EXP:09	Develop an Android application using controls like Button, TextView, EditText for designing a calculator having basic functionality like Addition, Subtraction, multiplication, and Division.

Aim:

To develop an Android application that functions as a basic calculator, allowing users to perform arithmetic operations (Addition, Subtraction, Multiplication, and Division) using interactive UI components like **Button**, **TextView**, and **EditText**.

Algorithm:

1. Start the app.
2. Display two input fields for numbers.
3. Show buttons: □ □ □ □ and (clear).
4. User enters two numbers and taps a button.
5. App checks if both inputs are valid numbers:
   - If not: show a toast message "Enter valid numbers".
   - If valid:
     - Perform the selected operation.
     - Show the result on the screen.
6. Clear button resets everything.
7. End.

Code:

MainActivity.kt:

```
package com.example.mycalculator
import android.os.Bundle
import android.view.View
import android.widget.TextView
import androidx.appcompat.app.AppCompatActivity

class MainActivity : AppCompatActivity() {
    private lateinit var inputText: TextView
    private lateinit var resultText: TextView
    private var input: String = ""
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        inputText = findViewById(R.id.textInput)
        resultText = findViewById(R.id.textResult)
    }
    fun onNumberClick(view: android.view.View) {
        val button = view as android.widget.Button
        input += button.text
        inputText.text = input
    }
    fun onAllClear(view: View) {
```

```
    input = ""
    inputText.text = ""
    resultText.text = ""
}
fun onClearOne(view: View) {
    if (input.isNotEmpty()) {
        input = input.dropLast(1)
        inputText.text = input
    }
}
fun onEquals(view: android.view.View) {
    try {
        val expression = input.replace("×", "*").replace("÷", "/")
        val result = eval(expression)
        resultText.text = "= $result"
    } catch (e: Exception) {
        resultText.text = "Error"
    }
}
private fun eval(expr: String): Double {
    return object : Any() {
        var pos = -1
        var ch = 0

        fun nextChar() {
            ch = if (++pos < expr.length) expr[pos].code else -1
        }

        fun eat(charToEat: Int): Boolean {
            while (ch == ' '.code) nextChar()
            if (ch == charToEat) {
                nextChar()
                return true
            }
            return false
        }

        fun parse(): Double {
            nextChar()
            val x = parseExpression()
            if (pos < expr.length) throw RuntimeException("Unexpected: ${expr[pos]}")
            return x
        }

        fun parseExpression(): Double {
            var x = parseTerm()
            while (true) {
                when {
                    eat('+'.code) -> x += parseTerm()
                    eat('-'.code) -> x -= parseTerm()
```

```
                else -> return x
            }
        }
    }

    fun parseTerm(): Double {
        var x = parseFactor()
        while (true) {
            when {
                eat('*'.code) -> x *= parseFactor()
                eat('/'.code) -> x /= parseFactor()
                else -> return x
            }
        }
    }

    fun parseFactor(): Double {
        if (eat('+'.code)) return parseFactor()
        if (eat('-'.code)) return -parseFactor()

        var x: Double
        val startPos = pos
        if (eat('('.code)) {
            x = parseExpression()
            eat(')'.code)
        } else if (ch in '0'.code..'9'.code || ch == '.'.code) {
            while (ch in '0'.code..'9'.code || ch == '.'.code) nextChar()
            x = expr.substring(startPos, pos).toDouble()
        } else {
            throw RuntimeException("Unexpected: ${ch.toChar()}")
        }

        return x
    }
}.parse()
}
}
```

ActivityMain.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"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#000000">

    <!-- Display Input -->
    <TextView
        android:id="@+id/textInput"
```

```
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:textSize="40sp"
        android:textColor="#FFFFFF"
        android:gravity="end"
        android:padding="16dp"
        android:text=""
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintBottom_toTopOf="@+id/buttonGrid"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent" />

    <!-- Display Result -->
    <TextView
        android:id="@+id/textResult"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:textSize="28sp"
        android:textColor="#AAAAAA"
        android:gravity="end"
        android:paddingEnd="16dp"
        android:text=""
        app:layout_constraintTop_toBottomOf="@id/textInput"
        app:layout_constraintBottom_toTopOf="@+id/buttonGrid"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent" />

    <!-- Button Grid -->
    <GridLayout
        android:id="@+id/buttonGrid"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:columnCount="4"
        android:padding="8dp"
        android:layout_marginBottom="16dp"
        android:useDefaultMargins="true"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent">

