

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 № 4 з дисципліни «Алгоритми та
структури даних-1. Основи алгоритмізації»

«Дослідження лінійних алгоритмів»

Варіант 25

Виконав студент

Павленко Микита Андрійович

(шифр, прізвище, ім'я, по батькові)

Перевірив

Вечерковська Анастасія Сергіївна

(прізвище, ім'я, по батькові)

Лабораторна робота Дослідження складних циклічних алгоритмів

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

Варіант 25

Завдання

Дано число **a**. Знайти найближче до нього просте число.

1) Постановка задачі

За допомогою складного циклу визначити найближче до **a** просте число (може бути як лівіше **a** на числовій осі, так і правіше).

2) Побудова математичної моделі

Таблиця імен змінних:

Змінна	Тип	Ім'я	Призначення
Вхідне число	Цілий	a	Вхідні дані
Число, що передує a	Цілий	a_minus	Проміжні дані
Число, що йде після a	Цілий	a_plus	Проміжні дані
Перевірка того, чи просте число	Логічний	ne_proste1, ne_proste2	Проміжні дані
Лічильник ітерацій	Цілий	i, j	Проміжні дані
Результат	Цілий	result	Вихідні дані

Отже, математичне формулювання задачі зводиться до перевірки чисел «з двох боків» від числа **a** на умову їх простоти та подальше виведення.

3) Псевдокод алгоритму

Крок 1:

початок

Введення **a**

Ініціалізація та оголошення інших змінних

Оголошення умови зовнішнього циклу

Знаходження елемента, що задовольняє умові

Перевірка і присвоєння потрібного результату

Виведення результату

кінець

Крок 2:

початок

Введення **a**

a_minus = a

a_plus = a

ne_proste1 = true

ne_proste2 = true

Оголошення умови зовнішнього циклу

Знаходження елемента, що задовольняє умові

Перевірка і присвоєння потрібного результату

Виведення результату

кінець

Крок 3:

початок

Введення **a**

a_minus = a

a_plus = a

ne_proste1 = true

ne_proste2 = true

Поки (**ne_proste1 && ne_proste2**) повторити

Знаходження елемента, що задовольняє умові

Перевірка і присвоєння потрібного результату

Виведення результату

кінець

Крок 4:

початок

Введення **a**

a_minus = a

a_plus = a

ne_proste1 = true

ne_proste2 = true

Поки (**ne_proste1 && ne_proste2**) повторити

a_minus --

a_plus ++

i = 2

j = 2

ne_proste1 = false

ne_proste2 = false

Поки (**(i < a_minus) && !(ne_proste1)**) повторити

Якщо **a_minus % i == 0** то

ne_proste1 = true

інакше

ne_proste1 = false

все якщо

i ++

все повторити

Поки **((j < a_plus) && !(ne_proste2))** повторити

Якщо **a_plus % j == 0** то

ne_proste2 = true

інакше

ne_proste2 = false

все якщо

j ++

все повторити

все повторити

Перевірка і присвоєння потрібного результату

Виведення результату

кінець

Крок 5:

