

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)

Перевірив 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.

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	Проміжні/вихідні дані
Лічильник для обчислення зворотнього вигляду числа Num	Цілий	Num2	Проміжні дані
Зворотній вигляд змінної Num	Цілий	RevsNum	Проміжні дані

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

Задаємо новій змінній **Num2** значення Num1 для обчислення зворотнього числа.

Задаємо цикл для Num 2 > 0:

RevsNum – відзеркалене число **Num**

% - остача від ділення;

\ - ціла частина від ділення;

RevsNum = RevsNum*10 + Num2%10;

Num2 = Num2\10 запишемо як Num2 /= 10;

Наступним циклом виконується перевірка Num = RevsNum, якщо твердження вірне, то виводимо число, оскільки якщо воно дорівнює зворотньому вигляду самого себе то є паліндромом.

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 ініціалізації проміжних змінних.

Крок 4. Деталізуємо дію задання умови циклу для знаходження зворотнього числа.

Крок 5. Деталізуємо дію обчислення зворотнього числа.

Крок 6. Деталізуємо зміну значення лічильника.

Крок 7. Деталізуємо дію перевірки умови циклу для визначення чи дорівнює поточне число зворотньому.

Псевдокод алгоритму:

Крок 1

початок

задання умови першого циклу

ініціалізація проміжних змінних

задання умови циклу для знаходження зворотнього числа

обчислення зворотнього числа

зміна значення лічильника

задання циклу для перевірки рівності поточного числа зворотньому

виведення Num

кінець

Крок 2

початок

повторити для Num від 1000 до 9999

ініціалізація проміжних змінних

задання умови циклу для знаходження зворотнього числа

обчислення зворотнього числа

зміна значення лічильника

задання циклу для перевірки рівності поточного числа зворотньому

виведення Num

все повторити

кінець

Крок 3

початок

повторити для Num від 1000 до 9999

RevsNum = 0;

Num2 = Num;

задання умови циклу для знаходження зворотнього числа

обчислення зворотнього числа

зміна значення лічильника

задання циклу для перевірки рівності поточного числа зворотньому

виведення Num

все повторити

кінець

Крок 4

початок

повторити для Num від 1000 до 9999

RevsNum = 0;

Num2 = Num;

повторити

обчислення зворотнього числа

зміна значення лічильника

поки Num2 > 0

задання циклу для перевірки рівності поточного числа зворотньому

виведення Num

все повторити

кінець

Крок 5

початок

повторити для Num від 1000 до 9999

RevsNum = 0;

Num2 = Num;

повторити

RevsNum = RevsNum*10 + Num2%10;

зміна значення лічильника

поки Num2 > 0

задання циклу для перевірки рівності поточного числа зворотньому

виведення Num

все повторити

кінець

Крок 6

початок

повторити для Num від 1000 до 9999

RevsNum = 0;

Num2 = Num;

повторити

RevsNum = RevsNum*10 + Num2% 10;

Num2 \= 10;

поки Num2 > 0

задання циклу для перевірки рівності поточного числа зворотньому

виведення Num

все повторити

кінець

Крок 7

початок

повторити для Num від 1000 до 9999

RevsNum = 0;

Num2 = Num;

повторити

RevsNum = RevsNum*10 + Num2% 10;

Num2 \= 10;

поки Num2 > 0

якщо Num = RevsNum

то виведення Num

все якщо

все повторити

кінець

Блок-схема:

