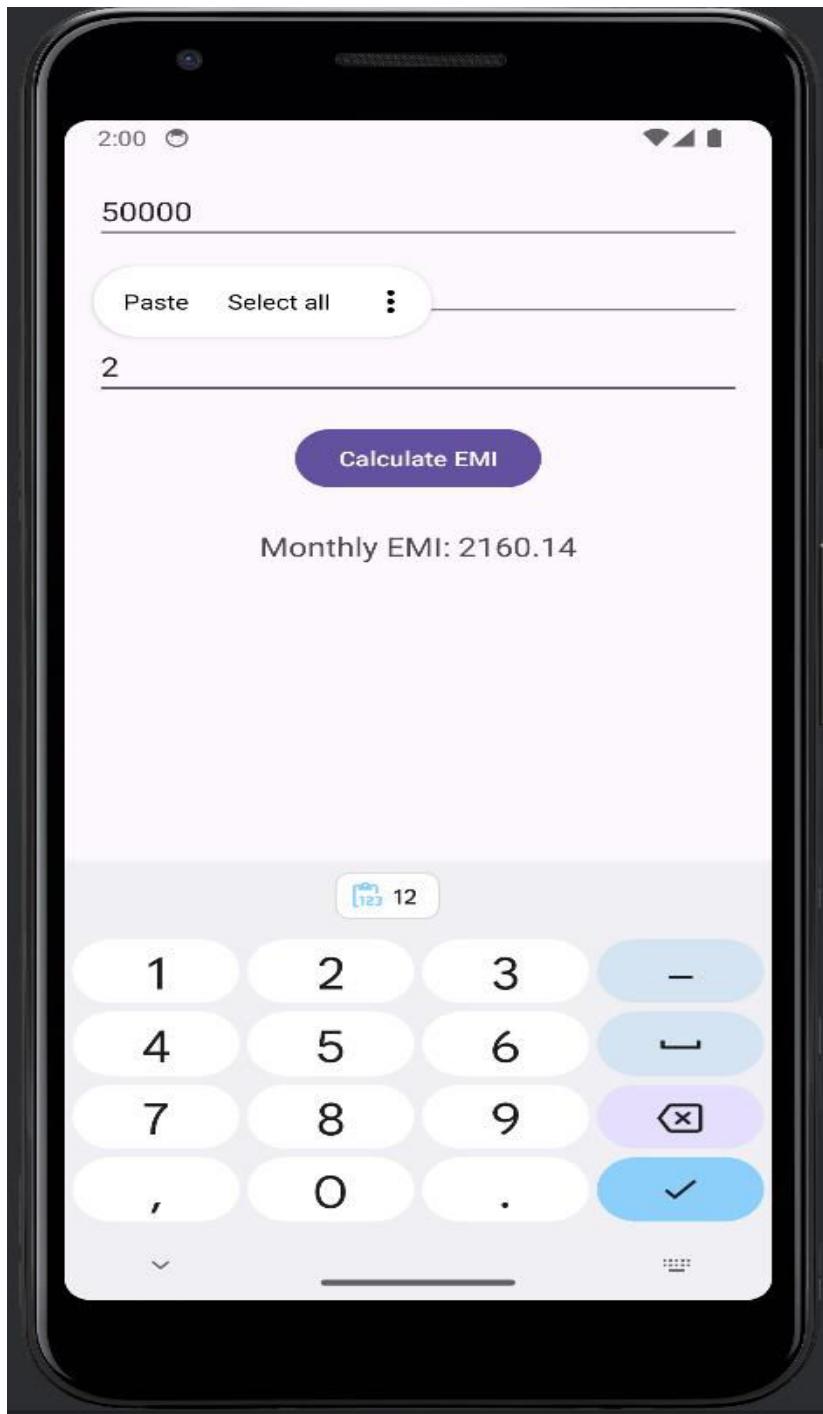
        calculateEMI.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                double p =
Double.parseDouble(principalAmount.getText().toString());
                double r =
Double.parseDouble(interestRate.getText().toString()) / 12 / 100;
                double n =
Double.parseDouble(loanTenure.getText().toString()) * 12;

                double emi = (p * r * Math.pow(1 + r, n)) / (Math.pow(1 + r,
n) - 1);

                DecimalFormat df = new DecimalFormat("###.##");
                emiResult.setText("Monthly EMI: " + df.format(emi));
            }
        });
    }
}

```

Output :



Experiment No - 6

Aim :- Write an android application that create an alert on receiving a message.

Theory :-

-Build a basic notification :-

A notification is its most basic & compact form also known as collapsed form displays an icon, a title and a small amount of text content. This section shows how to create a notification that can tap to launch an activity in your application.

-Declare the runtime permission.

Android 13 and higher supports a runtime permission for posting non-except notification from an app

The permission that you need to declare in your app's manifest file appears in the following.

- Set the notification content

To get started set the notification content channel using a notification ~~Compact~~, ~~Builder~~ object. The following example shows how to create notification with following :-

A small icon, set by set small icon(). This the only

user-visible content that is required.

- The body text set by `setContentText()`

- Create a channel & set the importance

Before you can deliver an notification on Android 8.0 and later, register your app's notification channel with system by passing an instance of `NotificationChannel` to `CreateNotificationChannel()`

- set the notification's top action.

Every notification must respond to a top, usually to open an activity in your app that corresponds to notification. To do so specify a content intent defined with a pending Intent object and pass it to `setContentIntent()`.

- Show the notification

To make notification appear, call `NotificationManagerCompat.notify()` passing it a unique ID for notifications.

Conclusion - Once successfully displayed on receiving a message.

Final (A)

Code :

activity_main.xml :

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

    <Button
        android:id="@+id	btnShowAlert"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Simulate Message Reception"
        android:layout_centerInParent="true"/>
</RelativeLayout>
```

MainActivity.java :

```
package com.example.myapplication6;

import androidx.appcompat.app.AlertDialog;
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;

public class MainActivity extends AppCompatActivity {

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

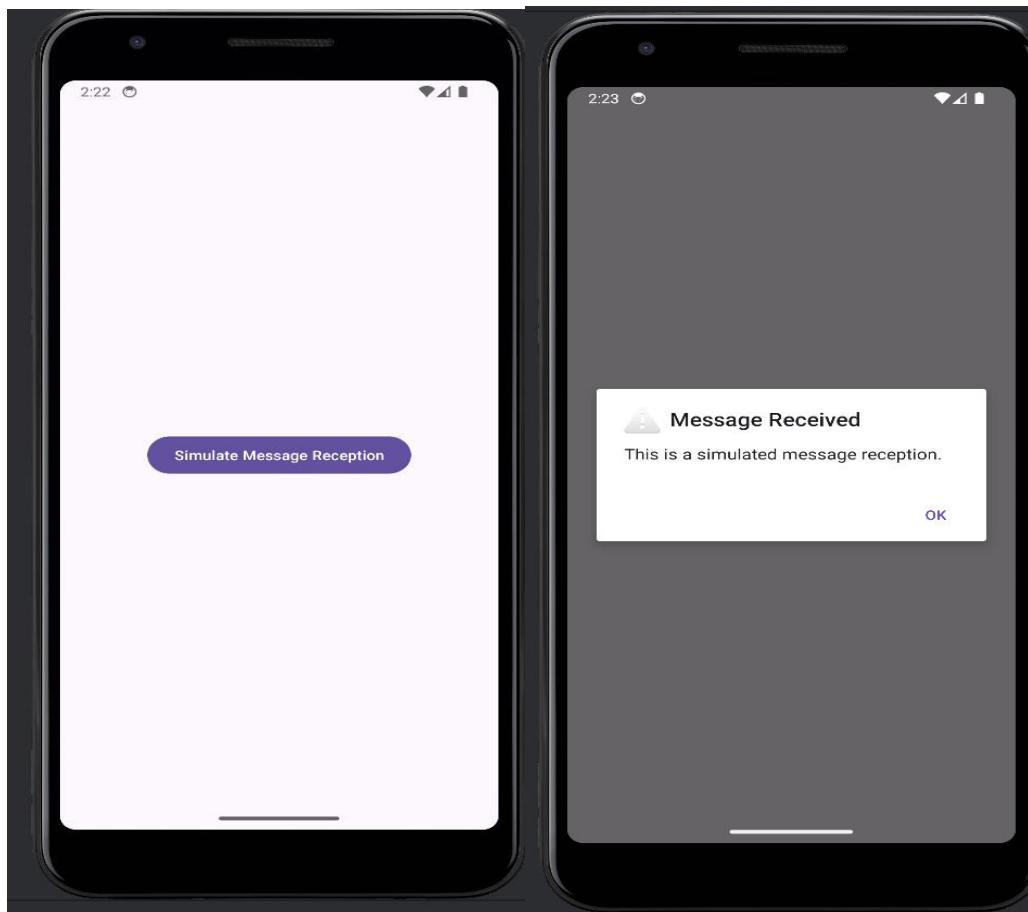
        Button btnShowAlert = findViewById(R.id.btnShowAlert);
        btnShowAlert.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                showAlert();
            }
        });
    }

    private void showAlert() {
        new AlertDialog.Builder(MainActivity.this)
            .setTitle("Message Received")
            .setMessage("This is a simulated message reception.")
    }
}
```

```
        // Specifying a listener allows you to take an action before
        // dismissing the dialog.
        // The dialog is automatically dismissed when a dialog
button is clicked.
        .setPositiveButton(android.R.string.yes, null)

        // A null listener allows the button to dismiss the dialog
and take no further action.
        .setIcon(android.R.drawable.ic_dialog_alert)
        .show();
    }
}
```

Output :



Experiment No 7

AIM :- Built an Android Application to develop basic math calculator

Theory :- This experiment focusses on Android Application on a basic calculator using Java in Android studio Code. It allows user to perform arithmetic operation on two input number & also calculate the square root of a single input number.

