**1. 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:background="#1B1919"

tools:context=".MainActivity">

<TextView

android:id="@+id/textView1"

android:layout\_width="406dp"

android:layout\_height="54dp"

android:paddingLeft="35dp"

android:paddingTop="10dp"

android:text="Calculator Example"

android:textSize="24sp"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintHorizontal\_bias="0.001"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent"

app:layout\_constraintVertical\_bias="0.003" />

<androidx.constraintlayout.widget.ConstraintLayout

android:id="@+id/frameLayout"

android:layout\_width="401dp"

android:layout\_height="616dp"

android:layout\_marginBottom="60dp"

android:background="#AEABAB"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/textView1"

app:layout\_constraintVertical\_bias="1.0">

<EditText

android:id="@+id/editText1"

android:layout\_width="366dp"

android:layout\_height="100dp"

android:ems="10"

android:inputType="text"

android:textSize="24sp"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintHorizontal\_bias="0.542"

app:layout\_constraintStart\_toStartOf="parent"

tools:layout\_editor\_absoluteY="26dp" />

<LinearLayout

style="@style/LayoutStyle"

app:layout\_constraintVertical\_bias="0.884">

<Button

android:id="@+id/btn17"

android:layout\_width="324dp"

android:layout\_height="wrap\_content"

android:layout\_marginStart="16dp"

android:layout\_marginTop="8dp"

android:text="="

android:textSize="24sp"

android:textStyle="bold"

app:cornerRadius="0dp"

app:strokeColor="#872525"

app:strokeWidth="5dp" />

</LinearLayout>

<LinearLayout

style="@style/LayoutStyle"

app:layout\_constraintVertical\_bias="0.377">

<Button

android:id="@+id/btn1"

style="@style/BtnStyle"

android:layout\_marginStart="16dp"

android:text="1"/>

<Button

android:id="@+id/btn2"

style="@style/BtnStyle"

android:layout\_marginStart="5dp"

android:text="2"/>

<Button

android:id="@+id/btn3"

style="@style/BtnStyle"

android:layout\_marginStart="5dp"

android:text="3" />

<Button

android:id="@+id/btn13"

style="@style/BtnStyle"

android:layout\_marginStart="25dp"

android:text="+" />

</LinearLayout>

<LinearLayout

style="@style/LayoutStyle"

app:layout\_constraintVertical\_bias="0.508">

<Button

android:id="@+id/btn4"

style="@style/BtnStyle"

android:text="4"

android:layout\_marginStart="16dp" />

<Button

android:id="@+id/btn5"

style="@style/BtnStyle"

android:layout\_marginStart="5dp"

android:text="5"/>

<Button

android:id="@+id/btn6"

style="@style/BtnStyle"

android:layout\_marginStart="5dp"

android:text="6" />

<Button

android:id="@+id/btn14"

style="@style/BtnStyle"

android:layout\_marginStart="25dp"

android:text="-" />

</LinearLayout>

<LinearLayout

style="@style/LayoutStyle"

app:layout\_constraintVertical\_bias="0.638">

<Button

android:id="@+id/btn7"

style="@style/BtnStyle"

android:layout\_marginStart="16dp"

android:text="7"/>

<Button

android:id="@+id/btn8"

style="@style/BtnStyle"

android:layout\_marginStart="5dp"

android:text="8" />

<Button

android:id="@+id/btn9"

style="@style/BtnStyle"

android:layout\_marginStart="5dp"

android:text="9" />

<Button

android:id="@+id/btn15"

style="@style/BtnStyle"

android:layout\_marginStart="25dp"

android:text="\*"/>

</LinearLayout>

<LinearLayout

style="@style/LayoutStyle"

app:layout\_constraintVertical\_bias="0.768">

<Button

android:id="@+id/btn10"

style="@style/BtnStyle"

android:layout\_marginStart="16dp"

android:text="." />

<Button

android:id="@+id/btn11"

style="@style/BtnStyle"

android:layout\_marginStart="5dp"

android:text="0" />

<Button

android:id="@+id/btn12"

style="@style/BtnStyle"

android:layout\_marginStart="5dp"

android:text="C" />

<Button

android:id="@+id/btn16"

style="@style/BtnStyle"

android:layout\_marginStart="25dp"

