

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 29

Виконав студент ІП-13 Романюк Діана Олексіївна  
(шифр, прізвище, ім'я, по батькові)

Перевірів \_\_\_\_\_  
( 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.

#### Варіант 29

#### Постановка задачі:

Знайти всі чотирьох значні (числа від 1000 до 9999) паліндроми (числа, які у віддзеркаленому вигляді дорівнюють самі собі).

#### Побудова математичної моделі:

#### Таблиця імен змінних

Змінна	Тип	Ім'я	Призначення
Лічильник, число-паліндром	Цілий	Num	Вихідні дані
Зворотній вигляд змінної Num	Цілий	RevsNum	Проміжні дані

**Num** – число від 1000 до 9999, що змінюється на одиницю під час виконання зовнішнього циклу.

**RevsNum** – відзеркалене число **Num**

**%** - остача від ділення;

**\** - ціла частина від ділення;

**RevsNum** = (Num % 10) \* 1000 відокремлюємо одиниці від числа Num і перетворюємо їх на тисячі за допомогою множення. Потім відокремлюємо число десятків і перетворюємо їх у сотні ((Num % 100) / 10)\*100. Відокремлюємо число сотень і перетворюємо їх у десятки ((Num % 1000) / 100)\*10. Відокремлюємо число тисяч – одиниці Num / 1000. Додаємо і отримуємо формулу для обчислення зворотнього числа:

**RevsNum** = (Num % 10) \* 1000 + ((Num % 100) / 10)\*100 + ((Num % 1000) / 100)\*10 + Num / 1000;

#### Задача:

Вивести всі паліндроми у порміжку від 1000 до 9999;

#### Розв'язання

Крок 1. Визначимо основні дії

Крок 2. Деталізуємо дію задання умови першого циклу для перевірки входження числа у заданий проміжок.

Крок 3. Деталізуємо дію обчислення зворотнього числа від поточного.

Крок 4. Деталізуємо дію перевірки умови другого циклу для визначення чи дорівнює поточне число зворотньому.

## Псевдокод алгоритму:

### Крок 1

#### початок

оголошення Num, RevsNum  
задання умови першого циклу  
обчислення зворотнього числа від поточного  
перевірки умови другого циклу  
виведення Num

#### кінець

### Крок 2

#### початок

оголошення Num, RevsNum

#### повторити

для Num від 1000 до 9999

обчислення зворотнього числа від поточного  
перевірки умови другого циклу  
виведення Num

#### все повторити

#### кінець

### Крок 3

#### початок

оголошення Num, RevsNum

#### повторити

для Num від 1000 до 9999

**RevsNum** = (Num % 10) \* 1000 + ((Num % 100) / 10) \* 100 +  
+ ((Num % 1000) / 100) \* 10 + Num / 1000  
перевірки умови другого циклу  
виведення Num

