
Основи програмування – 1. Алгоритми та структури даних

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 13

Виконав студент ІП-13, Жмайло Дмитро Олександрович
(шифр, прізвище, ім'я, по батькові)

Перевірила Вечерковська Анастасія Сергіївна
(прізвище, ім'я, по батькові)

Лабораторна робота 5

Дослідження складних циклічних алгоритмів

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

Індивідуальне завдання

Варіант 13

13. Натуральне число називається паліндромом, якщо його запис читається однаково з початку та з кінця (наприклад, 575, 9). Знайти всі числа-паліндроми, що не перевищують 100, та при піднесенні до квадрату також дають паліндроми.

Постановка задачі

Задані натуральні числа на інтервалі від 1 до 100 включно. Необхідно знайти числа, які є паліндромами і залишаються ними при піднесенні до квадрату. Будемо перевіряти всі числа від 1 до 100.

- 1) Для початку необхідно перевірити чи є поточне число паліндромом
- 2) Визначити чи є квадрат паліндрома паліндромом

І якщо число задовольняє умовам 1 і 2, то воно задовольняє умову задачі й ми можемо його вивести й приступити до перевірки іншого числа.

Якщо число не задовольняє умові 1 або 2, то ми можемо приступити до перевірки наступного числа

Побудова математичної моделі

Відповідно до умови складемо таблицю змінних:

| <i>Змінна</i> | <i>Тип</i> | <i>Назва</i> | <i>Призначення</i> |
|--------------------------|------------|-----------------|--------------------|
| Проміжне число i^2 | Цілий | sqrd_number | Проміжні дані |
| Обернене число | Цілий | reversed_number | Проміжні дані |
| Лічильник i | Цілий | i | Проміжні дані |
| Поточне значення розряду | Цілий | current | Проміжні дані |
| Результат | Цілий | result | Вихідні дані |

Для розрахунків введемо допоміжні функції:

- Функція знаходження остачі від ділення числа **a** на **b**: **Mod(a, b)**;
- Функція знаходження цілочисельної частки від ділення числа **a** на **b**:

Div(a, b);

- Функція піднесення числа **a** до степеня **b**: **Pow(a, b)**

Дія 1)

1.1) Для знаходження паліндромів чисел від **1** до **100** достатньо перевірити рівність першого й останнього розрядів числа. Для цього необхідно поділити поточне число на 10 за допомогою **цілочисельного ділення** та **ділення з остачею** і порівняти результати. Якщо результати однакові, то число є паліндромом (ця умова справджується й до 100)

1.2) Оскільки числа від 1 до 9 мають лише один розряд і вони завжди є паліндромами, тому що читаються однаково з двох сторін, через що введемо додаткову перевірку на наявність числа < 10 через сполучник або).

Якщо число не задовольняє хоча б одній з умов (**1.1**) та **1.2**), то воно не є паліндромом й ми переходимо до перевірки наступного числа. Якщо число задовольняє хоча б одній з умов, то ми переходимо до дії **2**.

Дія 2)

Для того, щоб перевірити чи є квадрат паліндрома паліндромом, зробимо наступні дії: прирівняємо значення змінної **sqrd_number** до числа **i** в квадраті; прирівняємо значення змінної **result** до **i**; **reversed_number** до **0**;

Далі необхідно за допомогою циклу:

1. відділяти покроково значення останнього розряду за допомогою функції **Mod(a, b)** й присвоювати це значення змінній **current**
2. присвоювати змінній **reversed_number** (число, яке ми читаємо ззаду наперед для перевірки наявності паліндромів) значення **(reversed_number * 10) + current** для перенесення минулого розряду “розвернутого” числа на розряд вище і додавання поточного розряду
3. необхідно цілочисельно поділити змінну **sqrd_number** на 10 для відкидання опрацьованого програмою розряду

Виконувати цей цикл будемо доти, доки значення змінної

sqrd_number > 0

Далі необхідно знову присвоїти змінній **sqrd_number** значення змінної **i**, піднесеної до квадрату для перевірки правильності виконання умови. **Якщо** значення змінних **sqrd_number** і **reversed_number** однакове, то умова виконана правильно й ми виводимо значення **result** й переходимо до перевірки наступного значення числа **i**. Якщо умова не виконується, то переходимо до наступного значення числа **i**, доки його значення не стане рівним числу **100**, після перевірки якого ми можемо завершити виконання програми

Розв'язання:

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

Крок 1. Визначимо основні дії;

Крок 2. Деталізуємо дію перевірки чисел від 1 до 100;

Крок 3. Деталізуємо дію знаходження паліндромів чисел від 1 до 100;

Крок 4. Деталізуємо дію присвоєння значень змінним `sqrд_number`, `reversed_number`, `result`;

Крок 5. Деталізуємо дію знаходження паліндромів серед квадратів паліндромів;

Крок 6. Деталізуємо дію перевірки виконання умови.