початок

Введення **a**

a_minus = a

a_plus = a

ne_proste1 = true

ne_proste2 = true

Поки **(ne_proste1 && ne_proste2)** повторити

a_minus --

a_plus ++

i = 2

j = 2

ne_proste1 = false

ne_proste2 = false

Поки **((i < a_minus) && (ne_proste1))** повторити

Якщо **a_minus % i == 0** то

ne_proste1 = true

інакше

ne_proste1 = false

все якщо

i ++

все повторити

Поки **((j < a_plus) && !(ne_proste2))** повторити

Якщо **a_plus % j == 0** то

ne_proste2 = true

інакше

ne_proste2 = false

все якщо

j ++

все повторити

все повторити

Якщо **ne_proste1 = false** то

result = i

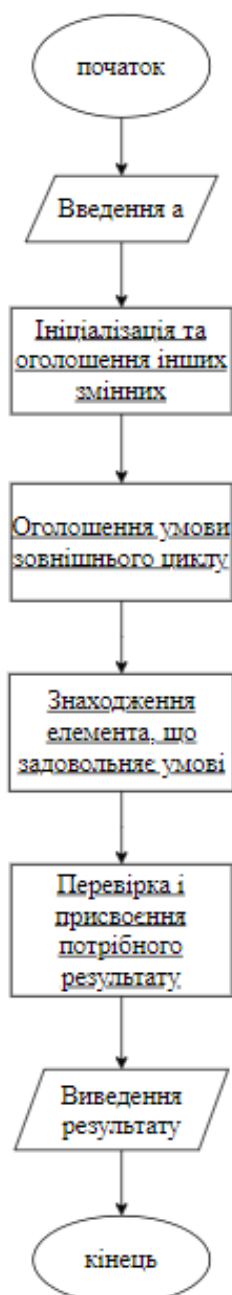
інакше

result = j

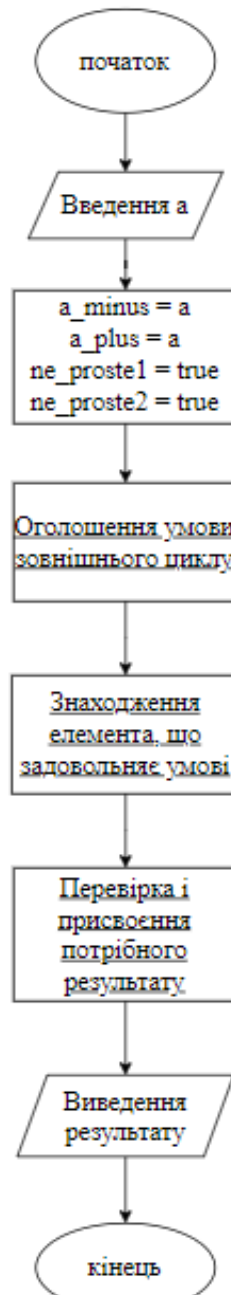
Виведення результату
кінець

4) Блок-схема алгоритму

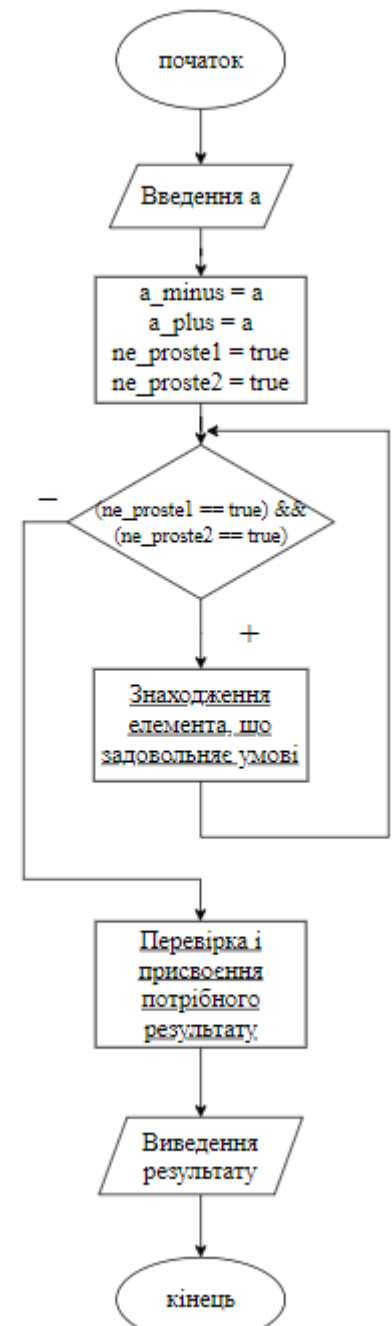
Крок 1



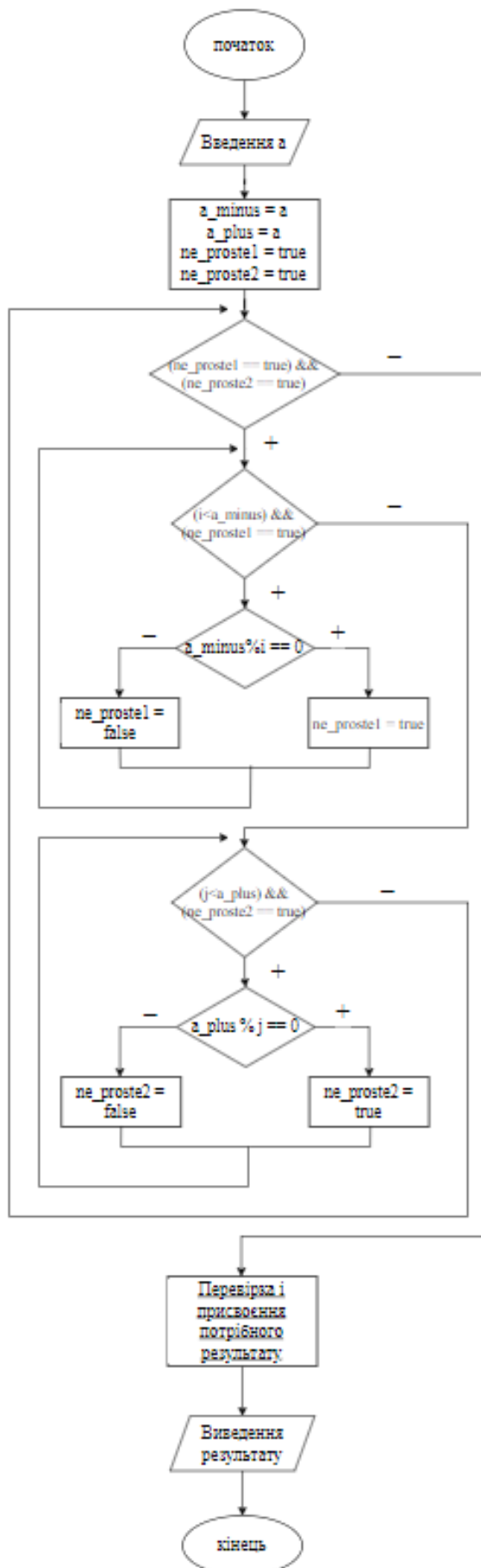
Крок 2



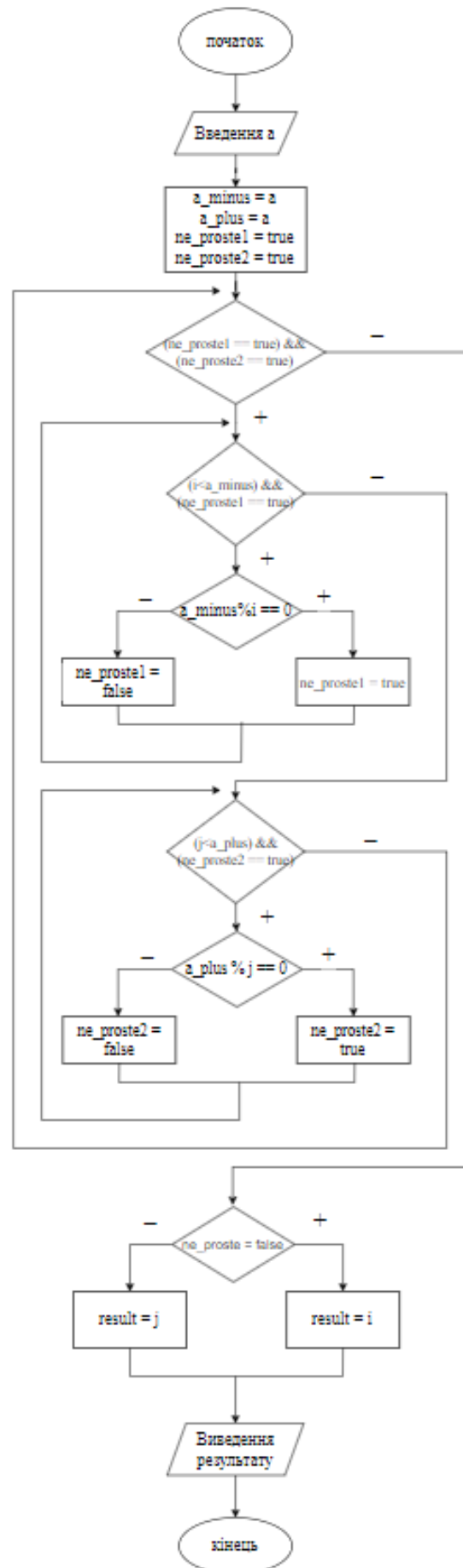
Крок 3



Крок 4



Крок 5



5) Випробування алгоритму

Блок	Дія
	Початок
1	Введення: a := 25
2	a_minus = 25 a_plus = 25 ne_proste1 = true ne_proste2 = true
