



## Лабораторна робота 5

### Дослідження складних циклічних алгоритмів

#### Варіант 10

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

#### Задача 10.

Дано натуральне число  $n$ . Знайти всі числа Мерсена, що не перевищують число  $n$ .

Просте число називається числом Мерсена, якщо його можна представити у вигляді:

$$2^p - 1, \text{ де } p \text{ — теж просте число}$$

**Постановка задачі.** Результатом розв'язку є частина послідовності простих чисел Мерсена, для визначення кількості яких, нам необхідне задане значення натурального числа. Інших початкових даних для розв'язку не потрібно.

**Математична побудова.** Складемо таблицю змінних.

Змінна	Тип	Ім'я	Призначення
Число $n$	Цілий	$n$	Початкове дане
Число mersen	Дійсний	mersen	Проміжна змінна
Число $p$	Дійсний	$p$	Проміжна змінна
Змінна isPrime	Булевий	isPrime	Проміжна змінна
Змінна $i$	Дійсний	$i$	Проміжна змінна
Результат mersen	Рядок	mersen	Результат

Таким чином, математичне формулювання задачі зводиться до обчислення членів послідовності(mersen), а для цього треба знайти послідовність простих чисел, значення яких буде приймати показник p. Однак, виконуватися задача буди лише до тих пір, поки число mersen менше за значення n.