If Built in Function are :-

- 1) onCreate() :- This method is called when the activity is created. It initializes UI element and set click listeners for button.
- 2) performCalculation(char operator) :- This method performs arithmetic calculator based on operator (+, -, *, /). It retrieves input values from Edit Text fields, checks for empty input, perform calculation & display result in TextView.
- 3) CalculateSquareRoot() :- This method calculates square root of number entered in one of Edit Text fields. It retrieves input value, checks for empty input value, checks for empty input, calculates square root using Math.sqrt() functions and display in TextView.
- u) DecimalFormat - This class is used to format numerical values.

in this case, to display the result with maximum of two decimal places.

5) `Find View By Id()` :- This method is used to bind and return a view from layout based on its ID.

6) `setOnClickListener()` :- This method is used to set a click listener on button to perform a specific action when button is clicked.

7) `Toast.makeText()` :- This method creates a toast message, a small pop-up message that briefly appears on screen to provide feedback to user.

8) `<EditText>` : Defines an input field where user enters numeric values.

9) `<Button>` : Define a clickable button for performing arithmetic operation or other action.

10) `<Text View>` : - Define a non editable text field for displaying result or other information.

11) `constraintLayout` :- layout manager used to create flexible layout with constraints.

Conclusion :- The provided code implements a basic calculator app in Android studio using Java. It helps in grasping essential concepts in App development.

Final (A)

Aim: Write an android application for basic operations on calculator.

Program:

MainActivity.java

```
package com.example.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(v -> {
    int firstNumber = Integer.parseInt(etFirstNumber.getText().toString());
    int secondNumber =
        Integer.parseInt(etSecondNumber.getText().toString());
    int result = firstNumber + secondNumber;
    tvResult.setText(result+" ");
});
btnSubtract.setOnClickListener(v -> {
    int firstNumber = Integer.parseInt(etFirstNumber.getText().toString());
    int secondNumber =
        Integer.parseInt(etSecondNumber.getText().toString());
    int result = firstNumber - secondNumber;
    tvResult.setText(result+" ");
});
btnMultiply.setOnClickListener(v -> {
    int firstNumber = Integer.parseInt(etFirstNumber.getText().toString());
    int secondNumber =
        Integer.parseInt(etSecondNumber.getText().toString());
    int result = firstNumber * secondNumber;
    tvResult.setText(result+" ");
});
btnDivide.setOnClickListener(v -> {
    int firstNumber = Integer.parseInt(etFirstNumber.getText().toString());
    int secondNumber =
        Integer.parseInt(etSecondNumber.getText().toString());
    int result = firstNumber / secondNumber;
});
```

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

    });

    btnClear.setOnClickListener(v -> {

        etFirstNumber.setText("");

        etSecondNumber.setText("");

        tvResult.setText("");

    });

}

activity_main.xml
```

```
<?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="35dp"
    android:textSize="30sp"
    android:textColor="#000000"
    android:text="Result: "
    android:layout_marginTop="50dp"
    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"/>
```

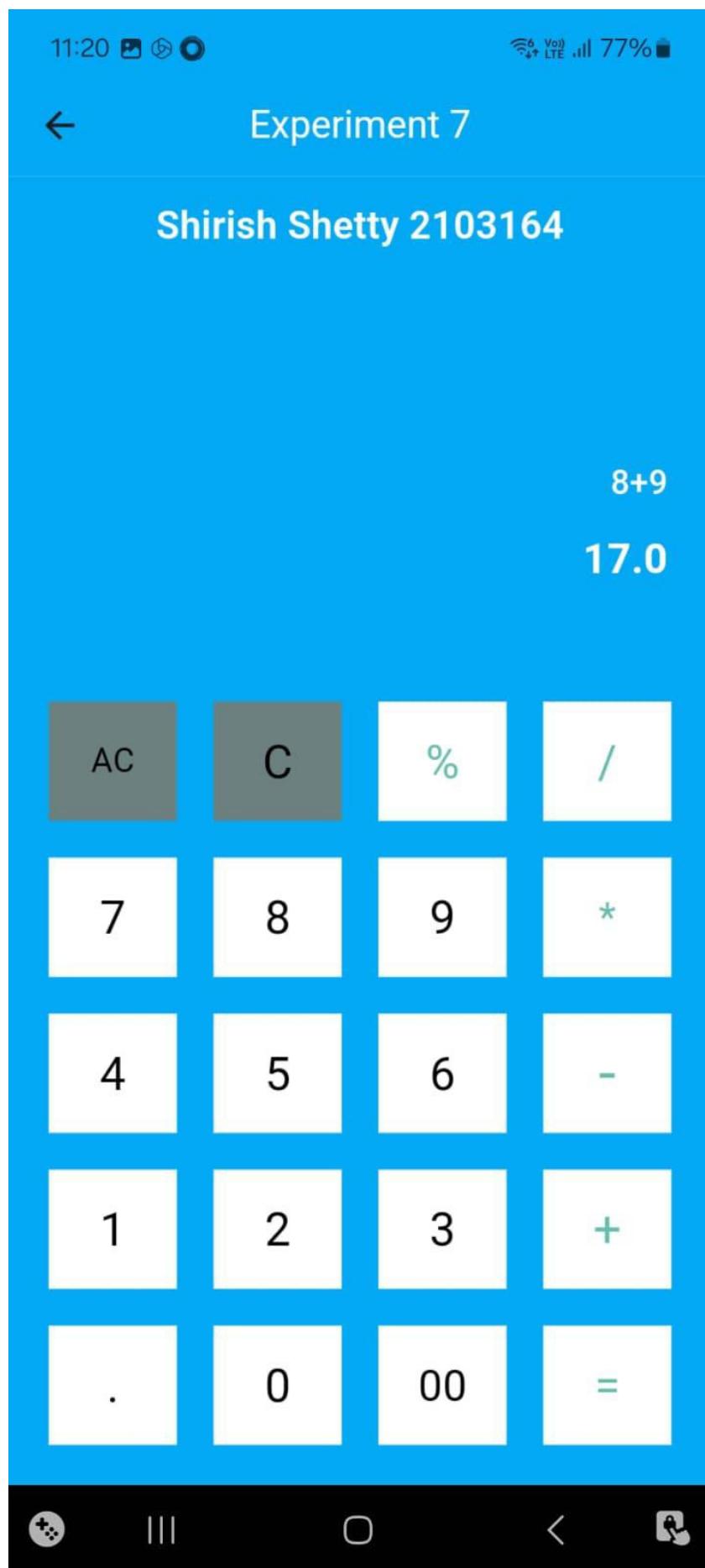
```
<Button  
    android:id="@+id	btn_subtract"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text=""  
  
    -  
    "  
  
        android:layout_marginLeft="20dp"  
        android:layout_marginRight="20dp"/>  
  
</LinearLayout>  
  
<LinearLayout  
    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"  
    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>
```

AndroidManifest.xml

