Program 3: Develop a native calculator application

Step 1: Project Setup

1. Launch Android Studio and create a new Android project with a suitable name and package.
2. Choose an appropriate form factor and minimum API level for your project.

Step 2: XML Layout

1. Open the `activity_main.xml` layout file.
2. Define the UI elements: two EditText views for input numbers, Buttons for each operation (add, subtract, multiply, divide, and square root), and a TextView to display the result.

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
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:padding="16dp"
    tools:context=".MainActivity">

    <EditText
        android:id="@+id/num1EditText"
```

```
        android:layout_width="0dp"
        android:layout_height="48dp"
        android:layout_marginTop="44dp"
        android:hint="Enter number 1"
        android:inputType="numberDecimal"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <EditText
        android:id="@+id/num2EditText"
        android:layout_width="0dp"
        android:layout_height="48dp"
        android:layout_marginTop="12dp"
        android:hint="Enter number 2"
        android:inputType="numberDecimal"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.47"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/num1EditText" />

    <Button
        android:id="@+id/addButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="20dp"
        android:text="+"
        android:textSize="16sp"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/num2EditText" />

    <Button
        android:id="@+id/subtractButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="20dp"
        android:text="-"
        android:textSize="16sp"
        app:layout_constraintEnd_toStartOf="@+id/multiplyButton"
        app:layout_constraintStart_toEndOf="@+id/addButton"
        app:layout_constraintTop_toBottomOf="@+id/num2EditText" />

    <Button
        android:id="@+id/multiplyButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="20dp"
        android:text="x"
        android:textSize="16sp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/num2EditText" />
```

```
    <Button
        android:id="@+id/divideButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="20dp"
        android:text="/"
        android:textSize="16sp"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/addButton" />

    <Button
        android:id="@+id/sqrtButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="20dp"
        android:layout_marginEnd="140dp"
        android:text="Sqrt"
        android:textSize="16sp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/subtractButton" />

    <TextView
        android:id="@+id/resultTextView"
        android:layout_width="84dp"
        android:layout_height="41dp"
        android:layout_marginStart="4dp"
        android:layout_marginTop="40dp"
        android:text="Result: "
        android:textSize="18sp"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/divideButton" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

Step 3: Java Code (MainActivity.java)

1. Create a new Java class named `MainActivity` that extends `AppCompatActivity`.
2. Declare private instance variables for the EditText views, TextView, and Buttons.

3. Inside the `onCreate` method, use `findViewById` to initialize the UI elements by their respective IDs.
4. Set `OnClickListener` for each operation Button to perform calculations when clicked.
5. Define a `performCalculation` method that takes an operator as an argument and performs the calculation based on the operator (+, -, *, /).
6. Inside the `performCalculation` method, parse the input numbers from the EditText views, perform the calculation, and display the result in the TextView.
7. Create a `calculateSquareRoot` method to calculate the square root of a number.
8. In both calculation methods, format the result using `DecimalFormat` to show a maximum of two decimal places.

```
package com.example.simplecalculator;

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

import androidx.appcompat.app.AppCompatActivity;

import com.example.simplecalculator.R;

import java.text.DecimalFormat;

public class MainActivity extends AppCompatActivity {

    // Declare variables to hold references to UI elements
```