3	(ne_proste1 && ne_proste2) == true
4	a_minus -= 24 a_plus += 26 i = 2 j = 2 ne_proste1 = false ne_proste2 = false
5	((i < a_minus) && !(ne_proste1)) == true
6	(a_minus % i == 0) == true
7	ne_proste = true i = 3
...	...
9	a_minus -= 23 a_plus += 27 i = 2 j = 2 ne_proste1 = false ne_proste2 = false
10	((i < a_minus) && !(ne_proste1)) == true
11	(a_minus % i == 0) == false
12	ne_proste = false

	i = 3
13	((i < a_minus) && !(ne_proste1)) == true
14	(a_minus % i == 0) == false
15	ne_proste = false i = 4
...	...
16	ne_proste = false i = 23
17	((i < a_minus) && !(ne_proste1)) == false
18	((j < a_plus) && !(ne_proste2)) == true
...	...
19	(ne_proste1 && ne_proste2) == false
20	(ne_proste1 == false)
21	Виведення: 23
	Кінець

Блок	Дія
	Початок
1	Введення: 35
2	Виведення: 31
	Кінець

Блок	Дія
	Початок
1	Введення: 10
2	Виведення: 11
	Кінець

6) Висновки

Я дослідив особливості роботи складних циклів та набув практичних навичок їх використання під час складання програмних специфікацій. Успішно виконав поставлену задачу.