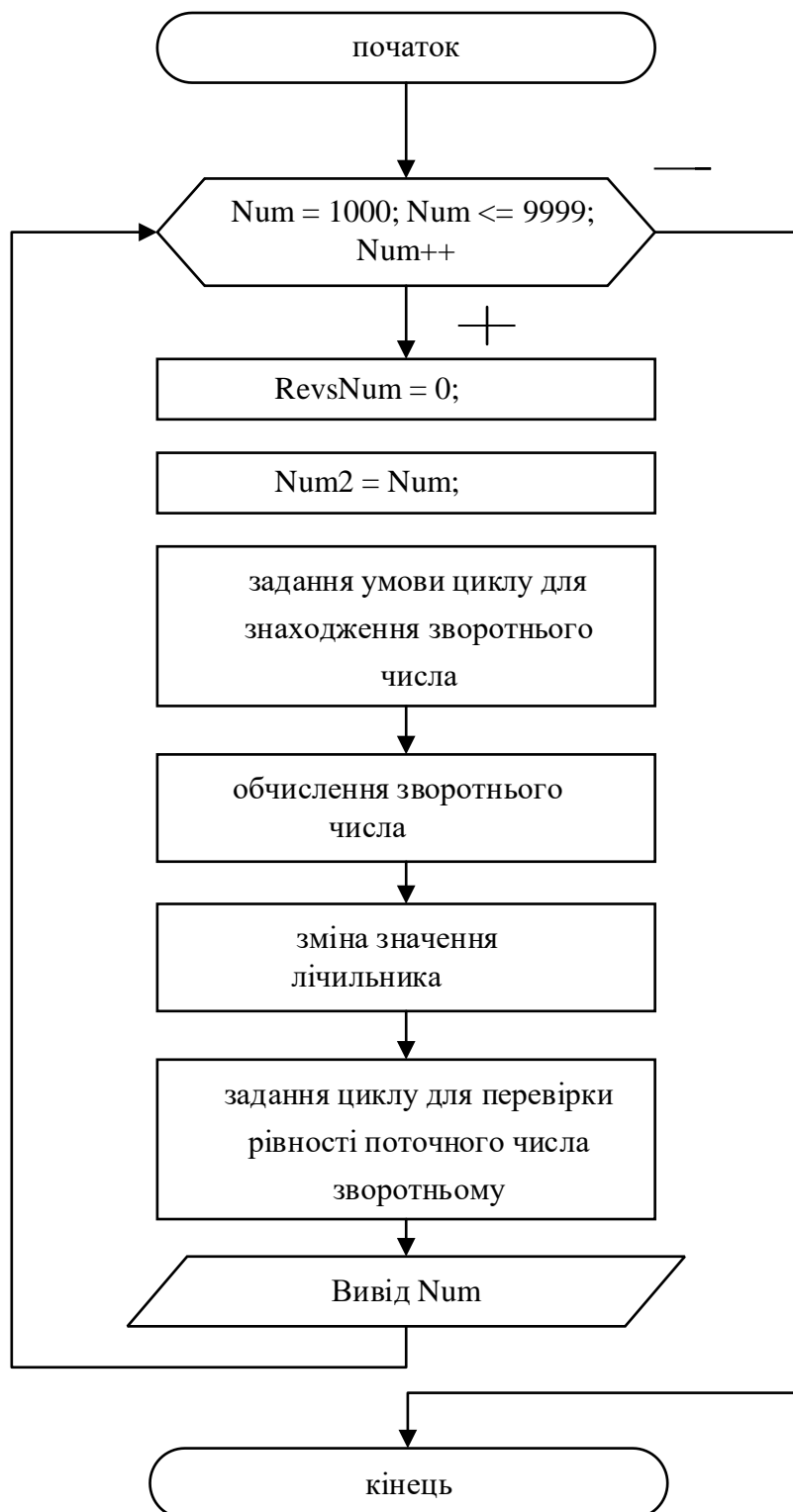
Крок 1



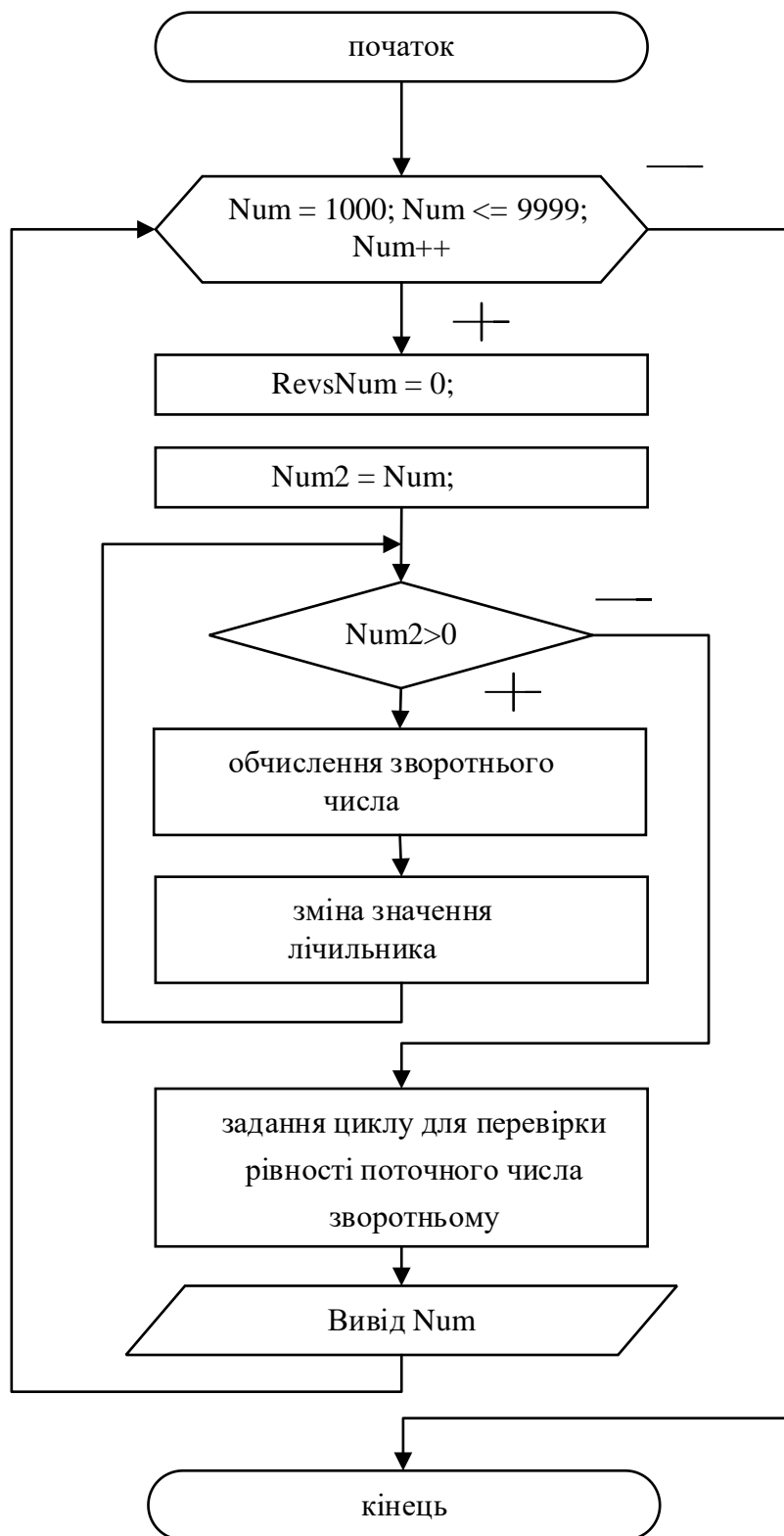