#### все повторити

#### кінець

### Крок 4

#### початок

оголошення Num, RevsNum

#### повторити

для Num від 1000 до 9999

**RevsNum** = (Num % 10) \* 1000 + ((Num % 100) / 10) \* 100 +  
+ ((Num % 1000) / 100) \* 10 + Num / 1000

**якщо** Num = RevsNum

то виведення Num

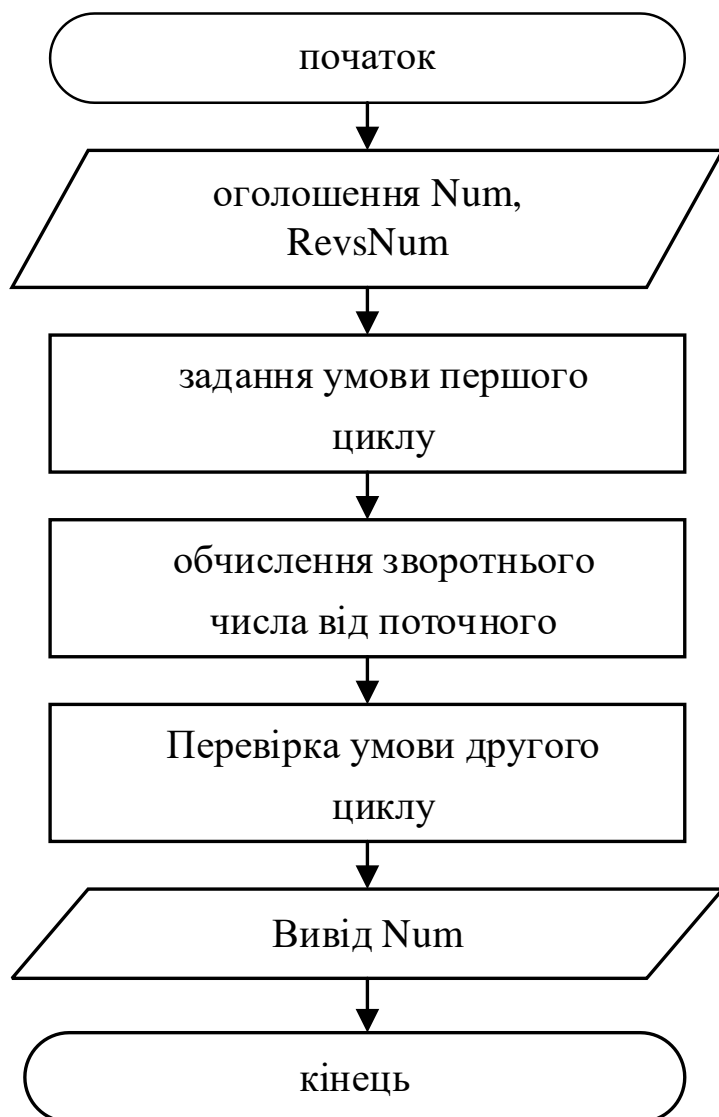
**все якщо**

#### все повторити

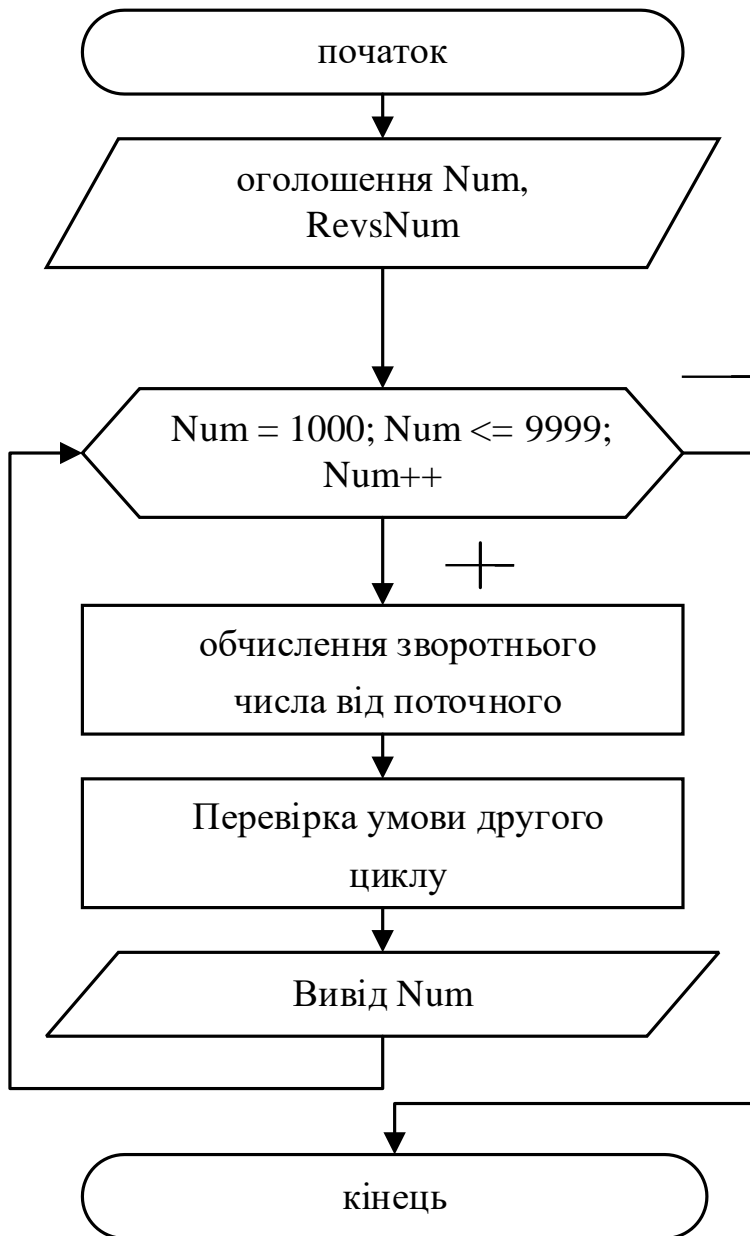
#### кінець

**Блок-схема:**

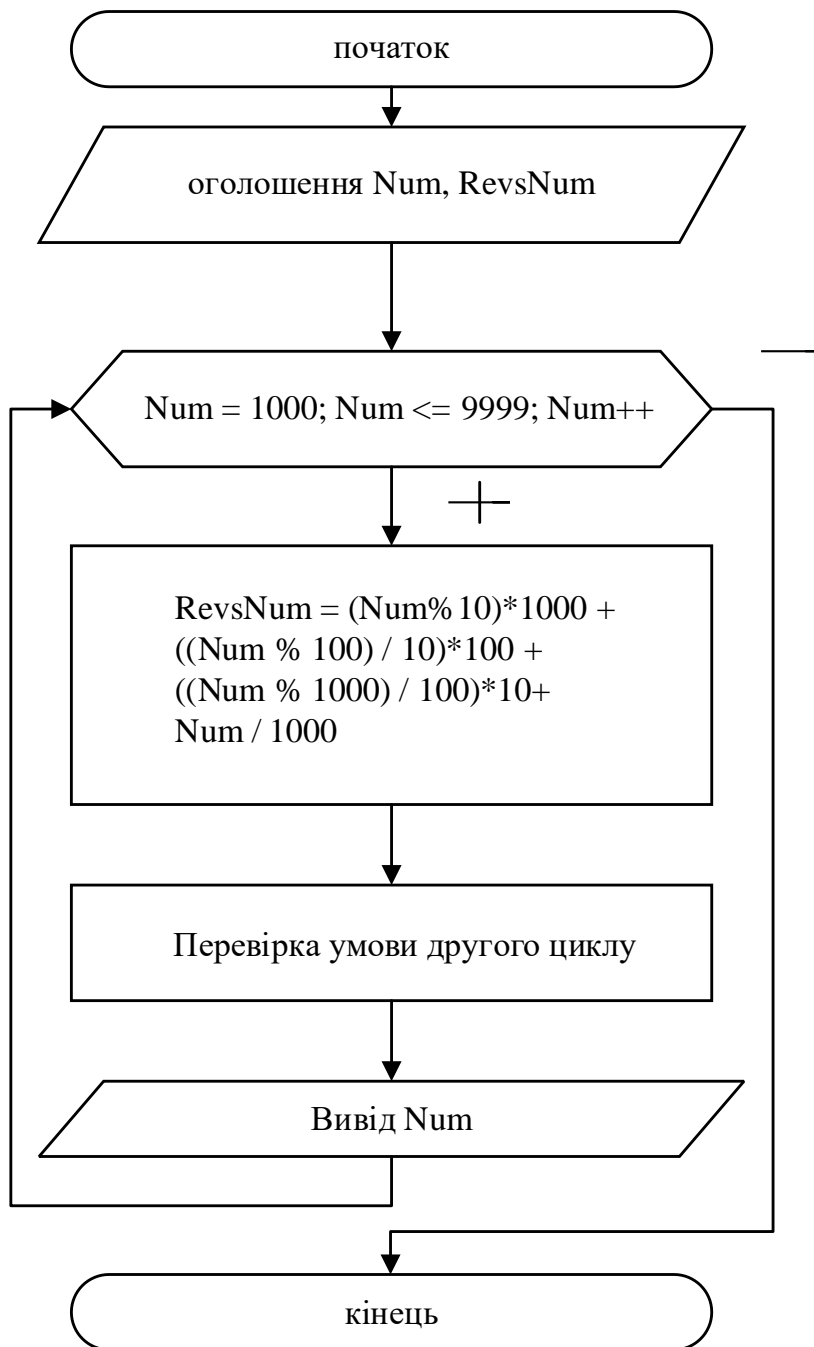
*Крок 1*



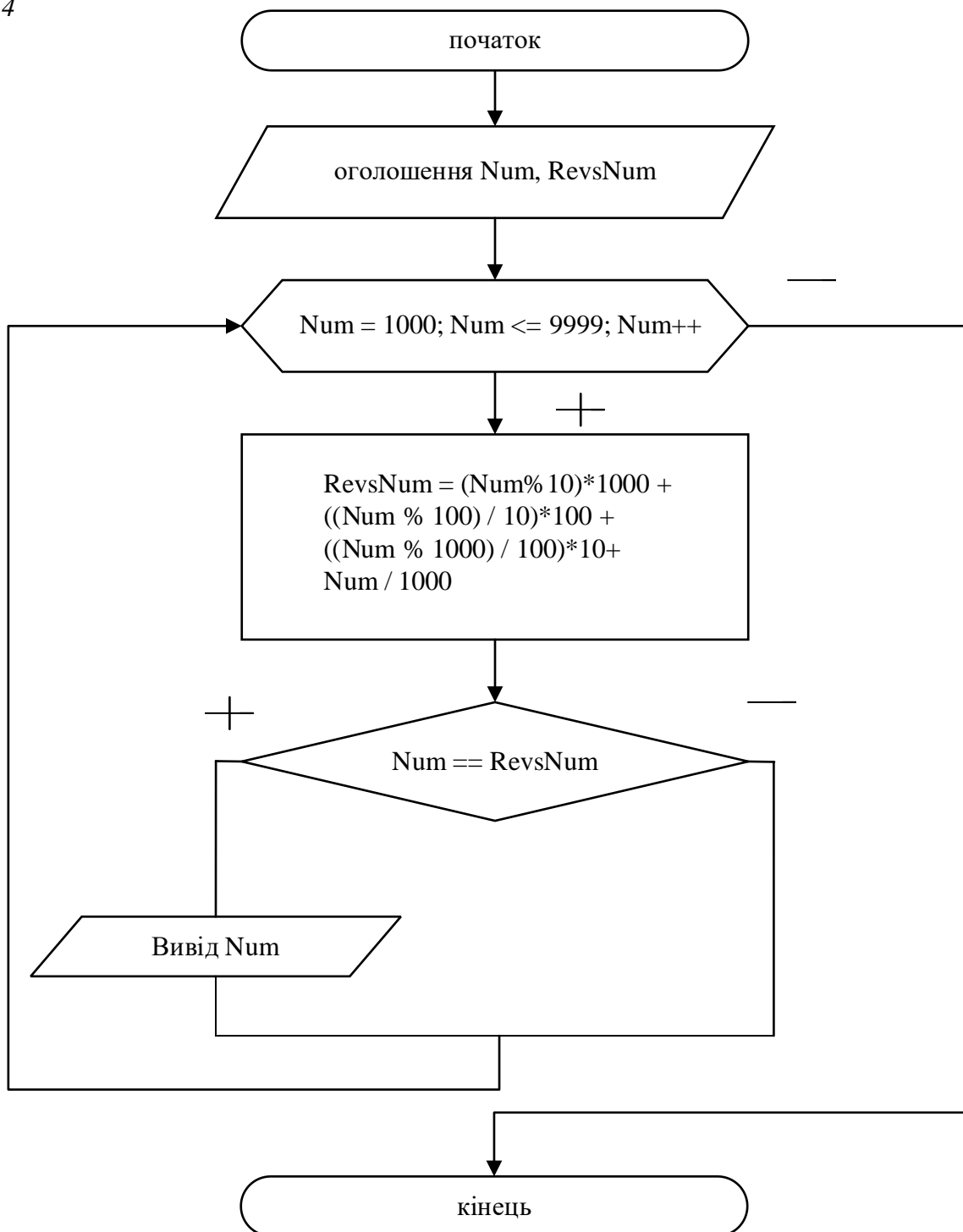
Крок 2



Крок 3



Крок 4