```
    android:label="Exp 7 Kanav Bhatia C12_2103020"
```

Output:-



Experiment No: 8

Aim:- Write a program to demonstrate cellular frequency reuse.

Theory :-

~~Frequency Reuse is scheme in which allocation and reuse of channels throughout a coverage region is done.~~

~~each cellular base station is allocation a group of radio channel or frequency subbands to be used within a small geographic area known as cell. The shape of cell is hexagonal.~~

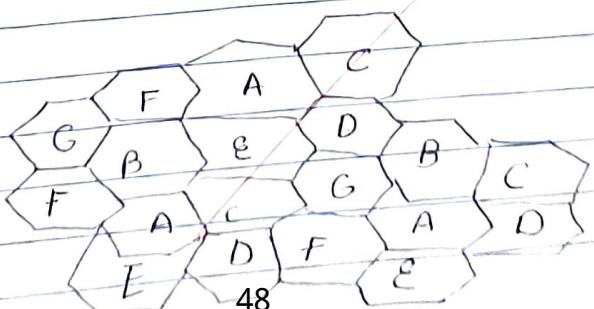
~~The process of selecting and allocating the frequency subbands for all of cellular base station within a system is called frequency reuse or frequency Planning.~~

~~Advantages of frequency Reuse :-~~

~~It improves 'Quality of service' (QoS)~~

~~In frequency Reuse Scheme, total bandwidth is divided into different sub-bands that are used by the cells~~

~~frequency reuse scheme allows operator to reuse the same frequency at different cell sites~~



Cell with same filter, uses same set of channels group or frequencies sub-band.

To find the total number of channels allocated to a cell,

$S = \text{Total number of duplex channel available to use.}$

$K = \text{Channel allocated to each cell } (K < S)$

$N = \frac{\text{Total}}{\text{Channels allocated to each cell}} \leftarrow \text{number of cells or cluster size.}$

Thus $S = KN$

Frequency Reuse Factor = $1/N$

The value of N is calculated by formula:-

$$N = I^2 + I * J + J^2$$

where I, J :- Positive integers indicating posn of cell

N :- Total number of cells / size of cluster.

If a Cluster is duplicated or repeated M times, then capacity C will be

$$C = MKN = MS \quad (\because S = KN)$$

Conclusion :- As the above experiment, hasn't about cellular frequency reuse

Final (AT)

Code:

Exp8FrequencyReuse.py :

```
#!/usr/bin/python
```

```
from math import *
```

```
# import everything from Tkinter module
from tkinter import *
```

```
# Base class for Hexagon shape
```

```
class Hexagon(object):
```

```
    def __init__(self, parent, x, y, length, color, tags):
        self.parent = parent
        self.x = x
        self.y = y
        self.length = length
        self.color = color
        self.size = None
        self.tags = tags
        self.draw_hex()
```

```
# draw one hexagon
```

```
def draw_hex(self):
    start_x = self.x
    start_y = self.y
    angle = 60
    coords = []
    for i in range(6):
        end_x = start_x + self.length * cos(radians(angle * i))
        end_y = start_y + self.length * sin(radians(angle * i))
        coords.append([start_x, start_y])
        start_x = end_x
        start_y = end_y
    self.parent.create_polygon(coords[0][0],
                               coords[0][1],
                               coords[1][0],
                               coords[1][1],
                               coords[2][0],
                               coords[2][1],
                               coords[3][0],
                               coords[3][1],
                               coords[4][0],
                               coords[4][1],
                               coords[5][0],
                               coords[5][1],
                               fill=self.color,
                               outline="black",
```

```
tags=self.tags)

# class holds frequency reuse logic and related methods
class FrequencyReuse(Tk):

    CANVAS_WIDTH = 800
    CANVAS_HEIGHT = 650
    TOP_LEFT = (20, 20)
    BOTTOM_LEFT = (790, 560)
    TOP_RIGHT = (780, 20)
    BOTTOM_RIGHT = (780, 560)

    def __init__(self, cluster_size, columns=16, rows=10, edge_len=30):
        Tk.__init__(self)
        self.textbox = None
        self.curr_angle = 330
        self.first_click = True
        self.reset = False
        self.edge_len = edge_len
        self.cluster_size = cluster_size
        self.reuse_list = []
        self.all_selected = False
        self.curr_count = 0
        self.hexagons = []
        self.co_cell_endp = []
        self.reuse_xy = []
```

```
self.canvas = Canvas(self,
                     width=self.CANVAS_WIDTH,
                     height=self.CANVAS_HEIGHT,
                     bg="#4dd0e1")

self.canvas.bind("<Button-1>", self.call_back)
self.canvas.focus_set()
self.canvas.bind('<Shift-R>', self.resets)
self.canvas.pack()
self.title("Frequency reuse and co-channel selection")
self.create_grid(16, 10)
self.create_textbox()
self.cluster_reuse_calc()

# show lines joining all co-channel cells

def show_lines(self):
    # center(x,y) of first hexagon
    approx_center = self.co_cell_endp[0]
    self.line_ids = []
    for k in range(1, len(self.co_cell_endp)):

        end_xx = (self.co_cell_endp[k])[0]
        end_yy = (self.co_cell_endp[k])[1]

        # move i^th steps
```

```
        l_id = self.canvas.create_line(approx_center[0],  
approx_center[1],  
                                end_xx,  
                                end_yy)  
  
        if j == 0:  
  
            self.line_ids.append(l_id)  
            dist = 0  
  
        elif i >= j and j != 0:  
  
            self.line_ids.append(l_id)  
            dist = j  
  
            # rotate counter-clockwise and move j^th step  
  
            l_id = self.canvas.create_line(  
                end_xx, end_yy, end_xx + self.center_dist *  
dist *  
                                cos(radians(self.curr_angle - 60)),  
                                end_yy + self.center_dist * dist *  
sin(radians(self.curr_angle - 60)))  
  
            self.line_ids.append(l_id)  
  
        self.curr_angle -= 60  
  
  
def create_textbox(self):  
    txt = Text(self.canvas,  
              width=80,  
              height=1,  
              font=("Helvetica", 12),  
              padx=10,
```

```
        pady=10)

    txt.tag_configure("center", justify="center")
    txt.insert("1.0", "Select a Hexagon")
    txt.tag_add("center", "1.0", "end")
    self.canvas.create_window((0, 600), anchor='w', window=txt)
    txt.config(state=DISABLED)
    self.textbox = txt

def resets(self, event):
    if event.char == 'R':
        self.reset_grid()

# clear hexagonal grid for new i/p

def reset_grid(self, button_reset=False):
    self.first_click = True
    self.curr_angle = 330
    self.curr_count = 0
    self.co_cell_endp = []
    self.reuse_list = []
    for i in self.hexagons:
        self.canvas.itemconfigure(i.tags, fill=i.color)

    try:
        self.line_ids
    except AttributeError:
```

```
    pass

else:
    for i in self.line_ids:
        self.canvas.after(0, self.canvas.delete, i)
    self.line_ids = []

if button_reset:
    self.write_text("Select a Hexagon")

# create a grid of Hexagons
def create_grid(self, cols, rows):
    size = self.edge_len
    for c in range(cols):
        if c % 2 == 0:
            offset = 0
        else:
            offset = size * sqrt(3) / 2
        for r in range(rows):
            x = c * (self.edge_len * 1.5) + 50
            y = (r * (self.edge_len * sqrt(3))) + offset + 15
            hx = Hexagon(self.canvas, x, y, self.edge_len,
                         "#fafafa",
                         "{}{},{}".format(r, c))
            self.hexagons.append(hx)
```

```
# calculate reuse distance, center distance and radius of the
hexagon

def cluster_reuse_calc(self):
    self.hex_radius = sqrt(3) / 2 * self.edge_len
    self.center_dist = sqrt(3) * self.hex_radius
    self.reuse_dist = self.hex_radius * sqrt(3 * self.cluster_size)

def write_text(self, text):
    self.textbox.config(state=NORMAL)
    self.textbox.delete('1.0', END)
    self.textbox.insert('1.0', text, "center")
    self.textbox.config(state=DISABLED)

#check if the co-channels are within visible canvas

def is_within_bound(self, coords):
    if self.TOP_LEFT[0] < coords[0] < self.BOTTOM_RIGHT[0] \
        and self.TOP_RIGHT[1] < coords[1] < self.BOTTOM_RIGHT[1]:
        return True
    return False

#gets called when user selects a hexagon
#This function applies frequency reuse logic in order to
#figure out the positions of the co-channels
def call_back(self, evt):
```

```
selected_hex_id = self.canvas.find_closest(evt.x, evt.y)[0]
hexagon = self.hexagons[int(selected_hex_id - 1)]
s_x, s_y = hexagon.x, hexagon.y
approx_center = (s_x + 15, s_y + 25)

if self.first_click:
    self.first_click = False
    self.write_text(
        """Now, select another hexagon such
        that it should be a co-cell of
        the original hexagon."""
    )
    self.co_cell_endp.append(approx_center)
    self.canvas.itemconfigure(hexagon.tags, fill="green")

for _ in range(6):
    end_xx = approx_center[0] + self.center_dist * i *
    cos(
        radians(self.curr_angle))
    end_yy = approx_center[1] + self.center_dist * i *
    sin(
        radians(self.curr_angle))

    reuse_x = end_xx + (self.center_dist * j) * cos(
        radians(self.curr_angle - 60))
```