Крок 2



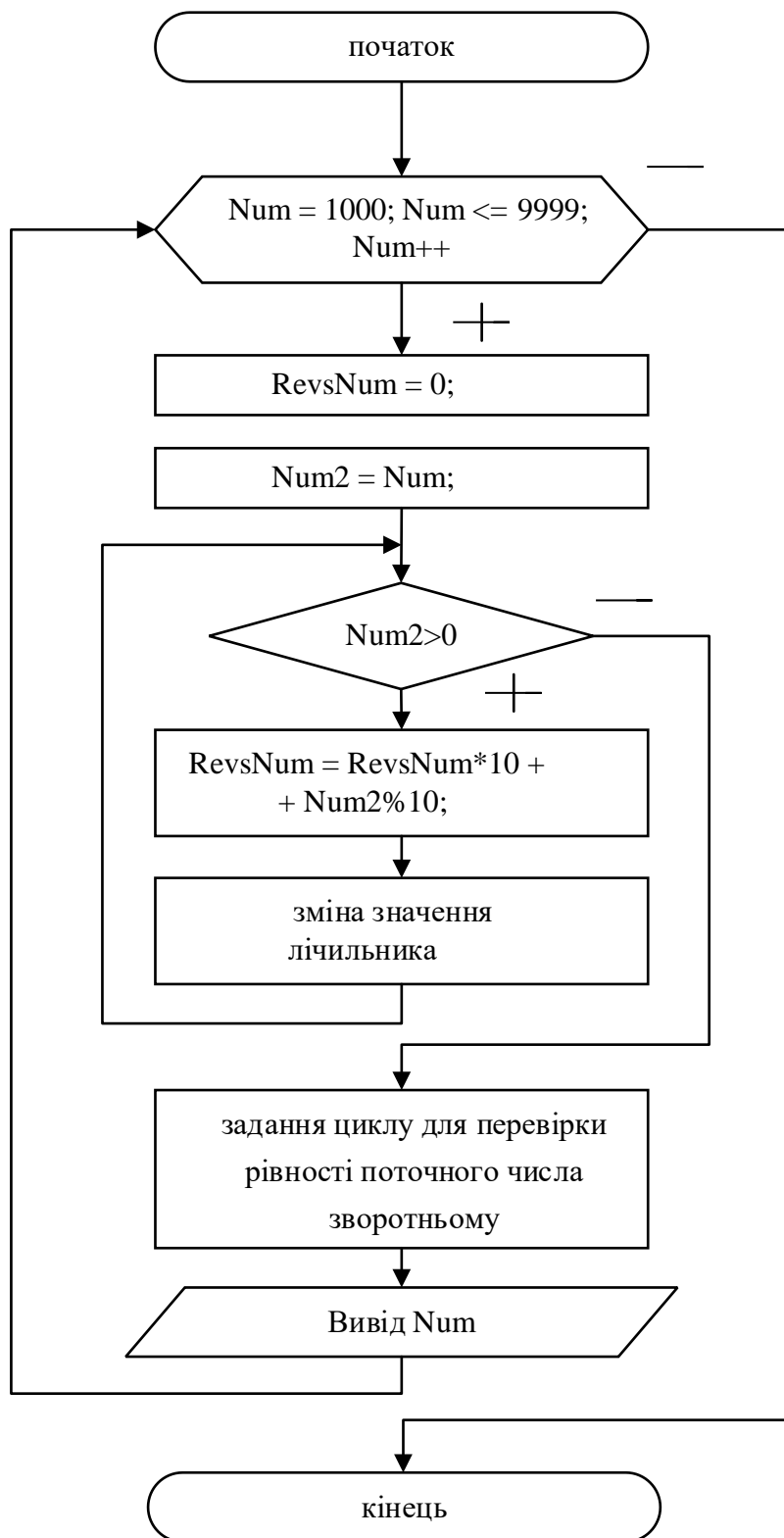
Крок 3



