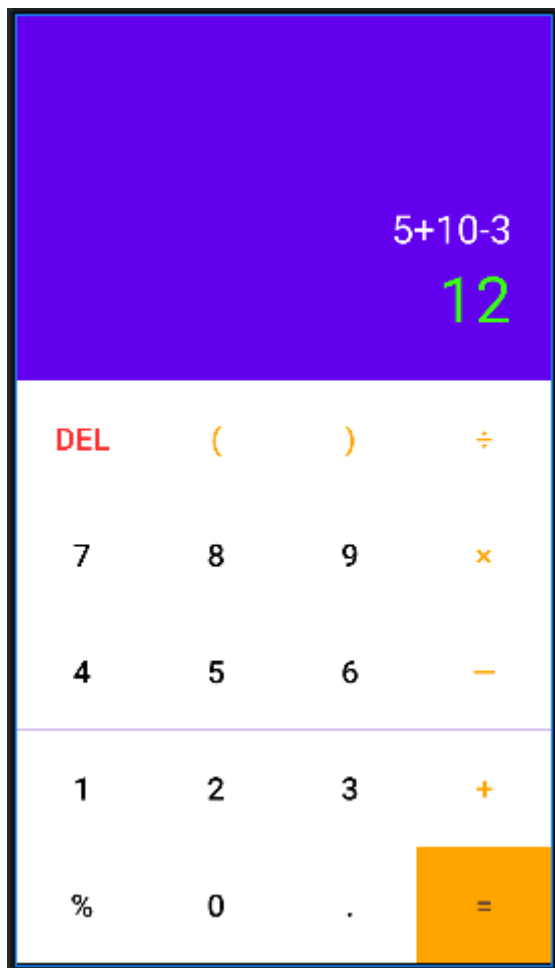


```
1 <?xml version="1.0" encoding="utf-8"?>
2 <resources>
3     <color name="black">#FF000000</color>
4     <color name="white">#FFFFFFF</color>
5     <color name="purple_200">#FFBB86FC</color>
6     <color name="purple_500">#FF6200EE</color>
7     <color name="purple_700">#FF3700B3</color>
8     <color name="teal_200">#FF03DAC5</color>
9     <color name="teal_700">#FF018786</color>
10    <color name="red">#FF3131</color>
11    <color name="pink">#FFC0CB</color>
12    <color name="orange">#FFA500</color>
13    <color name="neon_green">#39FF14</color>
14
15
16    <style name="Button_Style" parent="Widget.AppCompat.Button.Colored">
17        <item name="android:background">@color/white</item>
18        <item name="android:textSize">24sp</item>
19        <item name="android:textColor">@color/black</item>
20        <item name="android:gravity">center</item>
21    </style>
22 </resources>
23
```



```
package com.example.myapplication

import androidx.activity.enableEdgeToEdge
import androidx.core.view.ViewCompat
import androidx.core.view.WindowInsetsCompat
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import androidx.core.content.ContextCompat
import com.example.myapplication.databinding.ActivityMainBinding
import net.objecthunter.exp4j.ExpressionBuilder
import java.text.DecimalFormat
```

```

class MainActivity : AppCompatActivity() {
    lateinit var binding: ActivityMainBinding
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        binding = ActivityMainBinding.inflate(layoutInflater)
        setContentView(binding.root)
        binding.buttonClear.setOnClickListener {
            binding.input.text = ""
            binding.output.text = ""
        }
        binding.buttonBracketLeft.setOnClickListener {
            addToInputText( value: ")")
        }
        binding.buttonBracketLeft.setOnClickListener {
            addToInputText( value: "(")
        }
        binding.button0.setOnClickListener {
            addToInputText( value: "0")
        }
        binding.button1.setOnClickListener {
            addToInputText( value: "1")
        }
        binding.button2.setOnClickListener {
            addToInputText( value: "2")
        }
        binding.button3.setOnClickListener {
            addToInputText( value: "3")
        }
    }
}

```

```

private fun addToInputText(value: String) {
    binding.input.append(value)// Добавляем переданное значение в конец поля
}

// Функция для получения строки ввода
private fun getInputExpression(): String {
    return binding.input.text.toString()
}

```

```
private fun showResult() {
    try {
        val expression = getInputExpression().replace( oldValue: "%", newValue: "/100") //

        val result = ExpressionBuilder(expression).build().evaluate()
        binding.output.text =
            DecimalFormat( pattern: "0.#####").format(result).toString()
        binding.output.setTextColor(ContextCompat.getColor( context: this,
            R.color.neon_green))
    } catch (e: Exception) {
        binding.output.text = "Ошибка"
        binding.output.setTextColor(
            ContextCompat.getColor(
                context: this,
                R.color.red
            )
        )
    }
}
```