```
reuse_y = end_yy + (self.center_dist * j) * sin(  
    radians(self.curr_angle - 60))  
  
if not self.is_within_bound((reuse_x, reuse_y)):  
    self.write_text(  
        """co-cells are exceeding canvas  
boundary.  
Select cell in the center"""  
    )  
    self.reset_grid()  
    break  
  
if j == 0:  
    self.reuse_list.append(  
        self.canvas.find_closest(end_xx,  
end_yy)[0])  
elif i >= j and j != 0:  
    self.reuse_list.append(  
        self.canvas.find_closest(reuse_x,  
reuse_y)[0])  
  
    self.co_cell_endp.append((end_xx, end_yy))  
    self.curr_angle -= 60  
  
else:  
    curr = self.canvas.find_closest(s_x, s_y)[0]
```

```
if curr in self.reuse_list:  
    self.canvas.itemconfigure(hexagon.tags,  
        fill="green")  
  
    self.write_text("Correct! Cell {} is a co-  
cell.".format(  
        hexagon.tags))  
  
    if self.curr_count == len(self.reuse_list) - 1:  
        self.write_text("Great! Press Shift-R to  
restart")  
  
        self.show_lines()  
  
        self.curr_count += 1  
  
  
else:  
    self.write_text("Incorrect! Cell {} is not a co-  
cell.".format(  
        hexagon.tags))  
  
    self.canvas.itemconfigure(hexagon.tags, fill="red")  
  
  
if __name__ == '__main__':  
    print(  
        """Enter i & j values. common (i,j) values are:  
        (1,0), (1,1), (2,0), (2,1), (3,0), (2,2)"""  
    )  
  
    i = int(input("Enter i: "))  
    j = int(input("Enter j: "))  
  
    if i == 0 and j == 0:
```

```
    raise ValueError("i & j both cannot be zero")

elif j > i:
    raise ValueError("value of j cannot be greater than i")

else:
    N = (i**2 + i * j + j**2)
    print("N is {}".format(N))

freqreuse = FrequencyReuse(cluster_size=N)
freqreuse.mainloop()
```

Output:

(base) PS C:\Users\Shirish\OneDrive\Desktop\MCC Exp Documents>
python Exp8code.py

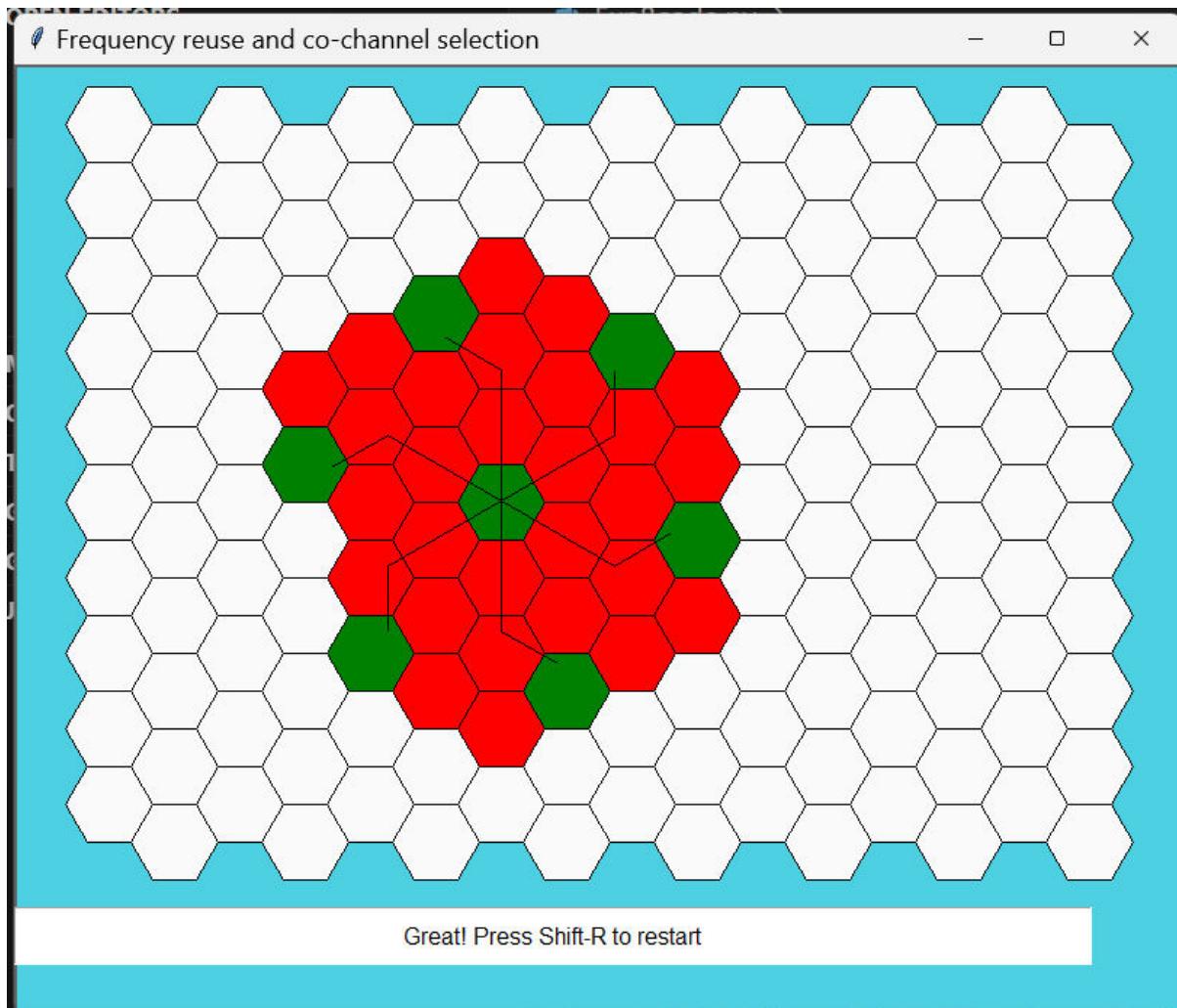
Enter i & j values. common (i,j) values are:

(1,0), (1,1), (2,0), (2,1), (3,0), (2,2)

Enter i: 2

Enter j: 1

N is 7



Experiment No:- 9

Aim:- Write a program to explain concept of DSSS.

Theory:-

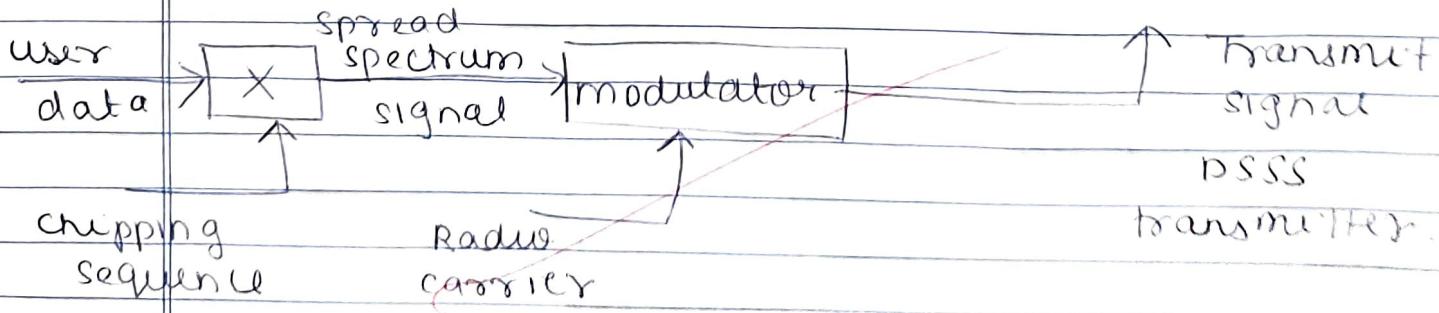
Spread Spectrum include techniques involving spreading bandwidth needed to transmit data, reducing narrowband interference

Direct sequence spread spectrum (DSSS)
system take user bit stream & perform XOR with so called chipping sequence

Wireless system use sequence 1011011000, called as Barker's code These Barker codes called as Barker's code. These Barker codes exhibit a good robustness against interference and insensitivity to multi-path propagation

The first step in DSSS transmitter is spreading of user data with chipping sequence (digital modulation)

The spread signal is then modulated within a radio carrier (radio modulation)

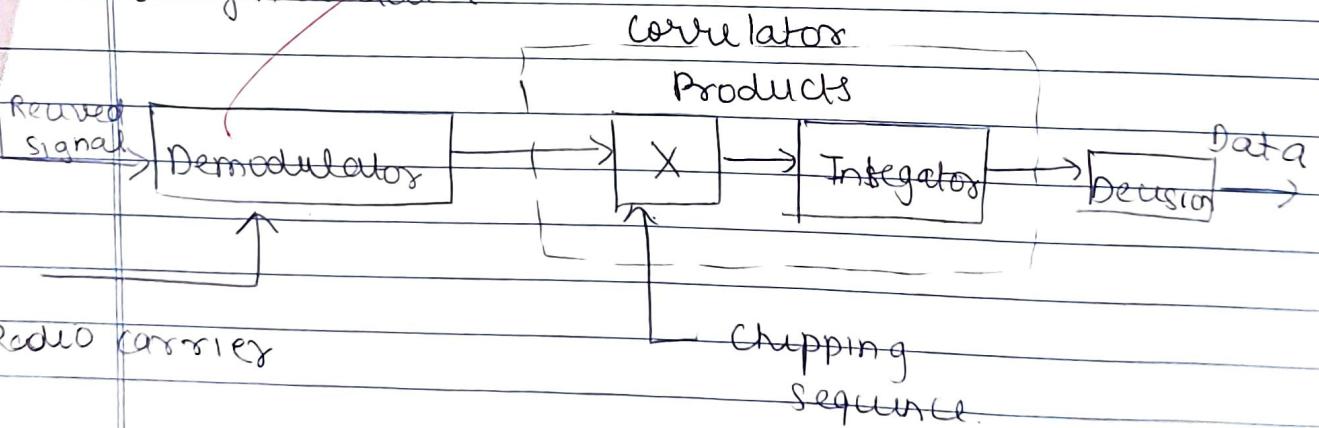


Assuming a user signal with bandwidth of 1MHz, spreading it with 11-chip Barker code would result in signal with 11MHz bandwidth.

The radio carrier then shift it to carrier frequency eg 2.4 GHz, this signal is then transmitted

DSSS Receiver:-

The Receiver has perform inverse functions of two transmitter modulation steps. However, noise and multi-path propagation requires additional mechanism to reconstruct the original data



The receiver has to Known Chipping sequence, sequence have to be precisely synchronized hence receiver calculator products of chip with incoming signal

This comprises XOR operation. An integer adds all these products

calculating product of chips & signal,
adding product in a integrator is also
called correlation.

Finally, in each bit period a decision
unit samples seems generated by integrated
and decides if this sum represent a binary
1 or a 0.

Eg:- Transmission of user data 01

Soln

User Data

0 1

XOR with Barker code ~~1 0 1 1 0 1 1 1 0 0 0 0 1 0 1 1 0 1 1 1 0 0 0~~

Spread Spectrum ~~1 0 1 1 0 1 1 1 0 0 0 0 1 0 0 1 0 0 0 0 1 1~~

They are concentrated to 22 digits and
Sent. At receiver, perform XOR operations
on received signal with same Barker code

Received signal ~~1 0 1 1 0 1 1 1 0 0 0 0 1 0 0 1 0 0 0 1 1 1~~ nce

XOR with Barker code ~~1 0 1 1 0 1 1 1 0 0 0 1 0 1 1 0 1 1 1 0 0 0~~

Result ~~0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1~~ thin

Now result is given to integer which performs
sum to products sum of products for
first part is 0 and next is 11. The decision
unit maps sum less than 4 to binary 0,
and sum larger than 7 to binary 1

This constitutes the original user data i.e. 01

Conclusion:-

Thus, understood concept Direct Sequence spread Spectrum (DSSS) and its significance in mobile computing

~~Bring~~ At

Code:

Exp9DSSS.java :