android:text="/" />

</LinearLayout>

</androidx.constraintlayout.widget.ConstraintLayout>

</androidx.constraintlayout.widget.ConstraintLayout>

2. MainActivity.kt

package com.example.calculator

import android.os.Bundle

import android.widget.Button

import android.widget.EditText

import androidx.appcompat.app.AppCompatActivity

import java.util.Stack

class MainActivity : AppCompatActivity() {

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

setContentView(R.layout.activity\_main)

val editText1 = findViewById<EditText>(R.id.editText1)

val btns = listOf<Button>(

findViewById(R.id.btn1),

findViewById(R.id.btn2),

findViewById(R.id.btn3),

findViewById(R.id.btn4),

findViewById(R.id.btn5),

findViewById(R.id.btn6),

findViewById(R.id.btn7),

findViewById(R.id.btn8),

findViewById(R.id.btn9),

findViewById(R.id.btn10),

findViewById(R.id.btn11),

findViewById(R.id.btn12),

findViewById(R.id.btn13),

findViewById(R.id.btn14),

findViewById(R.id.btn15),

findViewById(R.id.btn16),

findViewById(R.id.btn17)

)

for (btn in btns) {

btn.setOnClickListener {

val btnTxt = btn.getText().toString()

var dataCalc = " "; // = editText1.getText().toString() + btnTxt

// editText1.setText(dataCalc)

if (btnTxt == "C") {

clearEntry(editText1)

} else {

dataCalc = editText1.getText().toString() + btnTxt

editText1.setText(dataCalc)

}

if (btnTxt == "=") {

clearEntry(editText1)

dataCalc = editText1.getText().toString()

var result = evaluateExpression(dataCalc)

editText1.setText(result.toString())

}

}

}

}

}

fun evaluateExpression(expression: String): Double {

val tokens = tokenize(expression)

val postfix = infixToPostfix(tokens)

return evaluatePostfix(postfix)

}

fun tokenize(expression: String): List<String> {

// Split the expression into tokens (numbers, operators, etc.)

return expression.split(" ")

}

fun infixToPostfix(infix: List<String>): List<String> {

val output = mutableListOf<String>()

val stack = mutableListOf<String>()

for (token in infix) {

when {

token.isNumeric() -> output.add(token)

token == "(" -> stack.add(token)

token == ")" -> {

while (stack.isNotEmpty() && stack.last() != "(") {

output.add(stack.removeAt(stack.size - 1))

}

stack.removeAt(stack.size - 1) // Remove '('

}

else -> {

while (stack.isNotEmpty() && precedence(token) <= precedence(stack.last())) {

output.add(stack.removeAt(stack.size - 1))

}

stack.add(token)

}

}

}

while (stack.isNotEmpty()) {

output.add(stack.removeAt(stack.size - 1))

}

return output

}

fun precedence(operator: String): Int {

return when (operator) {

"+", "-" -> 1

"\*", "/" -> 2

else -> 0

}

}

fun precedence(operator: Char): Int {

return when (operator) {

'+', '-' -> 1

'\*', '/' -> 2

else -> 0

}

}

fun evaluatePostfix(postfix: List<String>): Double {

// Evaluate the postfix expression using a stack-based algorithm

// You need to implement a stack-based algorithm to evaluate the postfix expression

// For simplicity, let's assume the expression contains only two operands and one operator

val stack = Stack<Double>()

for (token in postfix) {

if (token.isNumeric()) {

stack.push(token.toDouble())

} else {

val operand2 = stack.pop()

val operand1 = stack.pop()

when (token) {

"+" -> stack.push(operand1 + operand2)

"-" -> stack.push(operand1 - operand2)

"\*" -> stack.push(operand1 \* operand2)

"/" -> stack.push(operand1 / operand2)

else -> throw IllegalArgumentException("Invalid operator")

}

}

}

return stack.pop()

}

fun String.isNumeric(): Boolean {

return try {

this.toDouble()

true

} catch (e: NumberFormatException) {

false

}

}

private fun Double.isWholeNumber(): Boolean {

return this % 1 == 0.0

}

fun clearEntry(edtTxt: EditText) {

var str = edtTxt.getText().toString()

str = str.substring(0, str.length - 1)

edtTxt.setText(str);

}