Псевдокод:

Крок 1

початок

перевірка чисел від 1 до 100

знаходження паліндромів чисел від 1 до 100

присвоєння значень змінним `sqrд_number`, `reversed_number`, `result`

знаходження паліндромів серед квадратів паліндромів

перевірка виконання умови

кінець

Крок 2

початок

повторити для i від 1 до number (включно) з кроком 1

знаходження паліндромів чисел від 1 до 100

присвоєння значень змінним sqrd_number, reversed_number, result

знаходження паліндромів серед квадратів паліндромів

перевірка виконання умови

все повторити

кінець

Крок 3

початок

повторити для i від 1 до number (включно) з кроком 1

якщо $(\text{Div}(i, 10) == \text{Mod}(i, 10)) \parallel (i < 10)$

то

присвоєння значень змінним sqrd_number, reversed_number, result

знаходження паліндромів серед квадратів паліндромів

перевірка виконання умови

все якщо

все повторити

кінець

Крок 4

початок

повторити для i від 1 до number (включно) з кроком 1

якщо $(\text{Div}(i, 10) == \text{Mod}(i, 10)) \parallel (i < 10)$

то

$\text{sqr_number} := \text{Pow}(i, 2)$

$\text{reversed_number} := 0$

$\text{result} := i$

знаходження паліндромів серед квадратів паліндромів

перевірка виконання умови

все якщо

все повторити

кінець

Крок 5

початок

повторити для i від 1 до number (включно) з кроком 1

якщо $(\text{Div}(i, 10) == \text{Mod}(i, 10)) \parallel (i < 10)$

то

$\text{sqr_d_number} := \text{Pow}(i, 2)$

$\text{reversed_number} := 0$

$\text{result} := i$

поки $\text{sqr_d_number} > 0$

$\text{current} := \text{Mod}(\text{sqr_d_number}, 10)$

$\text{reversed_number} := (\text{reversed_number} * 10) + \text{current}$

$\text{sqr_d_number} := \text{Div}(\text{sqr_d_number}, 10)$

все поки

перевірка виконання умови

все якщо

все повторити

кінець

Крок 6

початок

повторити для i від 1 до number (включно) з кроком 1
якщо $(\text{Div}(i, 10) == \text{Mod}(i, 10)) \parallel (i < 10)$
то
 $\text{sqr_d_number} := \text{Pow}(i, 2)$
 $\text{reversed_number} := 0$
 $\text{result} := i$
поки $\text{sqr_d_number} > 0$
 $\text{current} := \text{Mod}(\text{sqr_d_number}, 10)$
 $\text{reversed_number} := (\text{reversed_number} * 10) + \text{current}$
 $\text{sqr_d_number} := \text{Div}(\text{sqr_d_number}, 10)$
все поки
 $\text{sqr_d_number} := \text{Pow}(i, 2)$
якщо $\text{sqr_d_number} == \text{reversed_number}$
то
 Виведення result
все якщо
все якщо
все повторити

кінець

Блок-схема:

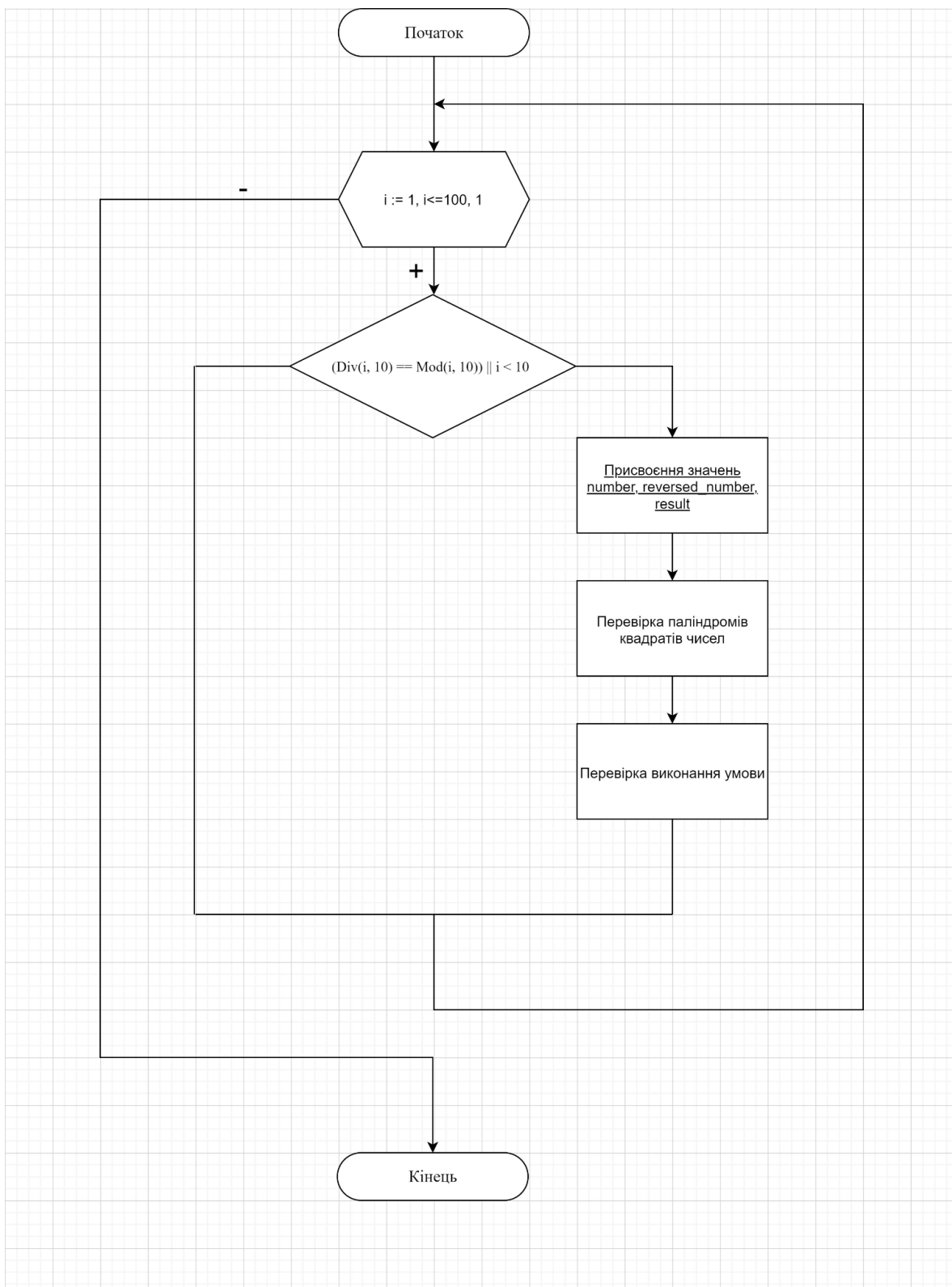
Крок 1



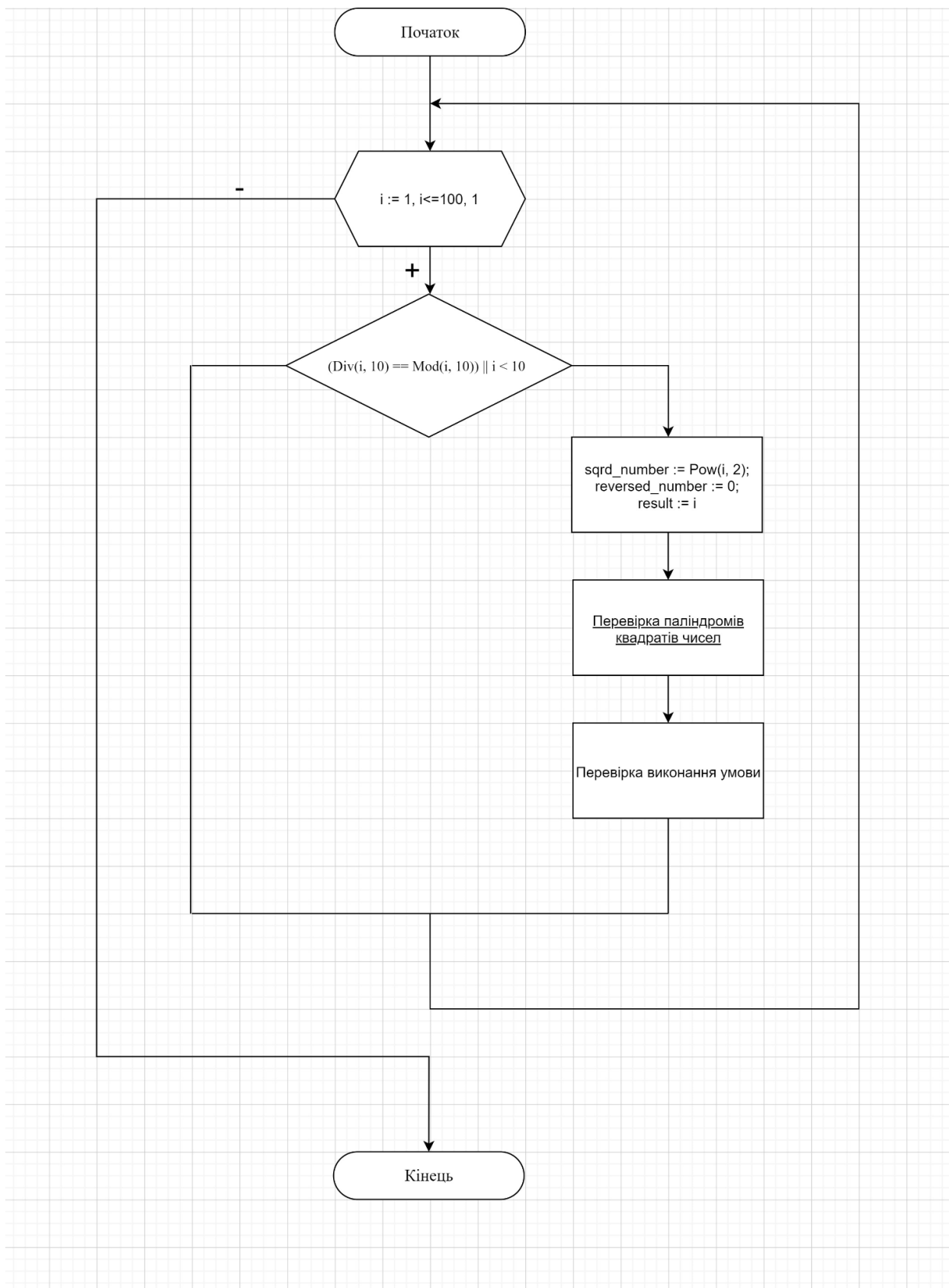
Крок 2



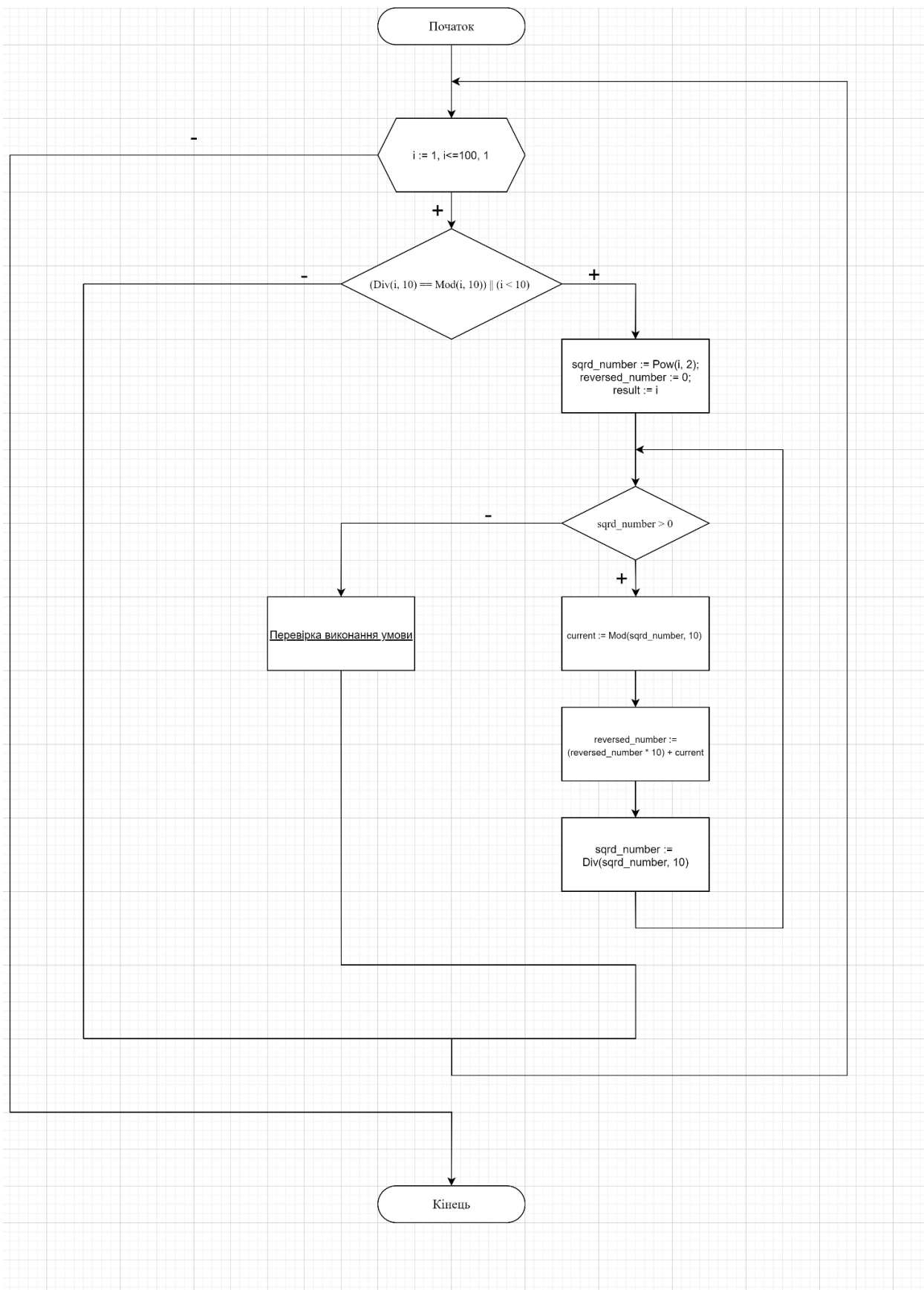
Крок 3



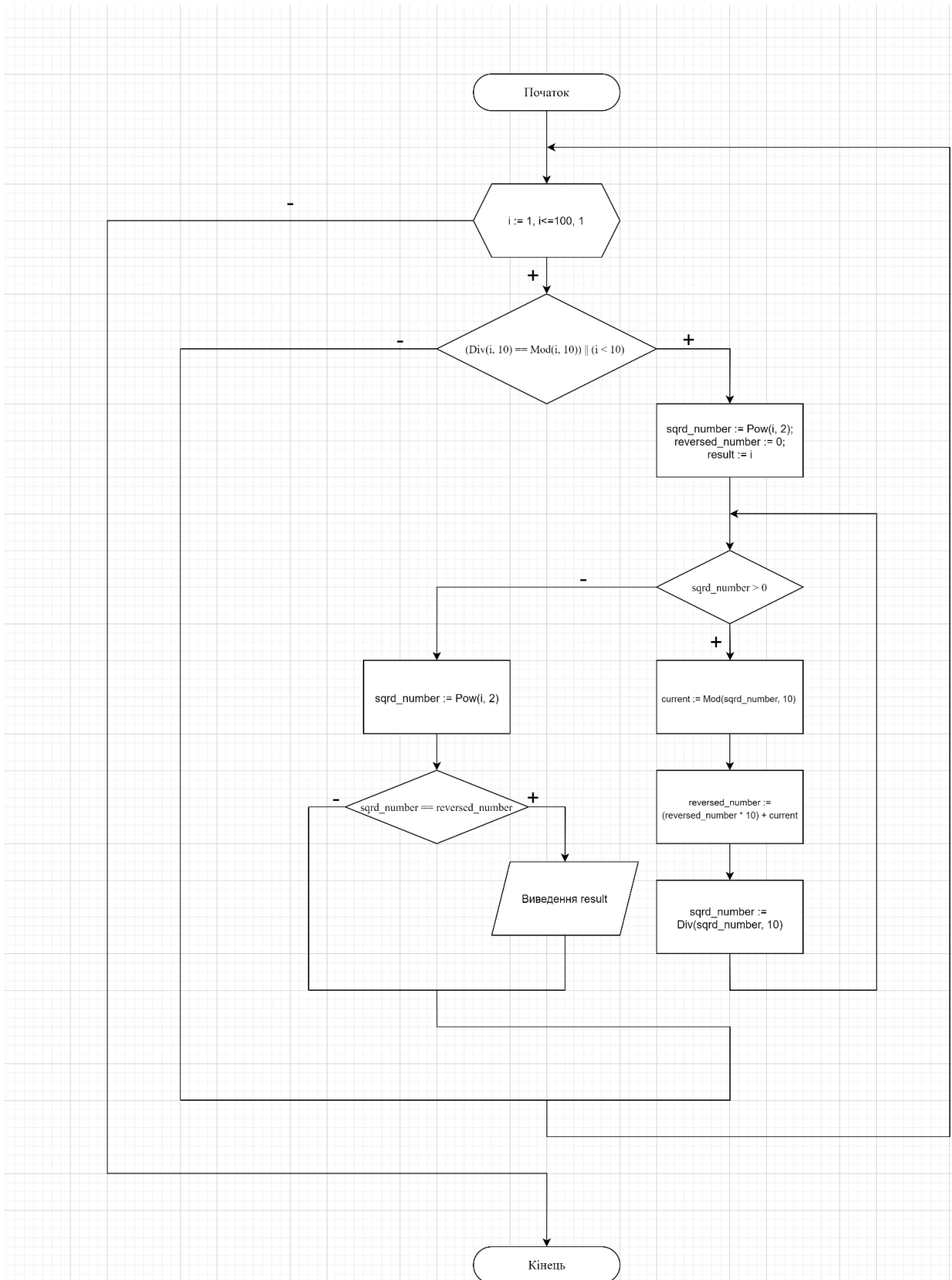
Крок 4



Крок 5



Крок 6



Випробування алгоритму:

| Блок | Дія |
|------|--|
| | Початок |
| 1 | i = 1 |
| 2 | $(\text{Div}(\mathbf{i}, 10) == \text{Mod}(\mathbf{i}, 10)) \parallel (\mathbf{i} < 10)$ виконується |
| 3 | reversed_number = 1, sqrd_number = 1 |
| 4 | reversed_number == sqrd_number виконується |
| 5 | Виведення 1 |
| 6 | i = 2 |
