Міністерство освіти і науки України

Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського"

Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

Звіт

з лабораторної роботи № 6 з дисципліни «Алгоритми та структури даних-1. Основи алгоритмізації»

«Дослідження рекурсивних алгоритмів»

Варіант 16

Виконав студент ІП-11 Кузьменков Дмитро Олегович

(шифр, прізвище, ім'я, по батькові)

Перевірив Мартинова. О. П

( прізвище, ім'я, по батькові)

Київ 2021

**Лабораторна робота 6**

**Дослідження рекурсивних алгоритмів**

**Мета** - дослідити особливості роботи рекурсивних алгоритмів та набути практичних навичок їх використання під час складання програмних специфікацій підпрограм.

# Завдання:

Дано натуральне число n. Отримати всі прості дільники цього числа.

**Постановка задачі:**

Почнемо з побудови арифметичного циклу, що буде знаходити усі числа i, на які число n ділиться націло. Кожне таке число пропускаємо через підпрограму func(), що перевірятиме число на простоту за допомогою арифметичного циклу. Якщо число i поділиться націло на будь-яке число від 2 до і-1, то число не зможе вважатися простим, тому функція поверне false, інакше – повертається true. Якщо підпрограма повертає true, то виводимо **і** як простий дільник n.

# Побудова математичної моделі. Складемо таблицю імен змінних:

|  |  |  |  |
| --- | --- | --- | --- |
| Змінна | Тип | Ім’я | Призначення |
| Задане число | Натуральне | n | Початкове дане |
| Дільник числа n | Натуральне | i | Початкове дане, проміжне дане, результат |
| Число, на яке будемо ділити i, щоб перевіряти його на простоту | Натуральне | j | Початкове дане,  проміжне дане |
| Функція перевірки числа на простоту | Функція | Func() |  |
| Змінна всередині функції | Натуральне | x | Проміжне дане |

Вводимо n

Крок 1. Деталізуємо цикл, що знаходить дільники n.

Крок 2. Деталізуємо функцію перевірки числа на простоту.

Виводимо i

1. **Псевдокод:**

# Початок

Вводимо n

1) Деталізуємо цикл, що знаходить дільники n

2) Деталізуємо функцію перевірки числа на простоту

Виводимо i

**Кінець**

*Крок1*

# Початок

Вводимо n

1) **Поки** i:= 2; i <= n; i=i+1

**Повторити**

**Якщо** n % i == 0

**То**

**Якщо** func(i) = true

**То** виводимо i

2) Деталізуємо функцію перевірки числа на простоту

**Кінець**

*Крок2*

# Початок

Вводимо n

1) **Поки** i:= 2; i <= n; i=i+1

**Повторити**

**Якщо** n % i == 0

**То**

**Якщо** func(i) = true

**То** виводимо i

2) **func(x):=**

**Поки** j:= 2; j < x; j=j+1

**Якщо** fib1 % j == 0

**то** return false

**інакше**

**Якщо** fib1 == j + 1

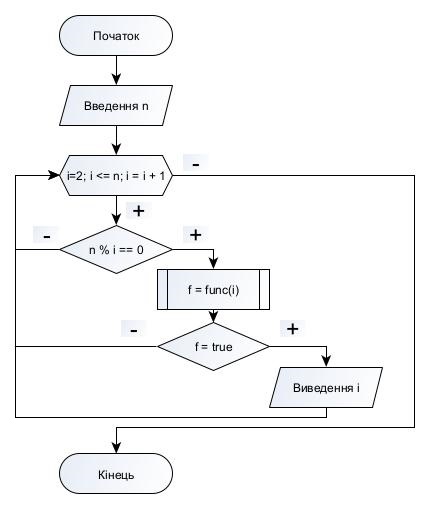
**то** return true

**Кінець**

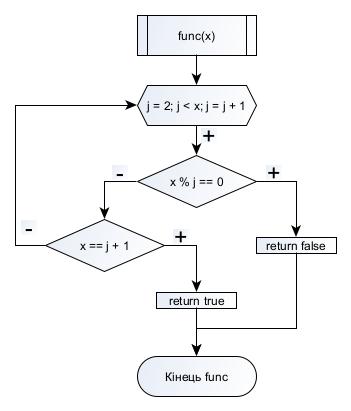
1. **Блок-схема**



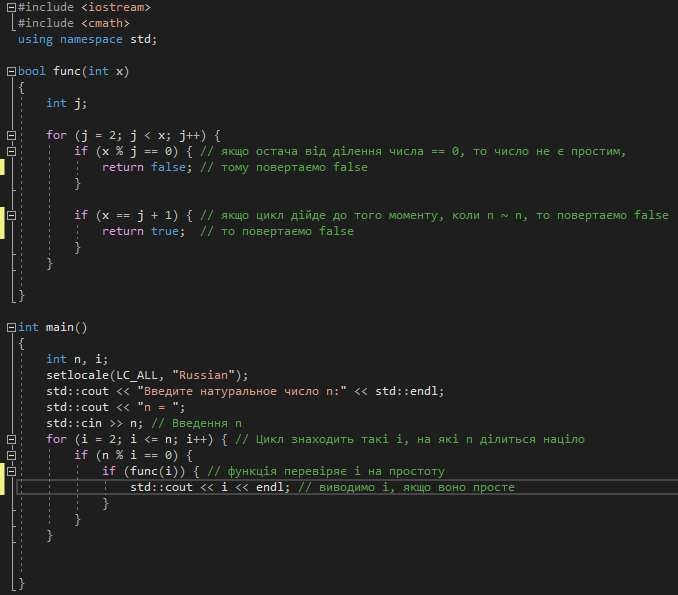
*Крок1*



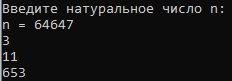
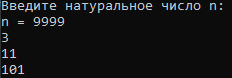
*Крок2*

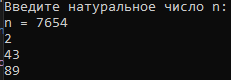


**4. Код (C++):**

****

**Результати:**

** **

****

**Висновок:** Виконуючи лабораторну роботу я дослідив особливості роботи рекурсивних алгоритмів та набув практичних навичок їх використання під час складання програмних специфікацій підпрограм. Побудував алгоритм та написав програму знаходження усіх простих дільників заданого числа n шляхом використання рекурсивного алгоритму та арифметичних циклів.