```
import java.util.Arrays;
```

```
public class Exp9DSSS {
```

```
    public static void main(String[] args) {
```

```
        // Original data signal (binary representation)
```

```
        // System.out.print("Enter the length of data signal: ");
```

```
        int[] dataSignal = { 0, 1 };
```

```
        // Spreading code (PN sequence) Barker's code
```

```
        int[] spreadingCode = { 1, 0, 1, 1, 0, 1, 1, 1, 0, 0, 0 };
```

```
        // Spread the data signal using DSSS
```

```
        int[] spreadSignal = spreadDSSS(dataSignal, spreadingCode);
```

```
        // Display the results
```

```
        System.out.println("Original Data Signal: " +  
        Arrays.toString(dataSignal));
```

```
        System.out.println("Spreading Code (PN Sequence): " +  
        Arrays.toString(spreadingCode));
```

```
System.out.println("Spread Signal: " + Arrays.toString(spreadSignal));  
  
// Recover the original signal by despread  
int[] recoveredSignal = despreadDSSS(spreadSignal, spreadingCode);  
  
// Display the recovered signal  
System.out.println("Recovered Signal: " +  
Arrays.toString(recoveredSignal));  
}  
  
private static int[] spreadDSSS(int[] dataSignal, int[] spreadingCode) {  
    int[] spreadSignal = new int[dataSignal.length *  
spreadingCode.length];  
    for (int i = 0; i < dataSignal.length; i++) {  
        for (int j = 0; j < spreadingCode.length; j++) {  
            spreadSignal[i * spreadingCode.length + j] = dataSignal[i] ^  
spreadingCode[j];  
        }  
    }  
    return spreadSignal;  
}  
  
private static int[] despreadDSSS(int[] spreadSignal, int[]  
spreadingCode) {  
    int length = spreadSignal.length / spreadingCode.length;  
    int[] recoveredSignal = new int[length];  
}
```

```
for (int i = 0; i < length; i++) {  
    int sum = 0;  
    for (int j = 0; j < spreadingCode.length; j++) {  
        sum += spreadSignal[i * spreadingCode.length + j] ^  
spreadingCode[j];  
    }  
    System.out.print("Addition of " + " bit " + (i + 1) + " : " + sum);  
    recoveredSignal[i] = (sum > 7) ? 1 : 0;  
    if (sum > 7) {  
        System.out.println(", Since sum is more than 7, it is converted to  
1");  
    } else  
        System.out.println(", Since sum is less than 4, it is converted to  
0");  
    }  
    return recoveredSignal;  
}
```

Output:

```
(base) PS C:\Users\Shirish\OneDrive\Desktop\MCC Exp Documents>
cd "c:\Users\Rishab\OneDrive\Desktop\MCC Exp Documents\" ; if
(?) { javac Exp9DSSS.java } ; if (?) { java Exp9DSSS }
```

Original Data Signal: [0, 1]

Spreading Code (PN Sequence): [1, 0, 1, 1, 0, 1, 1, 1, 0, 0, 0]

Spread Signal: [1, 0, 1, 1, 0, 1, 1, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 1, 1, 1]

Addition of bit 1 : 0, Since sum is less than 4, it is converted to 0

Addition of bit 2 : 11, Since sum is more than 7, it is converted to 1

Recovered Signal: [0, 1]

```
(base) PS C:\Users\Shirish\OneDrive\Desktop\MCC Exp Documents>
```

Experiment-10

Aim:- Write a Program to implement A3, A5, A8, GSM Security Algorithm

Theory :-

Implementing the A3/A5/A8 GSM security algorithm in mobile computing involves integrating these algorithm into software or hardware of mobile device. Here is a high level overview of how can approach implementing these algorithm in programs of mobile computing.

Steps to implement A3/A5/A8 GSM security Algorithm in Mobile computing.

① Understand Algorithm specification:-

- Study specification & details of A3, A5, A8 algorithm to understand their operations & requirement

② Integrate Algorithms

- Implement A3 algorithm for generating SRES using subscriber secret key and random number (RAND)

Implement A8 algorithm to derive cipher key by encryption random number (RAND) with secret key (K_i)

Integrate A8 algorithm for encryption voice and data traffic using cipher key generated by A8.

③ Utilize cryptographic Libraries

- Utilize cryptographic libraries available for mobile platform to perform encryption & decryption operation required by algorithm

④ Secure key management

- ensure secure storage and management of sensitive key like K_i , K_c & other parameters within mobile to prevent unauthorized access.

⑤ Testing & Validation

- Thoroughly test implementation of these algorithm in simulated environment to validate their correctness & effectiveness in securing communication between the mobile service & the network

6.) Compliance & certification:-

- Ensure compliance with relevant standards and regulation governing mobile security such as GSM security standard to meet industry requirement & certification.

By following these steps and utilizing cryptographic libraries tailored for mobile platform, you can implement the A3/A5/A8 GSM security algorithm in a program for mobile computing, enhancing security of communication between mobile device & cellular networks.

Conclusion :- learnt about A3, A5, A8 GSM security in mobile computing.

~~Final~~ (P.T)

Code:

Exp10A3Algo.java :

```
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;
import java.util.Random;

public class Exp10A3Algo {

    public static void main(String[] args) {
        // Simulate generating a random Ki (secret key) and RAND (challenge)
        String ki = generateRandomHexString(32); // 128 bits (16 bytes) key
        String rand = generateRandomHexString(32); // 128 bits (16 bytes)
        challenge

