

Додаток 1

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 складних
циклічних алгоритмів»

Варіант 34

Виконав студент ІП-13 Шиманська Ганна Артурівна
(шифр, прізвище, ім'я, по батькові)

Перевірила Вечерковська Анастасія Сергіївна
(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.

Варіант 34

34. Визначити n перших простих чисел.

- **Постановка задачі**

Залежно від числа n вивести певну кількість простих чисел.

Побудова математичної моделі

Прості числа діляться лише на 1 і самі на себе.

Складемо таблицю змінних

<i>Змінна</i>	<i>Тип</i>	<i>Ім'я</i>	<i>Призначення</i>
Задане натуральне число	long	n	Вхідні дані
Простота числа	bool	prime	Проміжні дані
Ітерації вкладеного циклу	int	i	Проміжні дані
Поточна кількість знайдених простих чисел	long	counter	Проміжні дані
Просте число	long	p	Вихідні дані

Для обчислення кореня застосуємо функцію `sqrt()`.

Для обчислення остачі від ділення використовуємо `%`.

Розв'язання

Програмні специфікації запишемо у псевдокодi та графічній формi у вигляді блок-схеми.

Крок 1. Визначимо основні дії.

Крок 2. Перевіряємо, чи знайдена кількість простих чисел менша за необхідну.

Крок 3. Деталізуємо перевірку на простоту при $p = 1$

Крок 4. Перебираємо дільники числа.

Крок 5. Перевіряємо число на подільність

Крок 6. Перевіряємо, чи є число простим.

- **Псевдокод алгоритму**

Крок 1.

початок

Введення n

Перевіряємо, чи знайдена кількість простих чисел менша за необхідну.

Деталізуємо перевірку на простоту при $p = 1$

Перебираємо дільники числа.

Перевіряємо число на подільність

Перевіряємо, чи є число простим.

Виведення p

кінець

Крок 2.

початок

Введення n

поки $counter < n$

повторити

Деталізуємо перевірку на простоту при $p = 1$

Перебираємо дільники числа.

Перевіряємо число на подільність

Перевіряємо, чи є число простим.

Виведення p

$p = p + 1$

все повторити

кінець

Крок 3.

початок

Введення n

поки $\text{counter} < n$

повторити

$\text{prime} = \text{true}$

якщо $p == 1$

то

$\text{prime} = \text{false}$

все якщо

Перебираємо дільники числа.

Перевіряємо число на подільність

Перевіряємо, чи є число простим.

Виведення p

$p = p + 1$

все повторити

кінець

Крок 4.

початок

Введення n

поки $\text{counter} < n$

повторити

$\text{prime} = \text{true}$

якщо $p == 1$

то

$\text{prime} = \text{false}$

все якщо

повторити для i від 2 до \sqrt{p}

Перевіряємо число на подільність

все повторити

Перевіряємо, чи є число простим.

Виведення p

$p = p + 1$

все повторити

кінець

Крок 5.

початок

Введення n

поки $\text{counter} < n$

повторити

$\text{prime} = \text{true}$

якщо $p == 1$

то

$\text{prime} = \text{false}$

все якщо

повторити для i від 2 до \sqrt{p}

якщо $p \% i == 0$

то

prime = false

все якщо

все повторити

Перевіряємо, чи є число простим.

Виведення p

$p = p + 1$

все повторити

кінець

Крок 6.

