

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 8

Виконав студент ПІ-12 Волков Вадим Всеволодович
(шифр, прізвище, ім'я, по батькові)

Перевірив _____
(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.

Задача 8. Цифровий корінь натурального числа – це одноцифрове значення, яке отримується із цифр числа шляхом ітераційного процесу знаходження спочатку суми цифр даного числа, а потім, якщо потрібно, суми цифр значень, отриманих на попередній ітерації знаходження відповідних сум (якщо значення суми не є цифрою). Цей процес триває до тих пір, поки не буде отримано однорозрядне число. Наприклад, цифровим коренем числа 65536 є 7, так як $6+5+5+3+6=25$ і $2+5=7$. Знайти цифрові корені всіх простих чисел з інтервалу $[100, 200]$.

Побудова математичної моделі.

Змінна	Тип	Ім'я	Призначення
Ітеруєча змінна	Ціле	Iterator	Тимчасова змінна
Число з якого читають	Ціле	Number	Тимчасова змінна
Число з Number	Ціле	Digit	Тимчасова змінна
Сума цифр Number	Ціле	NewNumber	Тимчасова змінна

Задачею є проходження по числам від 100 до 200 і обчислення цифрового кореню натурального цих чисел. Для зовнішнього проходження потрібно використати арифметичний цикл. Всередині потрібно скопіювати значення з зовнішнього циклу у змінну Number і використовуючий внутрішній алгоритм, що рахує суму цифр Number і результат теж поміщує у Number, викликати його поки значення Number не стане однією цифрою за допомогою ітераційного циклу. Сам алгоритм з обчислення суми цифр складається з проходження по всім цифрам і додавання їх у тимчасову змінну NewNumber, значення якої у кінці копіюється у Number. Отримання цифр може бути реалізоване діями остачі від ділення на 10 та цілочисельним діленням на 10.

Розв'язання

Програмні специфікації запишемо у псевдокодi та графічній формi у вигляді блоксхеми.

Крок 1. Визначимо основні дії.

Крок 2. Деталізуємо дію проходження від 100 до 200 та обчислення суми чисел поки не залишиться 1 цифра і виведення її.

Крок 3. Деталізуємо дію задання Number значення Iterator.

Крок 4. Деталізуємо дію починаючи з Number обчислення суми чисел поки не залишиться 1 цифра.

Крок 5. Деталізуємо дію обчислення суми чисел Number у змінну NewNumber.

Крок 6. Деталізуємо дію виділення потрібної цифри Digit з Number.

Крок 7. Деталізуємо дію додавання Digit у NewNumber.

Псевдокод

крок 1

початок

Починаючи з всіх чисел від 100 до 200 обчислювати суму чисел поки не залишиться 1 цифра і виведення її

кінець

крок 2

початок

повторювати у Iterator від 100 до 200 з шагом 1:

Копіювати значення Iterator у змінну Number

Починаючи з Number обчислювати суму чисел поки не залишиться 1 цифра

Виведення N

все повторювати

кінець

крок 3

початок

повторювати у Iterator від 100 до 200 з шагом 1:

Number := Iterator

Починаючи з Number обчислювати суму чисел поки не залишиться 1 цифра

Виведення N

все повторювати

кінець

крок 4

початок

повторювати у Iterator від 100 до 200 з шагом 1:

Number := Iterator

поки Number >= 10:

виконувати

Обчислити суму чисел Number у змінну NewNumber

Number := NewNumber

все поки

Виведення N

все повторювати

кінець

крок 5

початок

повторювати у Iterator від 100 до 200 з шагом 1:

Number := Iterator

поки Number >= 10:

виконувати

NewNumber := 0

поки Number >= 10:

виконувати

Виділення потрібної цифри Digit з Number

Додавання Digit у NewNumber

все поки

NewNumber := NewNumber + Number

Number := NewNumber

все поки

Виведення N

все повторювати

кінець

крок 6

початок

повторювати у Iterator від 100 до 200 з шагом 1:

Number := Iterator

поки Number >= 10:

виконувати

NewNumber := 0

поки Number >= 10:

виконувати

Digit := Number % 10

Number := Number / 10

Додавання Digit у NewNumber

все поки

NewNumber := NewNumber + Number

Number := NewNumber

все поки

Виведення N

все повторювати

кінець

крок 7

початок

повторювати у Iterator від 100 до 200 з шагом 1:

Number := Iterator

поки Number >= 10:

виконувати

NewNumber := 0

поки Number >= 10:

виконувати

Digit := Number % 10

Number := Number / 10

NewNumber := NewNumber + Digit

все поки

NewNumber := NewNumber + Number

Number := NewNumber

все поки

Виведення N

все повторювати

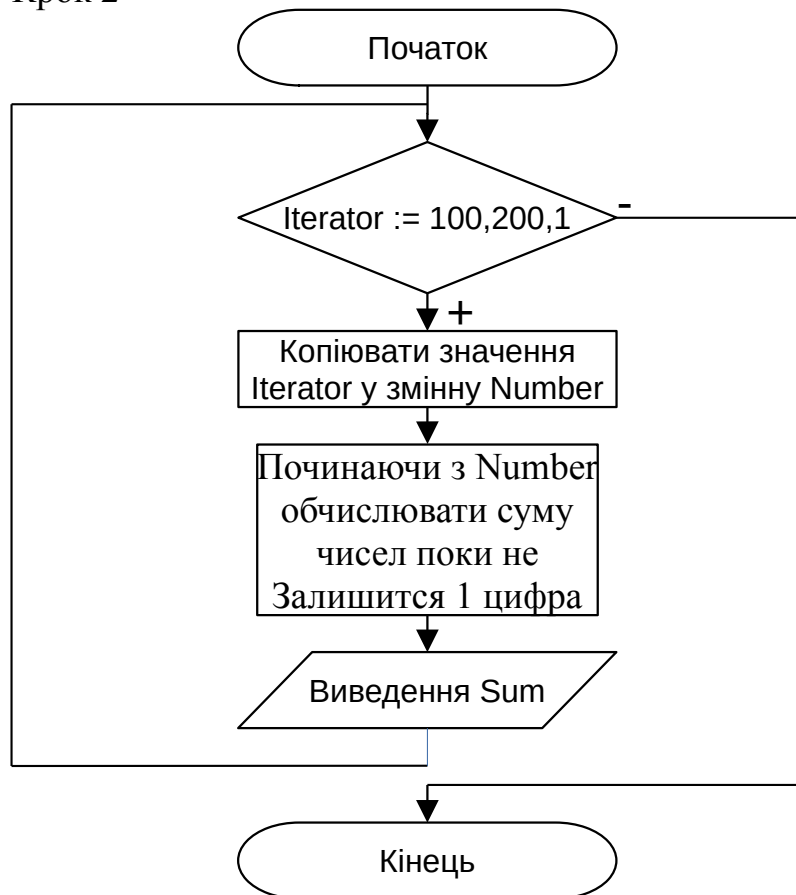
кінець

Блок схема

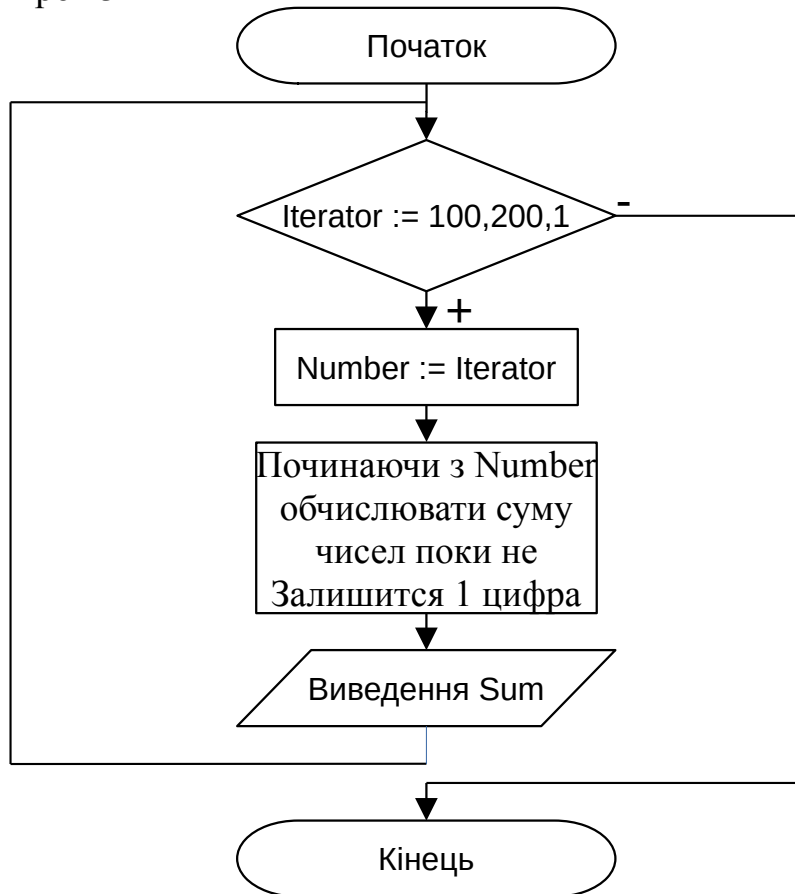
Крок 1



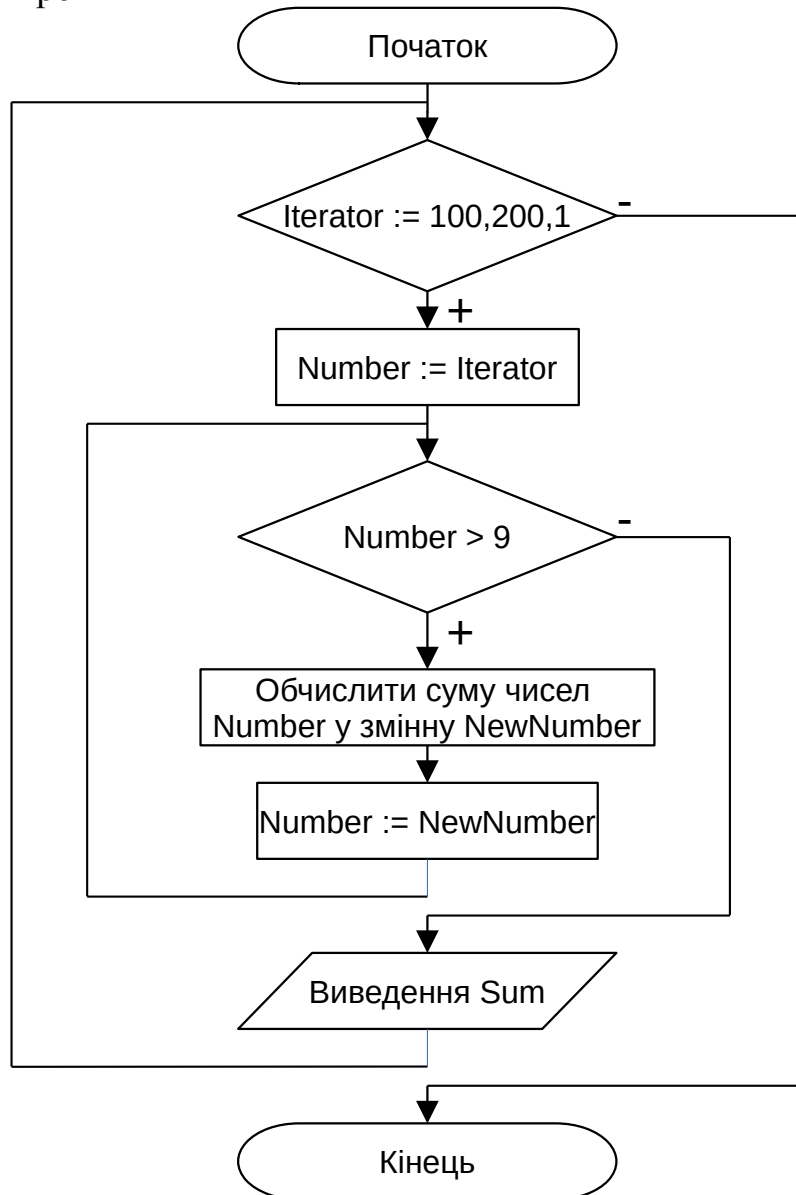
Крок 2



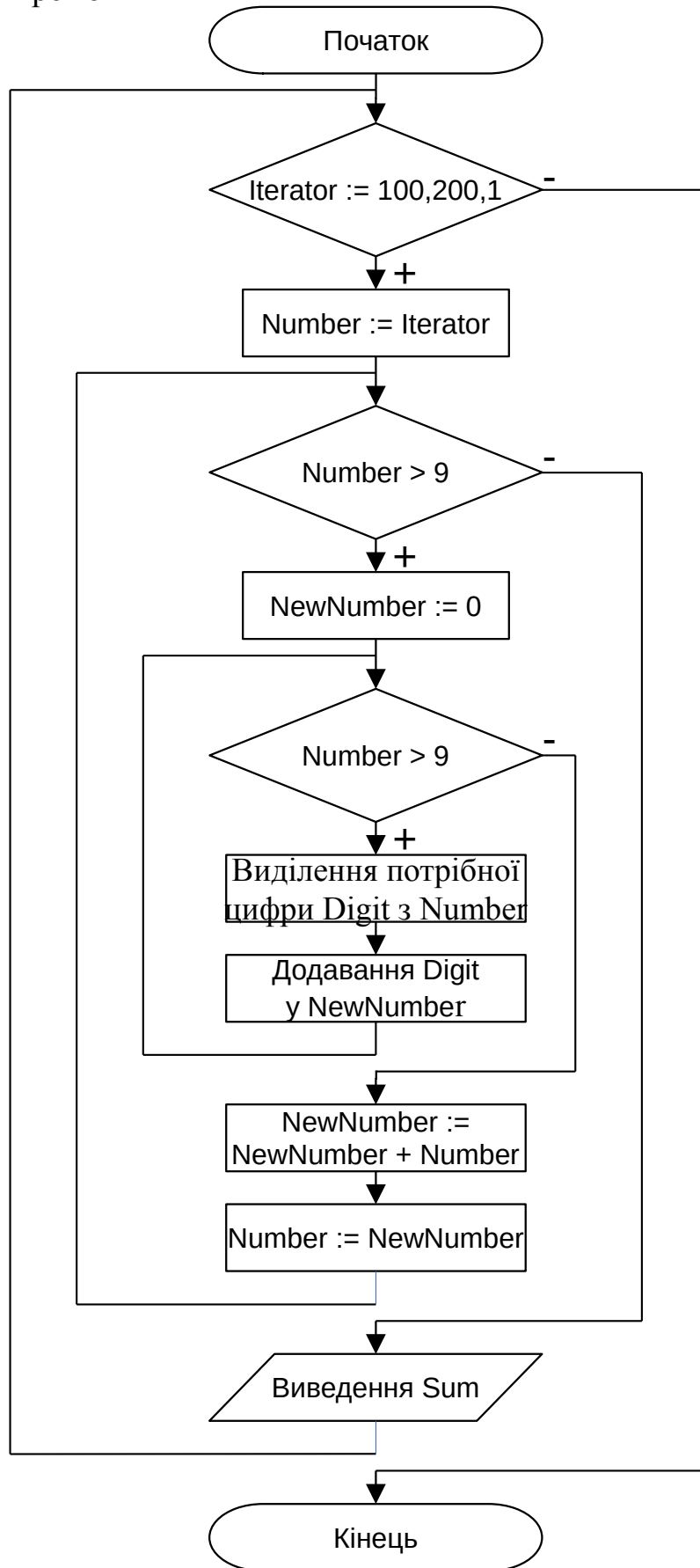