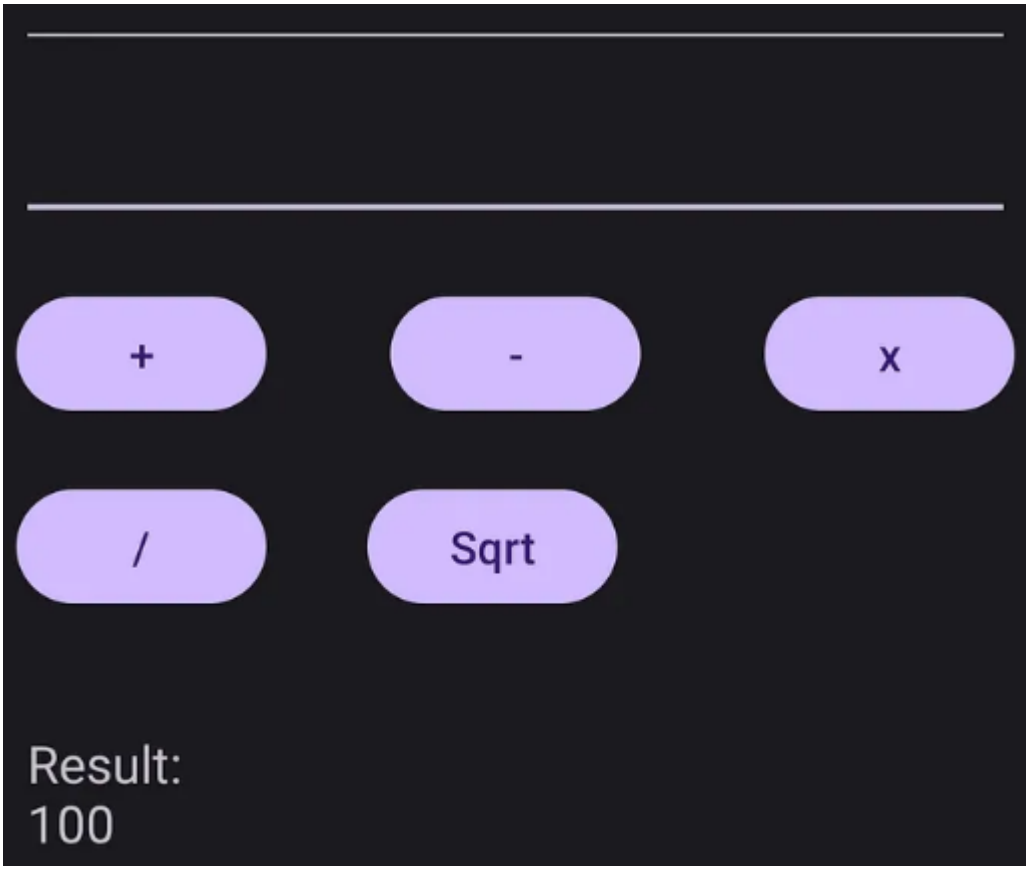
```
    private EditText num1EditText, num2EditText;
    private TextView resultTextView;

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

        // Initialize UI elements from the layout
        num1EditText = findViewById(R.id.num1EditText);
        num2EditText = findViewById(R.id.num2EditText);
        resultTextView = findViewById(R.id.resultTextView);

        // Set click listeners for arithmetic operation buttons

        Button addButton = findViewById(R.id.addButton);
        addButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                performCalculation('+');
            }
        });

        Button subtractButton = findViewById(R.id.subtractButton);
        subtractButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                performCalculation('-');
            }
        });

        Button multiplyButton = findViewById(R.id.multiplyButton);
        multiplyButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                performCalculation('*');
            }
        });

        Button divideButton = findViewById(R.id.divideButton);
        divideButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                performCalculation('/');
            }
        });

        Button sqrtButton = findViewById(R.id.sqrtButton);
        sqrtButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                calculateSquareRoot();
```

```
            }
        });
    }

    private void performCalculation(char operator) {
        // Get the values entered in the input fields
        String num1Str = num1EditText.getText().toString();
        String num2Str = num2EditText.getText().toString();

        // Check if either input field is empty
        if (num1Str.isEmpty() || num2Str.isEmpty()) {
            Toast.makeText(this, "Please enter both numbers",
Toast.LENGTH_SHORT).show();
            return; // Exit the method to prevent calculations with empty
inputs
        }

        // Convert the input values to numeric format
        double num1 = Double.parseDouble(num1Str);
        double num2 = Double.parseDouble(num2Str);
        double result = 0;

        // Perform the selected calculation based on the operator
        switch (operator) {
            case '+':
                result = num1 + num2;
                break;
            case '-':
                result = num1 - num2;
                break;
            case '*':
                result = num1 * num2;
                break;
            case '/':
                if (num2 != 0) {
                    result = num1 / num2;
                } else {
                    Toast.makeText(this, "Cannot divide by zero",
Toast.LENGTH_SHORT).show();
                    return; // Exit the method if division by zero is
attempted
                }
                break;
        }

        // Format and display the calculation result
        DecimalFormat df = new DecimalFormat("#.##");
        resultTextView.setText("Result: " + df.format(result));
    }

    private void calculateSquareRoot() {
        String num1Str = num1EditText.getText().toString();
```

```
        // Check if the input field is empty
        if (num1Str.isEmpty()) {
            Toast.makeText(this, "Please enter a number",
Toast.LENGTH_SHORT).show();
            return; // Exit the method to prevent calculations with empty
inputs
        }

        double num = Double.parseDouble(num1Str);
        double sqrtResult = Math.sqrt(num);
        DecimalFormat df = new DecimalFormat("#.##");
        resultTextView.setText("Square Root: " + df.format(sqrtResult));
    }
}
```

The `calculateSquareRoot` method takes the input number from the first EditText view, calculates its square root using the `Math.sqrt` function, and displays the result with two decimal places in the TextView.

Output


+
-
x
/
Sqrt
Result:
100