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 № 5 з дисципліни

«Алгоритми та структури даних-1.

Основи алгоритмізації»

«Дослідження лінійних алгоритмів»

Варіант 25

Виконав студент ІП-14 Радзівіло Валерія Артемівна

(шифр, прізвище, ім'я, по батькові)

Перевірив Мартинова Оксана Петрівна

( прізвище, ім'я, по батькові)

Київ 2021

**Лабораторна робота 5**

**Дослідження складних циклічних алгоритмів**

**Мета:** дослідити особливості роботи складних циклів та набути практичних навичок їх використання під час складання програмних специфікацій.

**Задача**:



Користувач задає а. Знайдемо найближчі прості до а числа та порівняємо їх відстані до а. Це число може бути більшим або меншим. Вони можуть бути і рівними, тоді ми ставимо, що 2 числа рівновіддалені від а.

**Математична модель**

|  |  |  |  |
| --- | --- | --- | --- |
| **Змінна** | **Тип** | **Ім’я** | **Призначення** |
| Функція checkPrime | boolean | checkPrime | Функція |
| Ціле число a | Цілий | a | Частина функції |
| Дійсне число a | Дійсний | а | Початкове данне |
| Ціле число amount | Цілий | amount | Проміжкове дане |
| Цілочисельний масив primes | Масив | primes | Проміжкове дане |
| Цілочисельний масив dist | Масив | primes | Проміжкове дане |
| Ціле число prime | Цілий | prime | Проміжкове дане |
| Ціле число i | Цілий | i | Проміжкове дане |
| Ціле число з масиву primes | Цілий | primes[i] | Результат |

Програмні специфікації запишемо у псевдокоді та графічній формі у вигляді блок-схеми.

Крок 1. Визначимо основні дії.

Крок 2. Знаходимо найближче просте число.

**Псевдокод**

*крок* 1

**початок**

введення а

обчислення найближчого простого числа

виведення найближчого простого числа

**кінець**

*крок* 2

**початок**

**функція** checkPrime

**якщо** a<=1

**то**

повернути 0

**якщо** a==2 || a==3

**то**

повернути 1

**якщо** a % 2 == 0 || a % 3==0

**то**

повернути 0

**інакше**

**повторити**

i=5

**якщо** a%i == 0 || a%(i+2) == 0

**то**

повернути 0

i+=6

**поки** i\*i<=6

повернути 1

введення а

**якщо** a<=1

**то**

**вивести** closest: 2

**якщо** checkPrime(a) == 1

**то**

**вивести** a is a prime number

**інакше**

i =0

**повторити**

prime = amount+amount/100

**якщо** checkPrime(prime) == 1

**то**

primes[i] = prime

dist[i] = abs(prime - a)