Крок 3



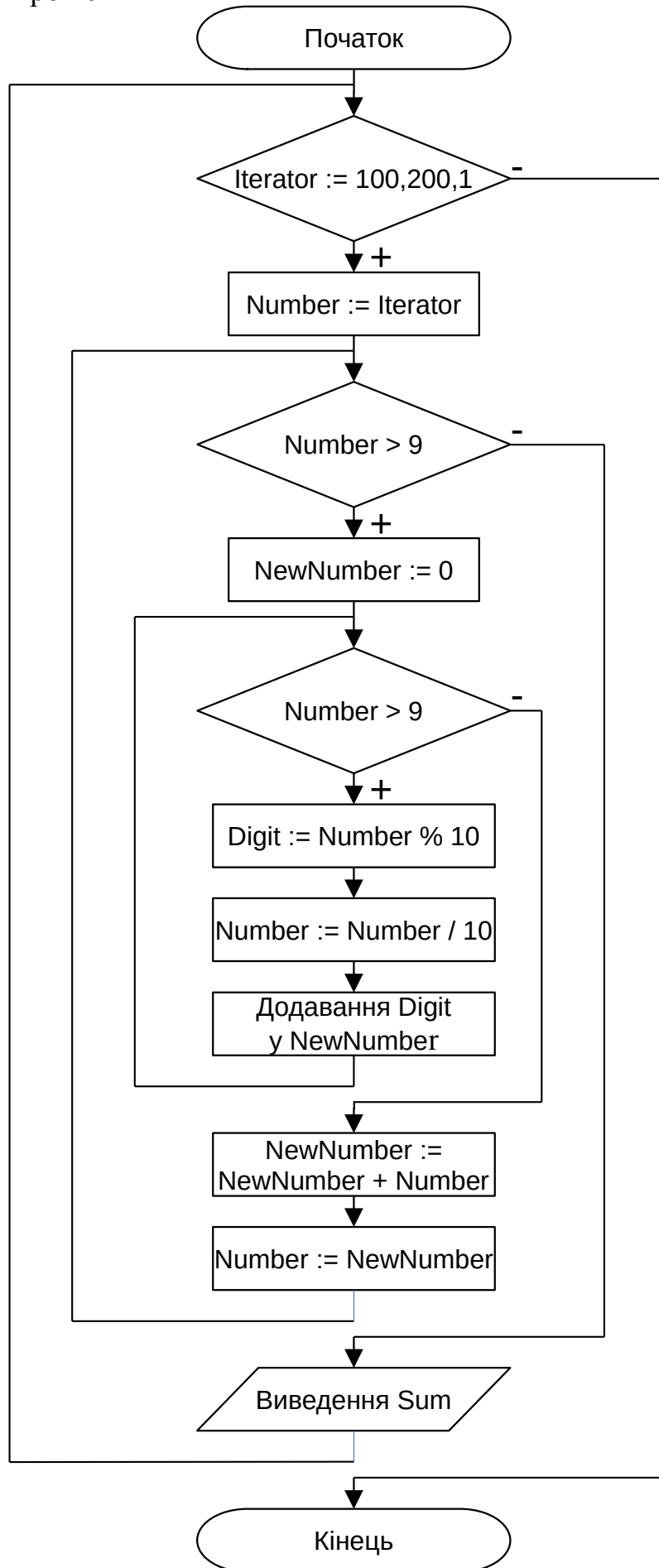
Крок 4



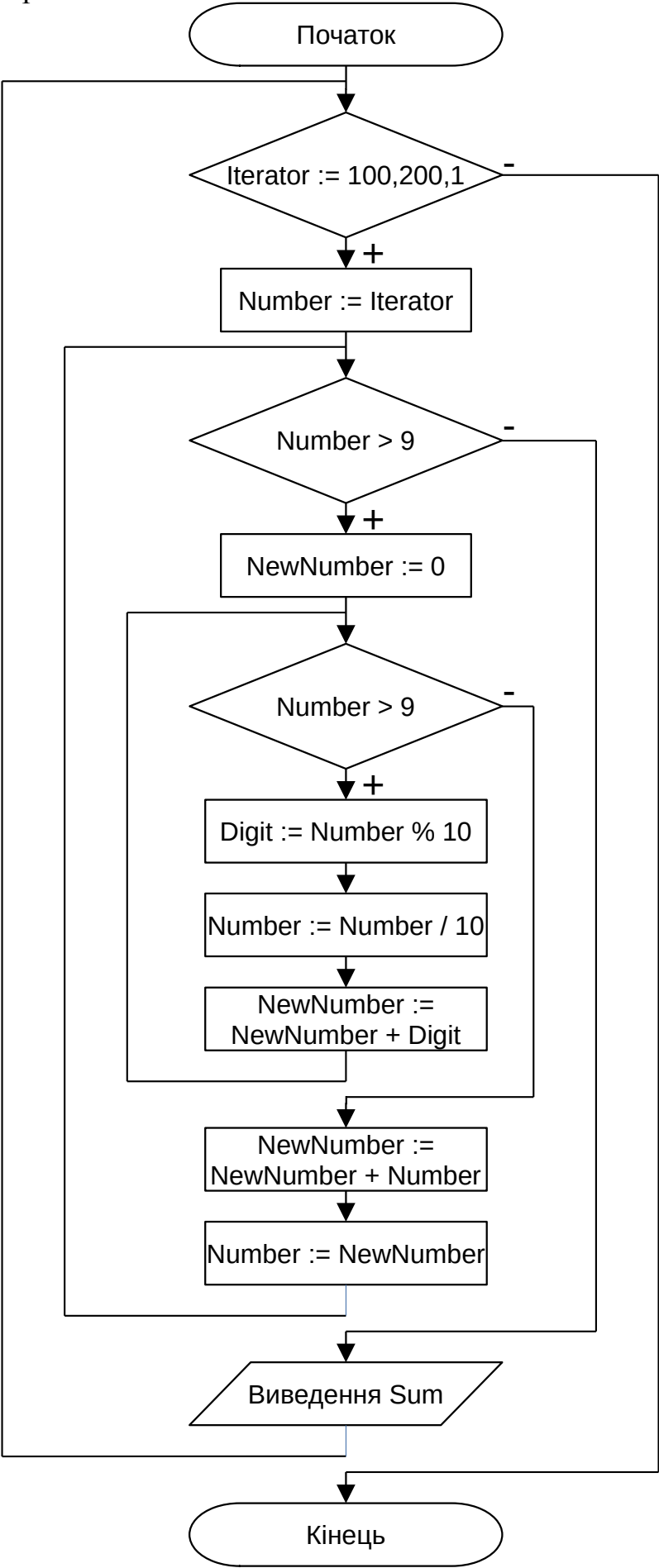
Крок 5



Крок 6



Крок 7



Перевіримо правильність алгоритма на довільних конкретних значеннях початкових даних:

Бл ок	Дія	Дія	Дія
	Початок	Початок	Початок

	Для Iterator = 130	Для Iterator = 621	Для Iterator = 975
1	Number = 130	Number = 621	Number = 975
2	$130 > 9$ - так	$621 > 9$ - так	$975 > 9$ - так
3	NewNumber = 0	NewNumber = 0	NewNumber = 0