        // Display the generated Ki and RAND
        System.out.println("Ki (Secret Key): " + ki);
        System.out.println("RAND (Challenge): " + rand);

        // Calculate the expected response (SRES) using the A3 algorithm
        String sres = calculateSRES(ki, rand);

        // Display the calculated SRES
    }
}
```

```
        System.out.println("SRES (Expected Response): " + sres);
    }

private static String generateRandomHexString(int length) {
    Random random = new Random();
    StringBuilder randomHex = new StringBuilder();

    for (int i = 0; i < length; i++) {
        int randomInt = random.nextInt(16); // 0-15
        randomHex.append(Integer.toHexString(randomInt));
    }

    return randomHex.toString();
}

private static String calculateSRES(String ki, String rand) {
    try {
        // Concatenate Ki and RAND
        String input = ki + rand;

        // Use SHA-1 hash function to calculate SRES
        MessageDigest sha1 = MessageDigest.getInstance("SHA-1");
        byte[] hashBytes = sha1.digest(hexStringToByteArray(input));

        // Convert the hash to a hexadecimal string
    }
}
```

```
StringBuilder sres = new StringBuilder();
for (byte b : hashBytes) {
    sres.append(String.format("%02X", b));
}

return sres.toString();
} catch (NoSuchAlgorithmException e) {
    e.printStackTrace();
    return null;
}

private static byte[] hexStringToByteArray(String hexString) {
    int len = hexString.length();
    byte[] data = new byte[len / 2];

    for (int i = 0; i < len; i += 2) {
        data[i / 2] = (byte) ((Character.digit(hexString.charAt(i), 16) << 4)
            + Character.digit(hexString.charAt(i + 1), 16));
    }

    return data;
}
```

Output:

```
(base) PS C:\Users\Shirish\OneDrive\Desktop\MCC Exp Documents>
cd "c:\Users\Rishab\OneDrive\Desktop\MCC Exp Documents\" ; if
(?) { javac Exp10A3Algo.java } ; if (?) { java Exp10A3Algo }
```

Ki (Secret Key): 75b3cd449c491cb7af27683de9dba3f8

RAND (Challenge): 5ed4d5250e87177fd4c4f4e9f9238cf4

SRES (Expected Response):

8A69046163903D54366D9AF1E410B40E56872AC6

Exp10A5Algo.java :

```
import java.lang.Math;

public class A5 {
    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;
}

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

public static void main(String[] args) {
    int[] a;
    System.out.println("Generating the 1st key identification number");
}
```

```
a = GenerateBits();
for (int i = 0; i < 16; i++) {
    System.out.print(a[i]);
}
int[] b;
System.out.println("\n\nGenerating the 2nd 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 random number");
c = GenerateBits();
for (int i = 0; i < 16; i++) {
    System.out.print(c[i]);
}
int[] d;
System.out.println("\n\nGenerating the barker code");
d = GenerateBits();
for (int i = 0; i < 16; i++) {
    System.out.print(d[i]);
}
int[] z;
System.out.println("\n\nAND of 1st key and 2nd key");
z = AND(a, b);
int[] p;
System.out.println("\n\nXOR of random number and the AND of 1st key and 2nd key");
p = XOR(z, c);
System.out.println("\n\nXOR of the above number and barker code");
p = XOR(p, d);
```

```
int[] q;  
System.out.println("\n\nXOR of random number and the AND of 1st key and 2nd key");  
q = XOR(z, c);  
System.out.println("\n\nXOR of the above number and barker code");  
q = XOR(q, d);  
int flag = 0;  
for (int i = 0; i < 16; i++) {  
    if (p[i] != q[i]) {  
        flag = 1;  
        break;  
    }  
}  
if (flag == 1) {  
    System.out.print("\n\nEncryption Failed");  
} else {  
    System.out.print("\n\nEncryption Passed");  
}  
  
}  
}
```

Output:

```
(base) PS C:\Users\Rishab\OneDrive\Desktop\MCC Exp Documents>  
cd "c:\Users\Rishab\OneDrive\Desktop\MCC Exp Documents\" ; if  
($?) { javac Exp10A5Algo.java } ; if ($?) { java Exp10A5Algo }
```

Generating the 1st key identification number

0111001001011110

Generating the 2nd key identification number

1111110111000001

Generating the random number

1011000011011100

Generating the barker code

0001011100001111

AND of 1st key and 2nd key

0111000001000000

XOR of random number and the AND of 1st key and 2nd key

1100000010011100

XOR of the above number and barker code

1101011110010011

XOR of random number and the AND of 1st key and 2nd key

1100000010011100

XOR of the above number and barker code

1101011110010011

Encryption Passed

Keystream bit = $0 \wedge 0 \wedge 0 = 0$

Keystream bit = $0 \wedge 0 \wedge 0 = 0$

Keystream bit = $1 \wedge 1 \wedge 0 = 0$

Keystream bit = $0 \wedge 1 \wedge 1 = 0$

Keystream bit = $1 \wedge 1 \wedge 1 = 1$

Keystream bit = $0 \wedge 0 \wedge 1 = 1$

Exp10A8Algo.java :

```
public class A8Algorithm {  
    // Secret key (Ki)  
    private static final int[] KI = { 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1 };  
  
    // A8 Algorithm  
    public static int[] generateKeyStream(int[] rand) {  
        int[] keyStream = new int[rand.length];
```

```
// Generate key stream

for (int i = 0; i < rand.length; i++) {

    keyStream[i] = rand[i] ^ KI[i % KI.length];
}

return keyStream;
}

// Example usage

public static void main(String[] args) {

    // Example random number (RAND)

    int[] rand = { 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0 };

    // Generate key stream using A8 algorithm

    int[] keyStream = generateKeyStream(rand);

    // Print key stream

    System.out.println("Key Stream:");

    for (int keyBit : keyStream) {

        System.out.print(keyBit);
    }
}
```

Output:

```
(base) PS C:\Users\Shirish\OneDrive\Desktop\MCC Exp Documents>
cd "c:\Users\Shirish\OneDrive\Desktop\MCC Exp Documents\" ; if
(?) { javac Exp10A8Algo.java } ; if (?) { java Exp10A8Algo }
```

Step 1: RAND = 1, KI = 0, Key Bit = RAND ^ KI = 1 ^ 0 = 1

Step 2: RAND = 0, KI = 1, Key Bit = RAND \wedge KI = 0 \wedge 1 = 1

Step 3: RAND = 1, KI = 0, Key Bit = RAND \wedge KI = 1 \wedge 0 = 1

Step 4: RAND = 0, KI = 1, Key Bit = RAND \wedge KI = 0 \wedge 1 = 1

Step 5: RAND = 1, KI = 0, Key Bit = RAND \wedge KI = 1 \wedge 0 = 1

Step 6: RAND = 0, KI = 1, Key Bit = RAND \wedge KI = 0 \wedge 1 = 1

Step 7: RAND = 1, KI = 0, Key Bit = RAND \wedge KI = 1 \wedge 0 = 1

Step 8: RAND = 0, KI = 1, Key Bit = RAND \wedge KI = 0 \wedge 1 = 1

Step 9: RAND = 1, KI = 0, Key Bit = RAND \wedge KI = 1 \wedge 0 = 1

Step 10: RAND = 0, KI = 1, Key Bit = RAND \wedge KI = 0 \wedge 1 = 1

Step 11: RAND = 1, KI = 0, Key Bit = RAND \wedge KI = 1 \wedge 0 = 1

Step 12: RAND = 0, KI = 1, Key Bit = RAND \wedge KI = 0 \wedge 1 = 1

Step 13: RAND = 1, KI = 0, Key Bit = RAND \wedge KI = 1 \wedge 0 = 1

Step 14: RAND = 0, KI = 1, Key Bit = RAND \wedge KI = 0 \wedge 1 = 1

Step 15: RAND = 1, KI = 0, Key Bit = RAND \wedge KI = 1 \wedge 0 = 1

Step 16: RAND = 0, KI = 1, Key Bit = RAND \wedge KI = 0 \wedge 1 = 1

Key Stream:

1111111111111111

MCC Assignment-1

Q1) Explain merits and demerits and with details :-

a.) Snooping TCP

- It is a technique in computer networking which involves manipulating or intercepting TCP traffic.
- The attacker impersonates a trusted entity and exploits TCP/IP stack vulnerabilities for unauthorized access.
- Potential malicious activities include eavesdropping, session hijacking & man in the middle attack.
- The process includes intercepting TCP packets, manipulating their contents, and potentially hijacking established sessions.

Advantages:-

- 1.) Since end to end TCP semantic is preserved, the crashing timing of foreign agent is not altered.
- 2.) When mobile host moves to another foreign agent, there is no need of handover state.

Disadvantages:-

- 1.) negative acknowledgement between the foreign agent and mobile host needs additional mechanism on mobile host to maintain transparency.
- 2.) If encryption is applied, snooping & buffering data may be useless.

b.) Mobile TCP

- Mobile TCP adapts the traditional TCP protocols to challenges in mobile networks like 3G, 4G, and 5G.
- Challenges include higher latency, variable bandwidth and frequent handovers affecting TCP performance.
- Adaptations like TCP Westwood and TCP NewReno have been proposed to address these mobile network challenges.

Advantages :-

- 1.) It maintains the TCP end-to-end semantics.
- 2.) Lost packets are automatically retransmitted to the new SH.

Disadvantages:-

- 1.) Packet loss on wireless link due to errors is propagated to the sender because SH does not act as proxy.
- 2.) M-TCP always assumes low bit error rates, which are not always valid assumption.

c.) I-TCP (Indirect TCP)

- splits the TCP connection into forward & reverse connection for better control over communication flow.
- Adaptive congestion window mechanism and

and base station and an optimized protocol specific to wireless links between base station & wireless host

Mobility :-

- ① Optimised wireless communication :- I-TCP optimises communication over wireless link by using specialized protocol tailored to wireless characteristics
- ② Centralized Proxy for Optimization : By using a centralized proxy, I-TCP can efficiently manage communication between fixed host & mobile devices via wireless links.

Demerits :-

- ① Loss of End to End semantics :- I-TCP sacrifices end to end semantics by splitting the connection and introducing a proxy element, potentially affecting certain aspect of communication integrity.
- ② Explain cellular (mobile) IP in details

Cellular IP, also known as mobile IP, is a communication protocol that enables mobile device users to maintain the same permanent IP address while moving between different network. This protocol was created by extending different network

(1) Mobile Node (MN) :- It is hand held communication device that user carries e.g all phone

(2) Home Network :- It network to which mobile node originally belong as per its assigned IP address.

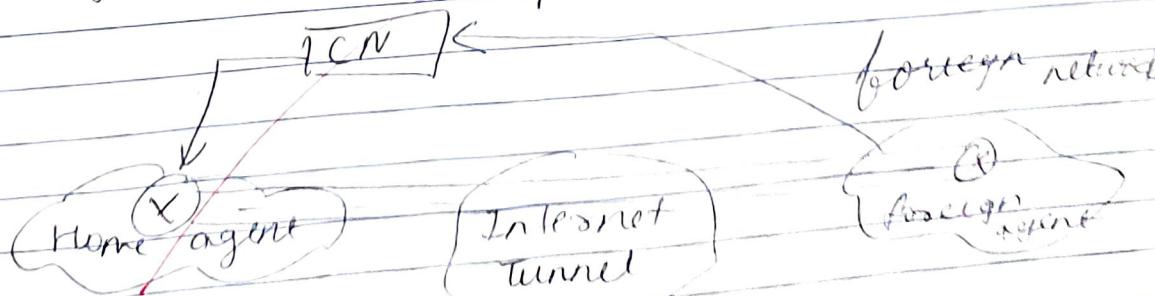
(3) Home Agent (HA) :- Is a router in home network to which the mobile nodes was originally connected.

Home address :- is permanent address assigned to mobile node

Foreign network :- is current network to which mobile node is visiting

Foreign agent :- A router on foreign network that form and packet from Home Agent to mobile node

care of Address :- is temporary address used by a mobile node while it is moving away from its home network corresponding node.



F5

3) Explain MIPv6 :-

→ Mobile IPv6 (MIPv6) is a protocol designed to support mobile connection within the internet protocol version 6 framework.