початок

Введення n

поки counter < n

повторити

prime = true

якщо $p == 1$

то

prime = false

все якщо

повторити для i від 2 до \sqrt{p}

якщо $p \% i == 0$

то

prime = false

все якщо

все повторити

якщо prime

то

Виведення p

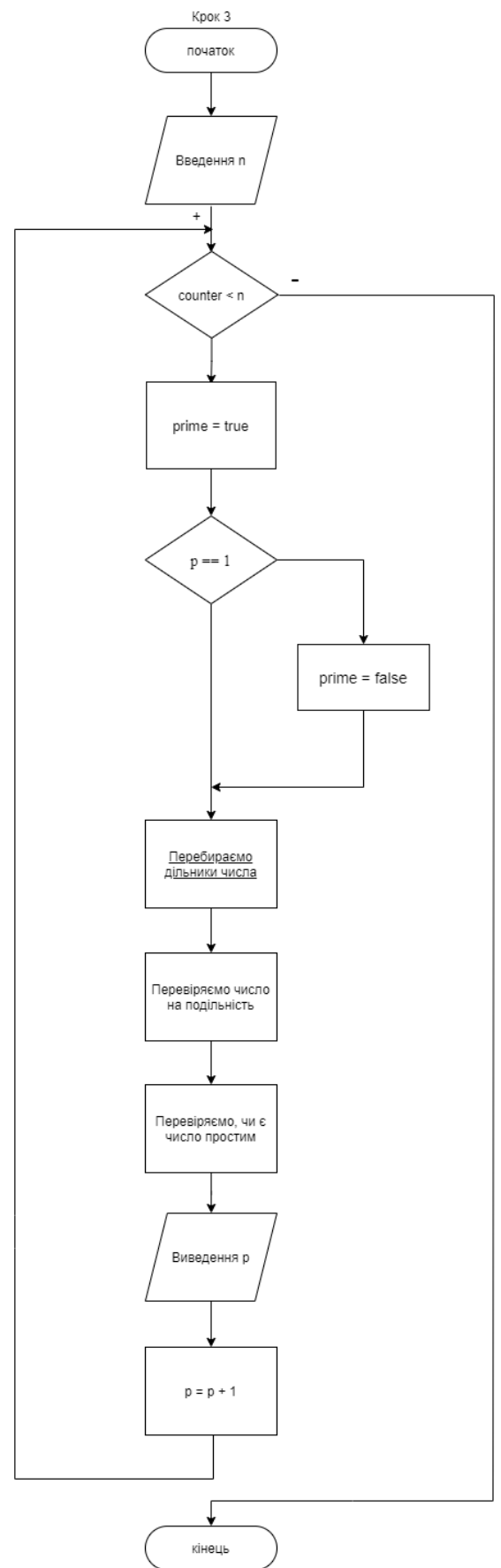
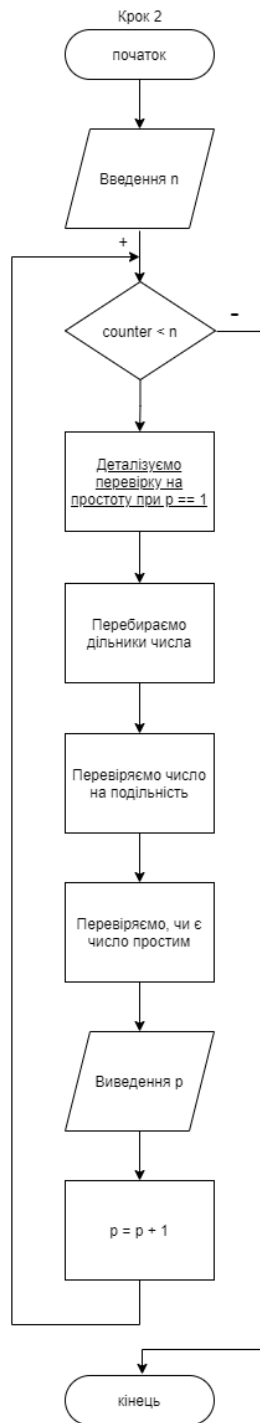
все якщо