**Випробування алгоритму:**

Блок	Дія
	Початок
1	Вивід 1001 1111 1221 1331 1441 1551 1661 1771 1881 1991 2002 2112 2222 2332 2442 2552 2662 2772 2882 2992 3003 3113 3223 3333 3443 3553 3663



	3773
	3883
	3993
	4004
	4114
	4224
	4334
	4444
	4554
	4664
	4774
	4884
	4994
	5005
	5115
	5225
	5335
	5445
	5555
	5665
	5775
	5885
	5995
	6006
	6116
	6226
	6336
	6446
	6556
	6666
	6776
	6886
	6996
	7007
	7117
	7227

	7337
	7447
	7557
	7667
	7777
	7887
	7997
	8008
	8118
	8228
	8338
	8448
	8558
	8668
	8778
	8888
	8998
	9009
	9119
	9229
	9339
	9449
	9559
	9669
	9779
	9889
	9999
	Кінець

## **Висновки:**

Під час виконання лабораторної роботи досліджено особливості роботи складних циклів. Отримано практичні навички їх використання під час складання лінійних програмних специфікацій. Побудовано математичну модель задачі та таблицю імен змінних. Розроблено псевдокод вирішення даної математичної задачі. Умовно розбито виконання коду на кроки, а також описано його виконання за допомогою створення відповідної блок-схеми. Перевірено умовне виконання коду за допомогою випробування алгоритму.