        <!-- Row 1 -->
        <Button android:text="AC" android:onClick="onAllClear" style="@style/CalcButton"
/>
        <Button android:text="C" android:onClick="onClearOne" style="@style/CalcButton"
/>
        <Button android:text="(" android:onClick="onNumberClick"
style="@style/CalcButton" />
        <Button android:text="÷" android:onClick="onNumberClick"
style="@style/CalcButton" />
```

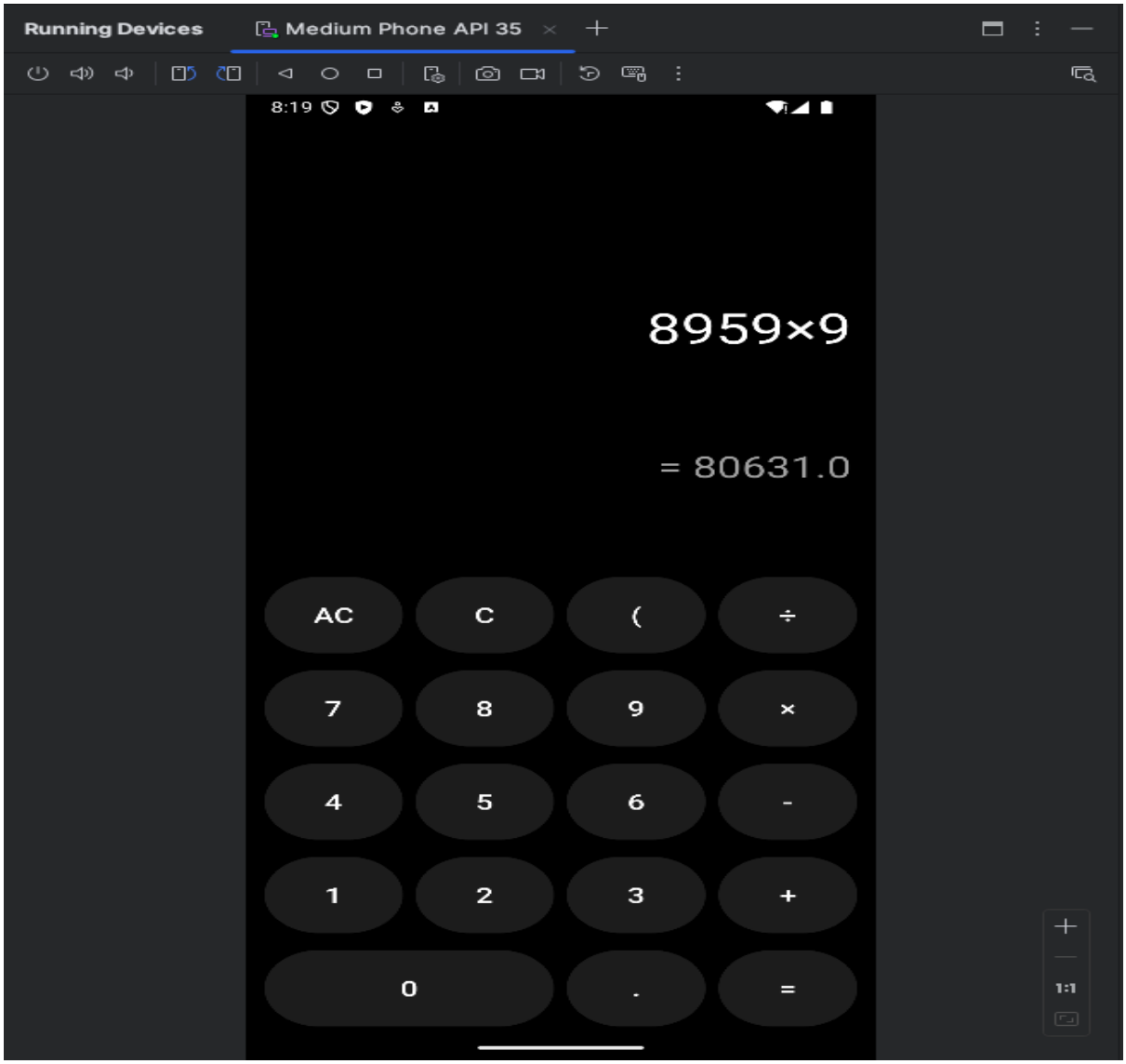
```
        <!-- Row 2 -->
        <Button android:text="7" android:onClick="onNumberClick"
style="@style/CalcButton" />
        <Button android:text="8" android:onClick="onNumberClick"
style="@style/CalcButton" />
        <Button android:text="9" android:onClick="onNumberClick"
style="@style/CalcButton" />
        <Button android:text="×" android:onClick="onNumberClick"
style="@style/CalcButton" />

        <!-- Row 3 -->
        <Button android:text="4" android:onClick="onNumberClick"
style="@style/CalcButton" />
        <Button android:text="5" android:onClick="onNumberClick"
style="@style/CalcButton" />
        <Button android:text="6" android:onClick="onNumberClick"
style="@style/CalcButton" />
        <Button android:text="-" android:onClick="onNumberClick"
style="@style/CalcButton" />

        <!-- Row 4 -->
        <Button android:text="1" android:onClick="onNumberClick"
style="@style/CalcButton" />
        <Button android:text="2" android:onClick="onNumberClick"
style="@style/CalcButton" />
        <Button android:text="3" android:onClick="onNumberClick"
style="@style/CalcButton" />
        <Button android:text="+" android:onClick="onNumberClick"
style="@style/CalcButton" />

        <!-- Row 5 -->
        <Button
            android:text="0"
            android:onClick="onNumberClick"
            style="@style/CalcButton"
            android:layout_columnSpan="2" />

        <Button android:text="." android:onClick="onNumberClick"
style="@style/CalcButton" />
        <Button android:text="=" android:onClick="onEquals" style="@style/CalcButton" />
    </GridLayout>

</androidx.constraintlayout.widget.ConstraintLayout>
```

Styles.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <style name="CalcButton">
        <item name="android:layout_width">0dp</item>
        <item name="android:layout_height">80dp</item>
```

```
        <item name="android:layout_columnWeight">1</item>
        <item name="android:layout_margin">4dp</item>
        <item name="android:textSize">20sp</item>
        <item name="android:textColor">#FFFFFF</item>
        <item name="android:backgroundTint">#1E1E1E</item>
    </style>
</resources>
```

Output:



Result:

Thus the give program is executed successfully.