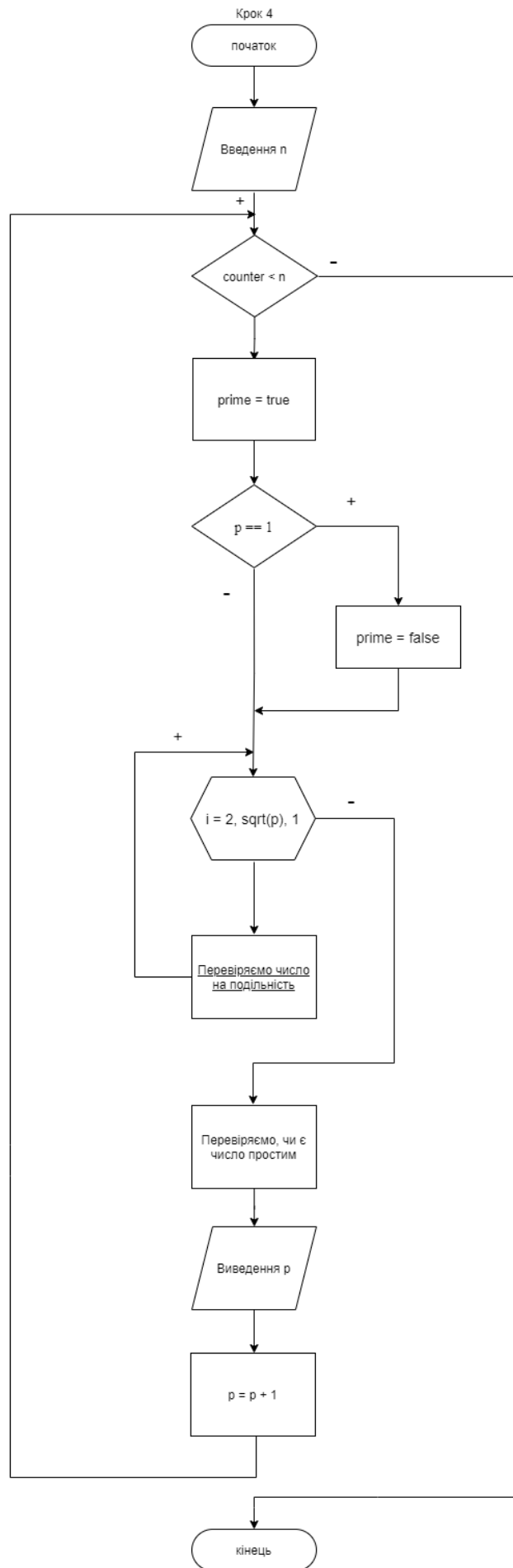
$$p = p + 1$$

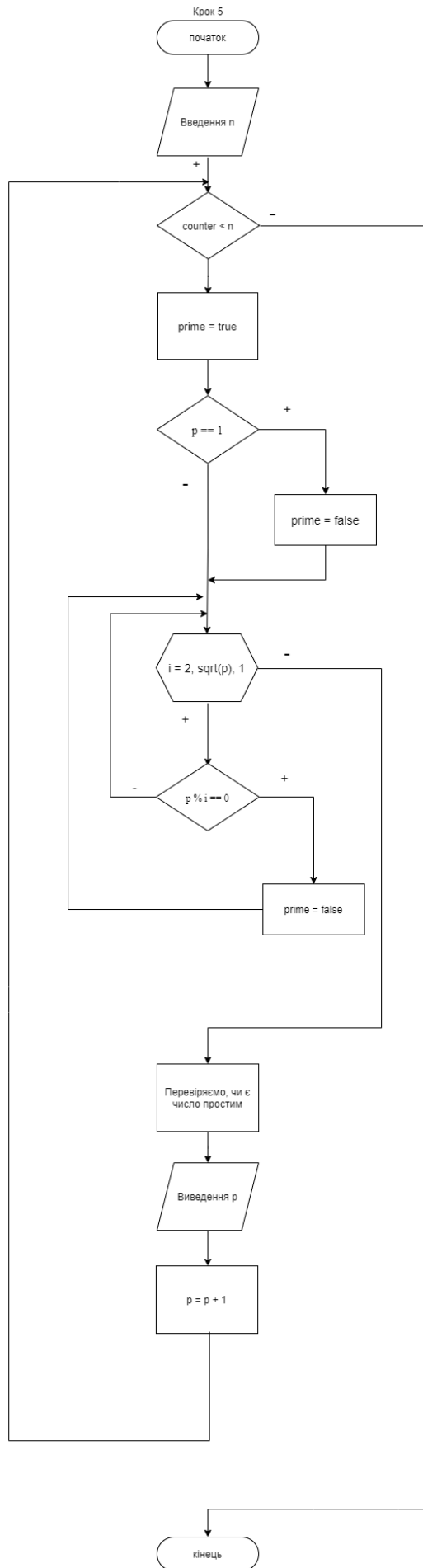
все повторити

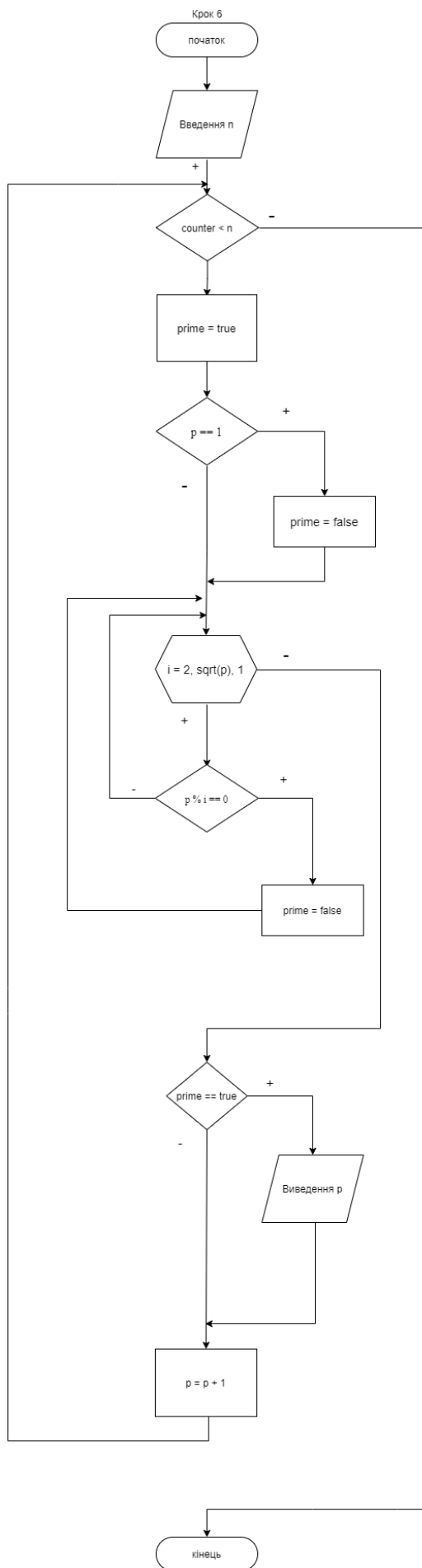
кінець

- **Блок-схема**









- **Випробування алгоритму**

Номер ітерації	Дія
	початок
	Введення $n = 5$
1	$p = 2$ $counter = 0$
2	$p = 3$ $counter = 1$
3	$p = 5$ $counter = 2$
4	$p = 7$ $counter = 3$
5	$p = 11$ $counter = 4$
	...
18	$p = 61$ $counter = 17$
19	$p = 67$ $counter = 18$
20	$p = 71$ $counter = 19$

- **Висновки:**

У цій роботі я дослідила складний циклічний алгоритм, застосувавши його для знаходження перших n простих чисел. Внаслідок роботи алгоритму ми можемо вивести задану

користувачем кількість таких чисел. Використовуючи структури з вкладеними циклами ми застосовуємо складний алгоритм зі з'єднання базових структур.