### ***Розв'язання:***

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

**Крок 1.** Визначимо основні дії.

**Крок 2.** Деталізуємо знаходження mersen.

**Крок 3.** Деталізуємо дію знаходження простого показнику p.

### ***Псевдокод***

*крок 1*

**початок**

Знаходження простого показнику p

Знаходження mersen

**кінець**

*крок 2*

**початок**

введення n

mersen = 1

**поки mersen < n**

Обчислення простого показнику p

mersen = pow(2, p) - 1

виведення mersen

p ++

**все повторити**

**кінець**

крок 3

введення n

mersen = 1

**поки mersen < n**

boolean isPrime = true

для i=2 поки i<=p/2 виконувати i++

якщо p % i == 0

isPrime = false

**break**

mersen = pow(2, p) - 1

виведення mersen

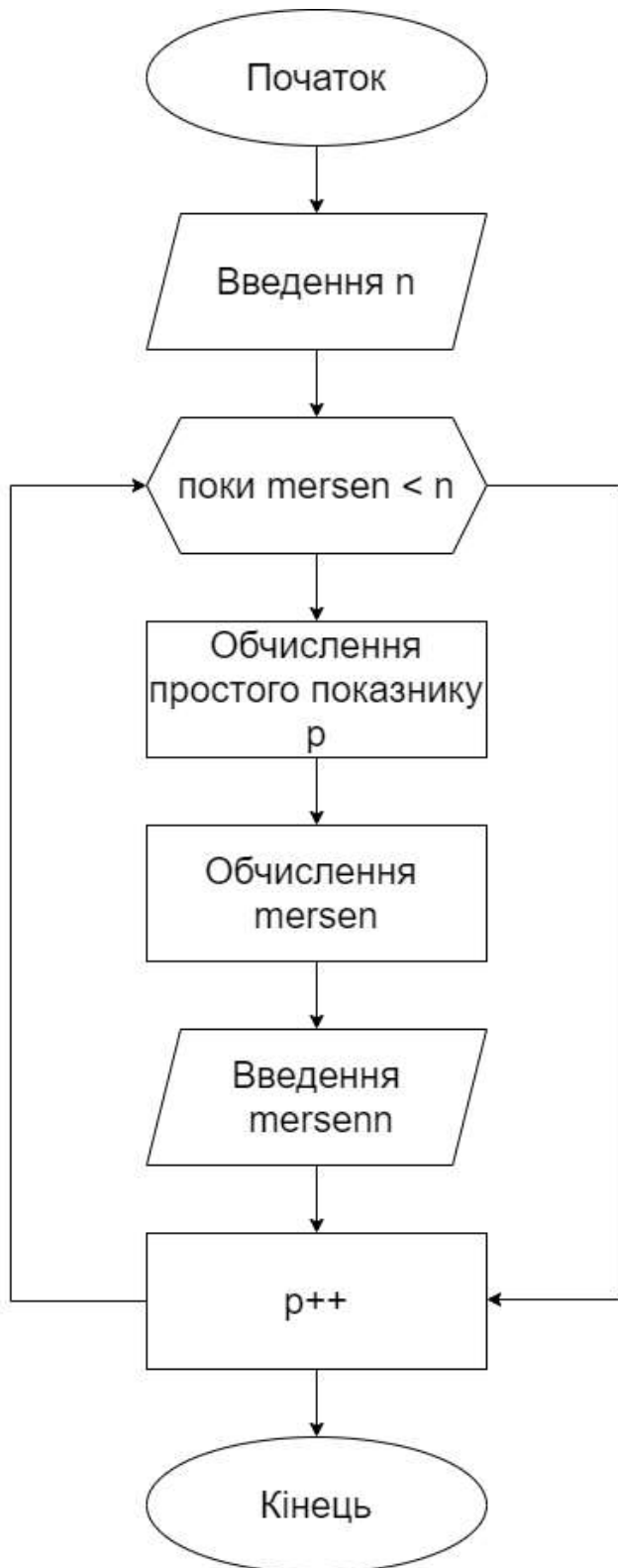
p ++

**все повторити**

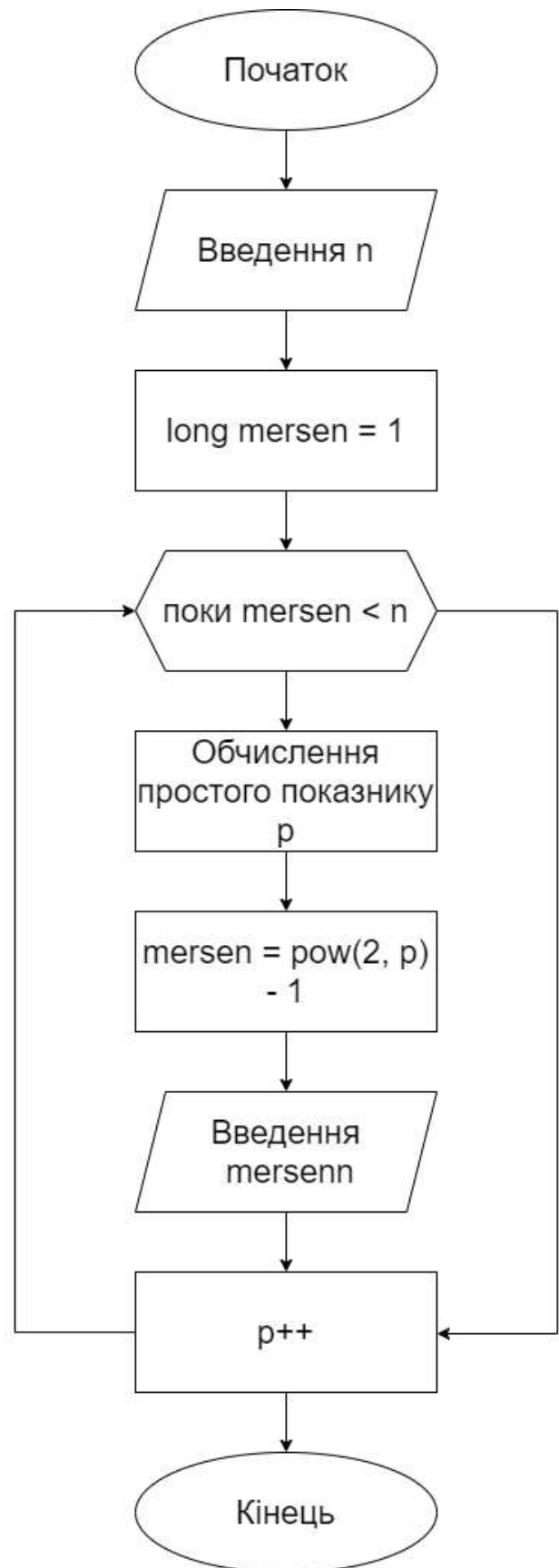
**кінець**

## Блок-схема

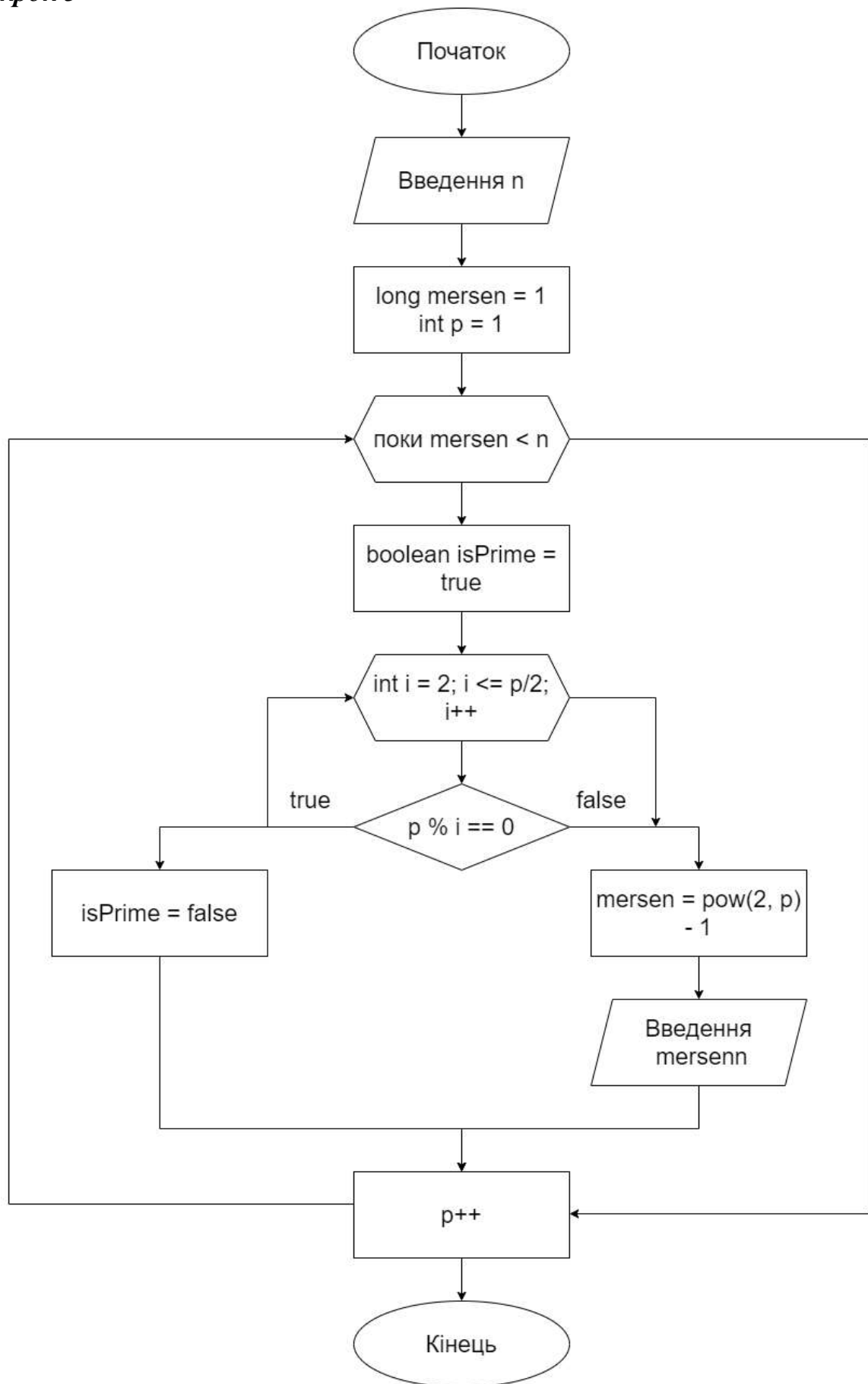
крок1



крок2



крок 3



## Випробування алгоритму

Блок	Дія
1	Початок
2	Введення $n = 35$
3	$\text{long mersen} = 1$
4	$\text{Int } p = 1$
5	$\text{boolean isPrime} = \text{true}$
6	$\text{Mersen} = 2-1=1$
7	Вивід: 1
8	$P = p + 1 = 2$
9	$\text{boolean isPrime} = \text{true}$
10	$\text{Mersen} = 4-1 = 3$
11	Вивід: 3
12	$P = p + 1 = 3$
13	$\text{boolean isPrime} = \text{true}$
14	$\text{Mersen} = 8 - 1 = 7$
15	Вивід: 7
16	$P = p + 1 = 4$
17	$\text{boolean isPrime} = \text{false}$
18	$P = p + 1 = 5$
19	$\text{boolean isPrime} = \text{true}$
20	$\text{Mersen} = 32 - 1 = 31$
21	Вивід: 31
22	Кінець (результат - виведення 1 3 7 31 )

## Висновок

Під час виконання лабораторної було досліджено подання керувальної дії чергування у вигляді умовної та альтернативної форм та набуто практичних навичок їх використання під час складання програмних специфікацій. Покращено навички написання псевдокоду, побудови та тестування алгоритмів.