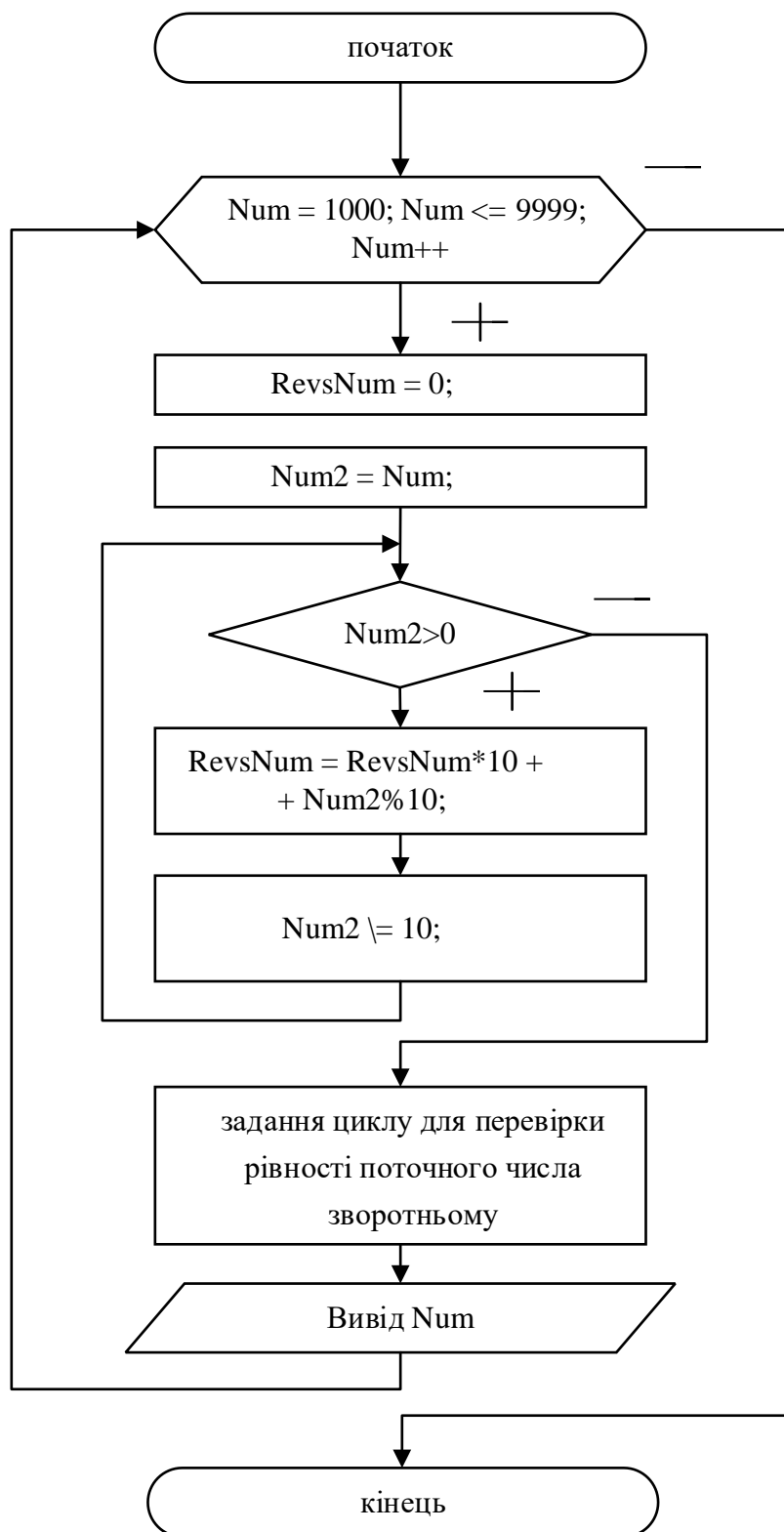
Крок 4



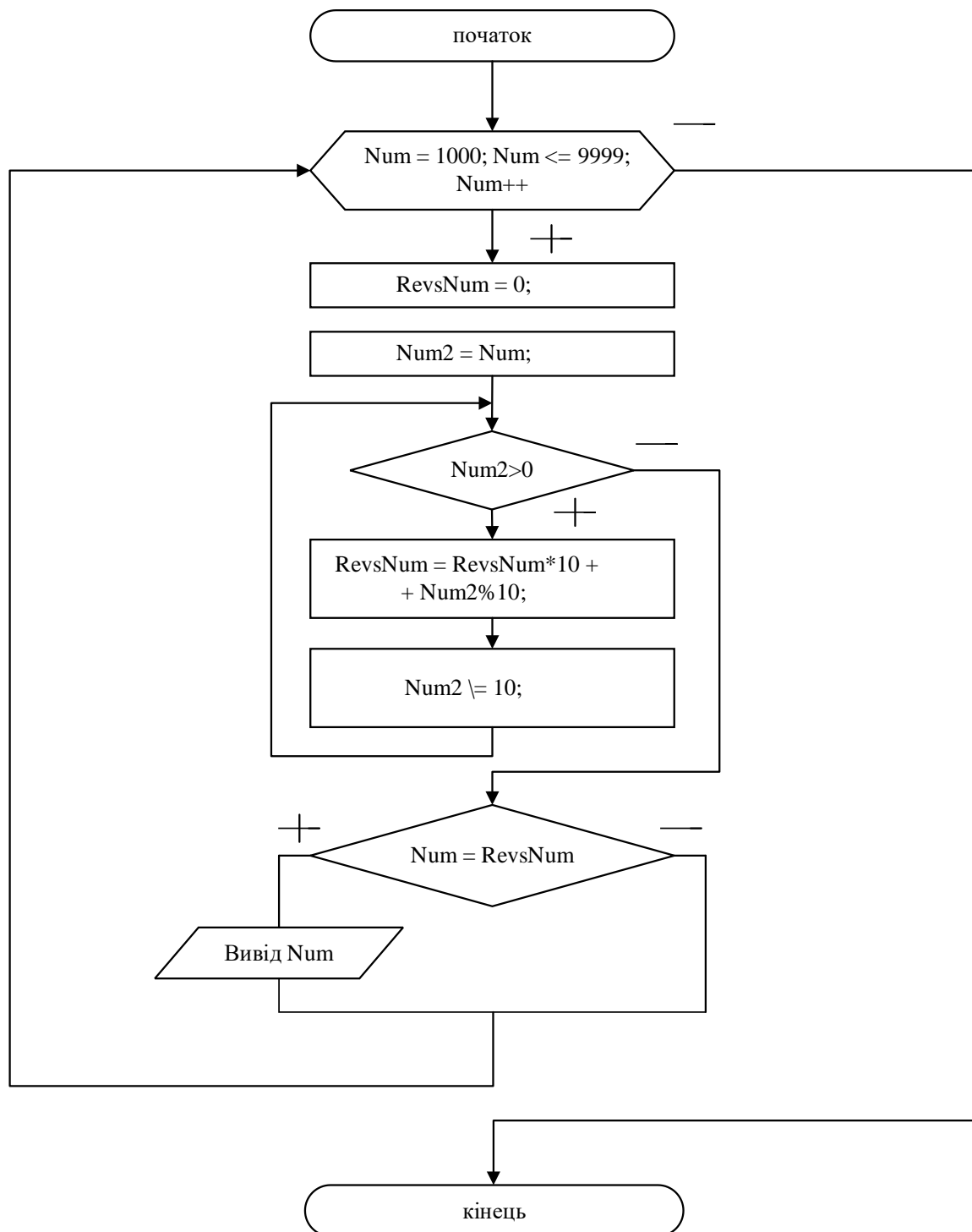
Крок 5



Крок 6



Крок 7



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.