4	$130 > 9$ - так	$621 > 9$ - так	$975 > 9$ - так
5	$\text{Digit} = 130 \% 10 = 0$	$\text{Digit} = 621 \% 10 = 1$	$\text{Digit} = 975 \% 10 = 5$
6	$\text{Number} = 130 / 10 = 13$	$\text{Number} = 621 / 10 = 62$	$\text{Number} = 975 / 10 = 97$
7	$\text{NewNumber} = 0 + 0 = 0$	$\text{NewNumber} = 0 + 1 = 1$	$\text{NewNumber} = 0 + 5 = 5$
8	$13 > 9$ - так	$62 > 9$ - так	$97 > 9$ - так
9	$\text{Digit} = 13 \% 10 = 3$	$\text{Digit} = 62 \% 10 = 2$	$\text{Digit} = 97 \% 10 = 7$
10	$\text{Number} = 13 / 10 = 1$	$\text{Number} = 62 / 10 = 6$	$\text{Number} = 97 / 10 = 9$
11	$\text{NewNumber} = 0 + 3 = 3$	$\text{NewNumber} = 1 + 2 = 3$	$\text{NewNumber} = 5 + 7 = 12$
12	$1 > 9$ - ні	$6 > 9$ - ні	$9 > 9$ - ні
13	$\text{NewNumber} = 3 + 1 = 4$	$\text{NewNumber} = 3 + 6 = 9$	$\text{NewNumber} = 12 + 9 = 21$
14	Number = 4	Number = 9	Number = 21
15	$4 > 9$ - ні	$9 > 9$ - ні	$21 > 9$ - так
16	Вивід: 4	Вивід: 9	NewNumber = 0
17	Наступна ітерація	Наступна ітерація	$21 > 9$ - так
18			$\text{Digit} = 21 \% 10 = 1$
19			$\text{Number} = 21 / 10 = 2$
20			$\text{NewNumber} = 0 + 1 = 1$
21			$2 > 9$ - ні
22			$\text{NewNumber} = 1 + 2 = 3$
23			$3 > 9$ - ні
24			Вивід: 3
			Наступна ітерація

	Кінець	Кінець	Кінець

Висновок:

На основі цього алгоритму по знаходженню цифрового коріня натурального числа було досліджено особливості роботи складних циклів та набуто практичних навичок їх використання під час складання програмних специфікацій.