→ mobile IPv6 allows a mobile node to transparently maintain connection while moving from one subnet to another. Each device is identified by home address through which it may be connecting to through another network. When connecting through a foreign network, a mobile device sends its location information to a home agent, which intercepts packets intended for device and tunnels them to current location.

Q4) write short note on : HAWAII

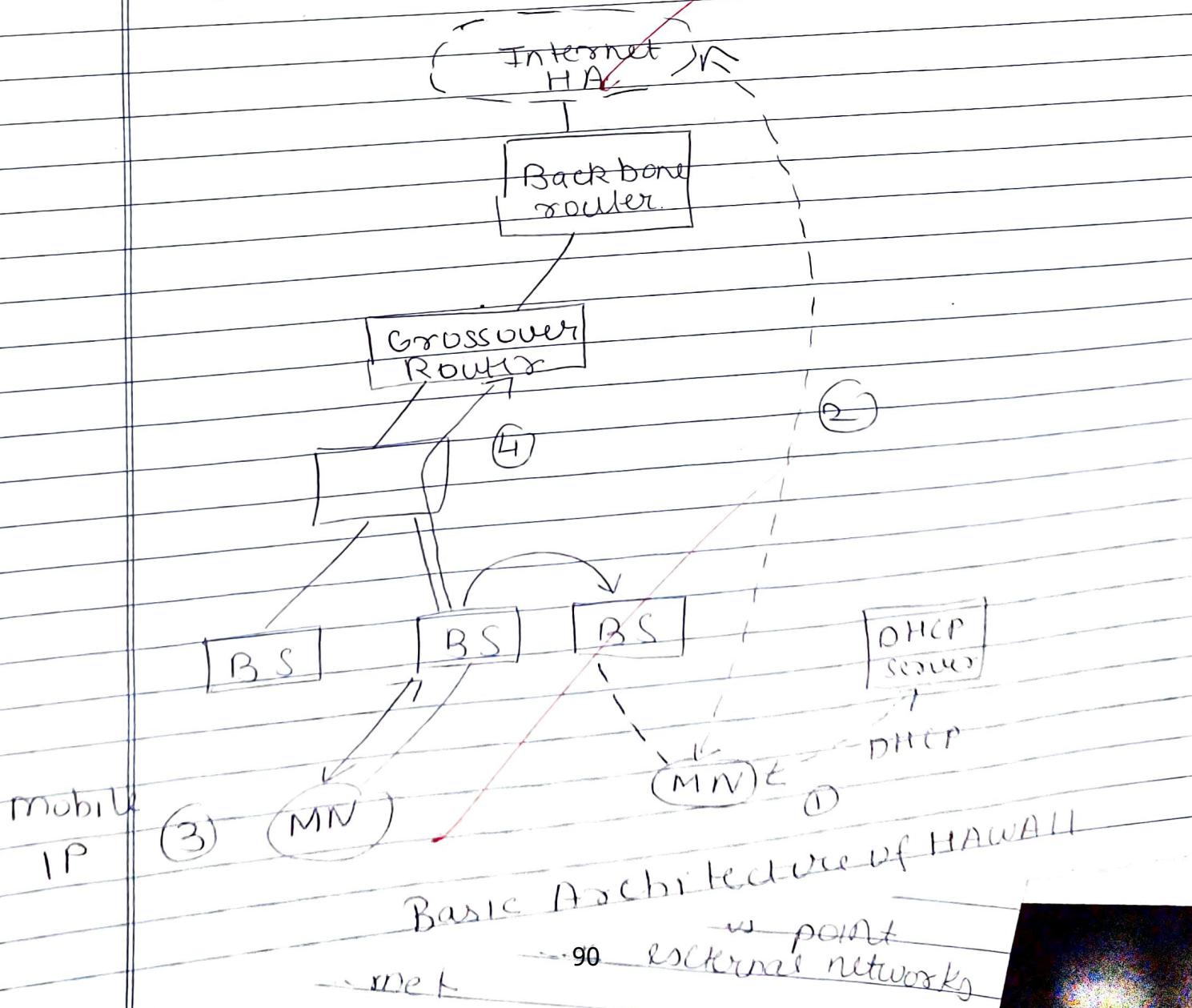
→ HAWAII tries to keep mobile mobility support as transparent as possible between home agent and mobile node.

→ On entering a HAWAII domain, a mobile node obtains an co-located COA and registers with the HA (step 2)

→ Additionally, when moving to another cell inside the foreign domain, the MN sends a registration request to new base station as to foreign agent (step 3)

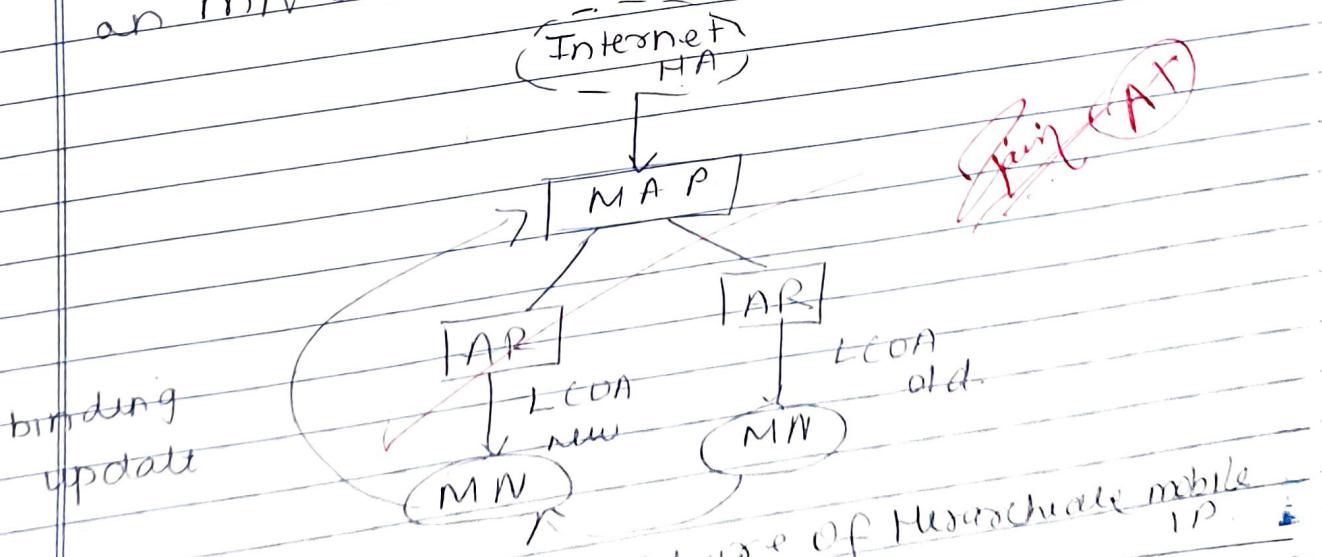
such as the internal network between VPL (89) and external network. It is

- The base station intercepts the registration request to the new base station, which reconfigure all routers on path from the old & new base station to so-called crossover router (step 4).
- When routing has been reconfigured successfully the base station sends a registration reply to the mobile node again as if it were a foreign agent.



Q5) Explain HMIPv6 in detail

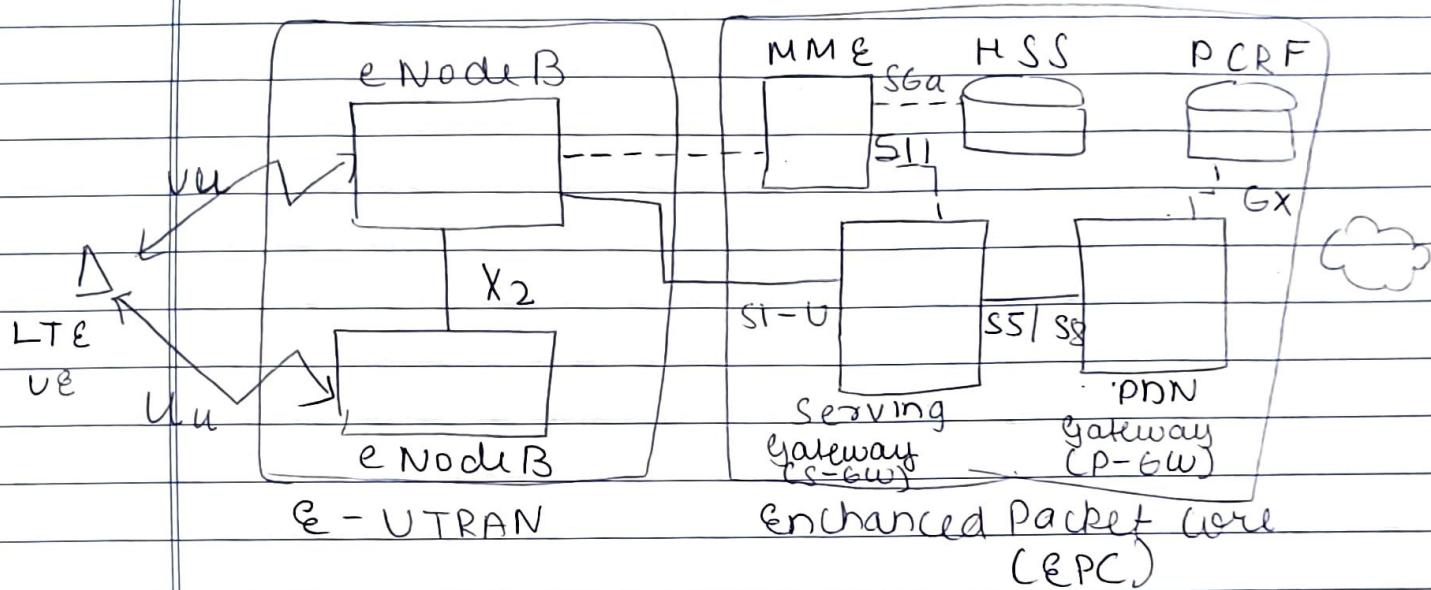
- HMIPv6 provides micro-mobility support by installing a mobility anchor point (MAP) which is responsible for certain domain & acts as a local HA within its domain for visiting MNs.
- As long as an MN stays the domain of a MAP, globally visible COA doesn't change.
- A MAP domain boundaries are defined by the AR advertising the MAP information to attached MN.
- MN register their RCOA with HA using a binding update.
- To support smooth handover between MAP domains an MN sends a binding update to its former MAP.



Basic architecture of Hierarchical mobile IP

MCC Assignment No: 2

Q1) Draw and explain SA&E architecture.



The Main components of SA&E architecture are:-

- (1) **User Equipment (UE):** mobile device used to connect to network.
- (2) **Evolved NodeB (eNB):** Base station that make up the E-UTRAN, the radio access network for LTE.
- (3) **Evolved Packet core (EPC):** core network of SA&E architecture consisting of following elements-
 - a.) **Mobil Management Entity (MME):**- This component is responsible for management of connection between UE and network.
 - b.) **Serving gateway (S-GW):**- Responsible for routing & forwarding user data between UE & PDN gateway.
 - c.) **PDN gateway (P-GW):**- It is point of connection between UE & external networks such as the internet.

Q2.) Describe evolution from UMTS to LTE.

- The evolution of UMTS to LTE represent a significant advancement in mobile technology. The key stages are :-

1) UMTS (3G) :-

- UMTS was upgradable to 4G network which provides faster data transfer rates and services like video calling
- UMTS utilized WCDMA as its air interface technology.

2) HSPA (High-Speed Packet Access)

- HSPA was an intermediate step to enhance data speed within UMTS framework
- HSPA improved data rates and both uplink & downlink transmission offering a smoother mobile experience

3) LTE (4G) :-

- LTE introduced as part of 4G, marked a significant leap in data speed & network efficiency
- LTE uses different air interfaces called OFDMA for downlink and SC-FDMA for uplink, providing improved spectral efficiency.

3) compare mobile generations (1G, 2G, 3G, 4G, 5G)

Features	1 G	2 G	3 G	4 G(LTE)	5 G
① Timeframe	1980-1990s	1990-2000s	2000s-2010s	2010- Present	2020- Present
② Data Rates	Up to 2 Kbps	Up to 64 Kbps	Up to 2Mbps	Up to 1 Gbps	Up to 20Gbps
③ Latency	High	High	Moderate	Low	Ultra Low
④ Network architecture	Circuit Switched	Circuit Switched	Partially packet switched	All IP	Divergent application
⑤ Application	Voice calls only	Voice calls (SMS, basic data)	Mobile internet, video calling	Streaming speed data, gaming	Diverse application (IoT, AR/VR)

Q4) What are self organizing networks?

- Self organizing network are initially automated various task traditionally handled by network operators, making them more automated.
- The core functionalities SON are :-
 - a.) Self configuration :- SON's can automatically configure the new base station added to network.
 - b.) Self optimization - SON's continuously monitor network performance metrics and based on real time data, they adjust parameters to optimize the performance.
 - c.) Self healing :- SON's can detect and automatically take corrective action such as rerouting.

Q5) Explain VOLTE in detail :-

- VOLTE stands for Voice over long term evolution which is technology that makes voice calls to be carried over LTE.

- Key aspects of VOLTE are
 - 1.) IP Based voice :- VOLTE uses packet-switched tech over LTE unlike 2G/3G circuits.
 - 2.) Enhanced Voice Quality :- offers HD voice with wider audio range for clearer sound.
 - 3.) Simultaneous Voice[Data] :- Allow talk, data concurrently.

- 4.) Efficient Spectrum use :- optimize network resource for better performance.

- 5.) RCS integration :- Includes advanced features like enhanced messaging and video calls.