Міністерство освіти і науки України
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра обчислювальної техніки

Лабораторна робота №3.1

з дисципліни

«Інтелектуальні вбудовані системи»

на тему

«РЕАЛІЗАЦІЯ ЗАДАЧІ РОЗКЛАДАННЯ ЧИСЛА НА ПРОСТІ МНОЖНИКИ (ФАКТОРИЗАЦІЯ ЧИСЛА)»

Виконав:
студент групи ІП-83
Карпюк Ігор Вікторович
номер залікової книжки: 8311

Перевірив:
викладач
Регіда Павло Геннадійович

Київ 2021

Основні теоретичні відомості

Факторизації лежить в основі стійкості деяких криптоалгоритмів, еліптичних кривих, алгебраїчній теорії чисел та кванових обчислень, саме тому дана задача дуже гостро досліджується, й шукаються шляхи її оптимізації.

На вхід задачі подається число n Є ℕ, яке необхідно факторизувати. Перед виконанням алгоритму слід переконатись в тому, що число не просте. Далі алгоритм шукає перший простий дільник, після чого можна запустити алгоритм заново, для повторної факторизації.

В залежності від складності алгоритми факторизації можна розбити на дві групи:

- Експоненціальні алгоритми (складність залежить експоненційно від довжини вхідного параметру);
- Субекспоненціальні алгоритми.

Існування алгоритму з поліноміальною складністю – одна з найважливіших проблем в сучасній теорії чисел. Проте, факторизація з даною складністю можлива на квантовому комп'ютері за допомогою алгоритма Шора.

Рис1. Алгоритми факторизації

Завдання

Розробити програма для факторизації заданого числа методом Ферма. Реалізувати користувацький інтерфейс з можливістю вводу даних.

Варіант

Варіант: 11

Число гармонік в сигналі n: 10

Гранична частота, ω гр: 1500

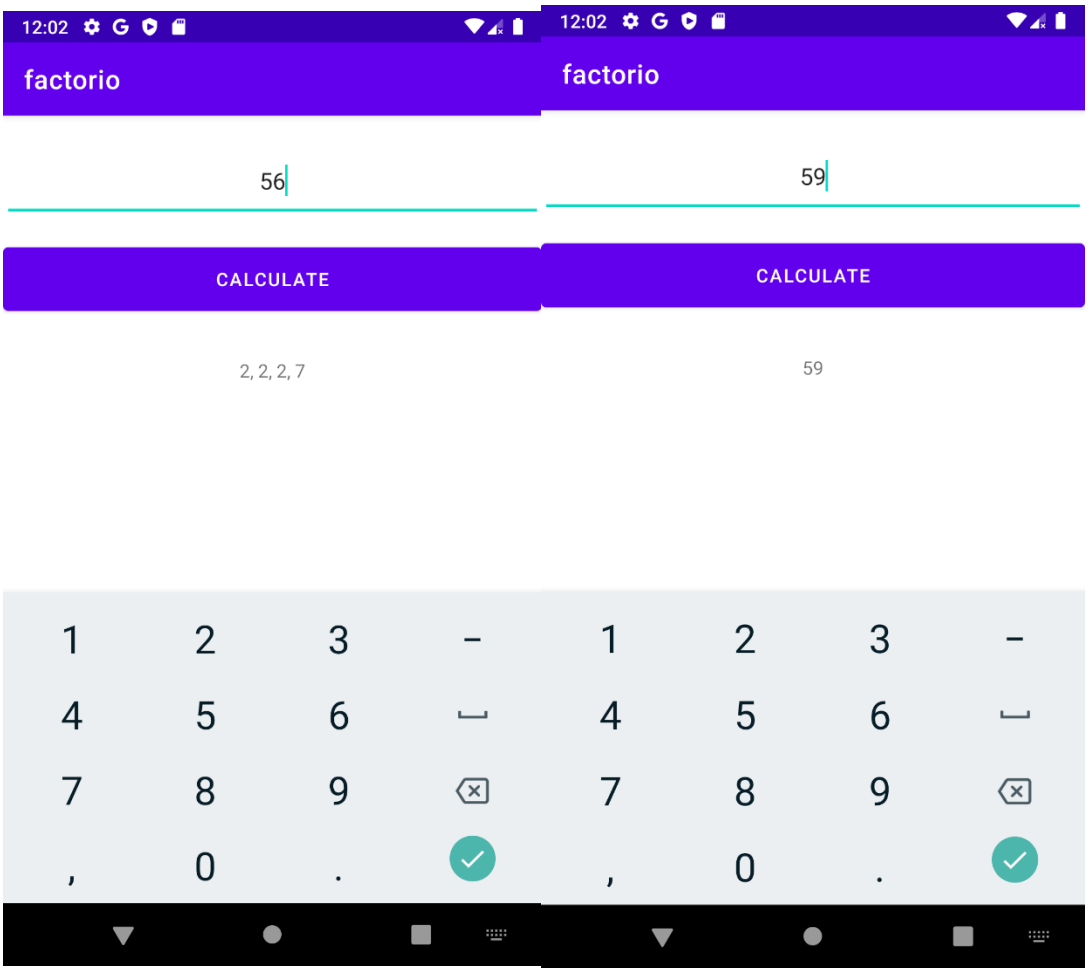
Кількість дискретних відліків, N: 256

Лістинг програми MainActivity.kt

```
package ua.kpi.comsys.factorio

import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.widget.Button
import android.widget.TextView
import android.widget.Toast
import com.google.android.material.textfield.TextInputEditText

class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        val input = findViewById<TextInputEditText>(R.id.inputField)
        val output = findViewById<TextView>(R.id.outputText)
        val calcBut = findViewById<Button>(R.id.calcBut)

        calcBut.setOnClickListener { v ->
            val original = input.text.toString().toInt()
            val factorsArray = factorize(original)

            if (factorsArray == null) {
                Toast.makeText(applicationContext,"wrong
number",Toast.LENGTH_SHORT).show()
            } else {
                output.text = factorsArray.joinToString()
            }
        }
```

```
    }
}

fun factorize(original: Int): MutableList<Int>? {

    // Return if the input number is up to 3
    if (original == 0 || original == 1 || original == 2 || original == 3) return
mutableListOf(original)
    if (original < 0) return null

    // Initiate the prime factors array
    val factors = mutableListOf<Int>()
    var changer: Int = original

    // Iterate through to half of the original number and find all prime factors
    var i: Int = 2
    while (i <= original / 2) {

        if (changer % i == 0) {
            factors.add(i)
            changer /= i
            if (changer == 1) return factors
            continue
        } else {
            i++
        }
    }

    // If found factors - return them. Otherwise - original is a prime number
    return if (factors.isEmpty()) mutableListOf(original) else factors
}
```

Результати роботи програми



Висновки