**якщо** i>1

**то**

**якщо** dist[i-1]==dist[i]

**то**

**вивести** primes[i-1] and primes [i] are equally distanced from a

**інакше**

**якщо** dist[i-1] < dist[i]

**то**

**вивести** closest: primes[i-1]

**якщо** dist[i-2]<dist[i-1]

**то**

**вивести** closest: primes[i-2]

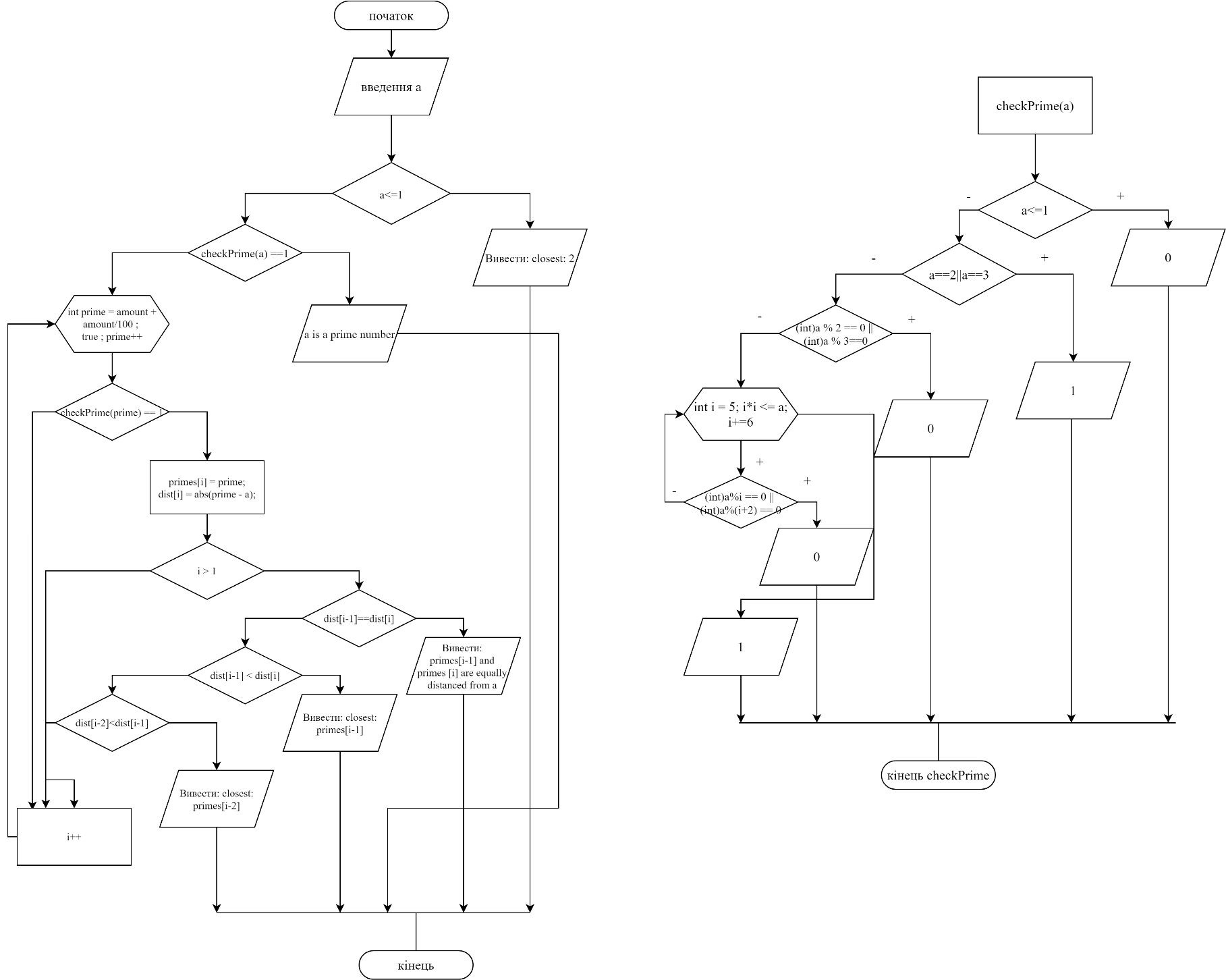
i++

prime ++

**поки** true

**Крок 1**

**Крок 2**

****

**Випробовування алгоритму:**

|  |  |
| --- | --- |
| 1. | Введемо дані: n = 4 |
| 2. | 4>1 |
| 3. | checkPrime(4) = 1 – не виконується |
| 4. | Ітерації:   1. primes[1]=2 2. primes[2]=3 3. primes[3]=5 |
| 5. | primes[2]= primes[3] |
| 6. | 3 and 5 are equally distanced from 4 |

**Висновок**: Було досліджено особливості роботи складних циклів та набуто практичних навичок їх використання під час складання програмних специфікацій. У алгоритмі були уточнені обмеження з використанням умов при яких задача буде виконана правильно, а також функція, що перевіряє чи є число простим. За допомоги складних циклів визначені найближчі прості числа до заданого з обмеженнями. За допомоги цієї лабораторної роботи можливо обчислення найближчих або найближчого простого числа до будь-якого заданого користувачем.