| 7 | $(\text{Div}(\mathbf{i}, 10) == \text{Mod}(\mathbf{i}, 10)) \parallel (\mathbf{i} < 10)$ виконується |
| 8 | reversed_number = 4, sqrd_number = 4 |
| 9 | reversed_number == sqrd_number виконується |
| 10 | Виведення 2 |
| 11 | i = 3 |

| | |
|----|---|
| 12 | $(\text{Div}(\mathbf{i}, 10) == \text{Mod}(\mathbf{i}, 10)) \parallel (\mathbf{i} < 10)$ виконується |
| 13 | reversed_number = 9, sqrd_number = 9 |
| 14 | reversed_number == sqrd_number виконується |
| 15 | Виведення 3 |
| 16 | i = 4 |
| 17 | $(\text{Div}(\mathbf{i}, 10) == \text{Mod}(\mathbf{i}, 10)) \parallel (\mathbf{i} < 10)$ виконується |
| 18 | reversed_number = 61, sqrd_number = 16 |
| 19 | reversed_number == sqrd_number не виконується |
| 20 | ... i = 10 |
| 21 | $(\text{Div}(\mathbf{i}, 10) == \text{Mod}(\mathbf{i}, 10)) \parallel (\mathbf{i} < 10)$ не виконується |
| 22 | i = 11 |
| 23 | $(\text{Div}(\mathbf{i}, 10) == \text{Mod}(\mathbf{i}, 10)) \parallel (\mathbf{i} < 10)$ |

Основи програмування – 1. Алгоритми та структури даних

| | |
|----|---|
| | виконується |
| 24 | reversed_number = 121, sqrd_number = 121 |

| | |
|----|---|
| 25 | reversed_number == sqrd_number виконується |
| 26 | Виведення 11 |
| 27 | ... i = 22 |
| 28 | (Div(i, 10) == Mod(i, 10)) (i < 10) виконується |
| 29 | reversed_number = 484, sqrd_number = 484 |
| 30 | reversed_number == sqrd_number виконується |
| 31 | Виведення 22 |
| 32 | ... i = 100 |
| 33 | (Div(i, 10) == Mod(i, 10)) (i < 10) не виконується |
| | Кінець |

Висновок:

На цій лабораторній роботі ми дослідили особливості роботи складних циклів та набули практичних навичок їх використання під час складання програмних специфікацій; навчилися оформлювати складні цикли в програмах у вигляді блок-схем та псевдокоду. Виконали програму покроково відповідно до нашого алгоритму й перевірили результати.