

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 7

Виконав студент ІП-13 Гогіберідзе Торніке Лашаєвич

(шифр, прізвище, ім'я, по батькові)

Перевірила

(прізвище, ім'я, по батькові)

Київ 2021

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 7

7. Натуральне число називається паліндромом, якщо його запис читається однаково з початку та з кінця (наприклад, 4884, 575, 9). Знайти всі числа-паліндроми, що не перевищують 1000.

Постановка задачі

Використовуємо складний циклічний алгоритм: зовнішній арифметичний цикл, внутрішній цикл з передумою і внутрішній арифметичний цикл (між собою останні два цикла – послідовні.) Використовуючи зовнішній арифметичний цикл, перебираємо всі натуральні числа, що не перевищують 1000. Використовуючи внутрішній цикл з передумою, знаходимо кількість цифр в числі, яке ми розглядаємо. Використовуючи внутрішній арифметичний цикл, «перевертаємо» число та записуємо його в змінній opNum. Вкінці кожного повторення зовнішнього циклу порівнюємо початкове число з його «перевернутим» числом (використовуємо умовну форму оператора вибору). Якщо обидва числа одинакові – виводимо наше початкове число.

Результатом розв'язку є знаходження та виведення всіх чисел-паліндромів, що не перевищують 1000.

Побудова математичної моделі

Таблиця імен змінних

Змінна	Тип	Ім'я	Призначення
Верхня границя паліндромів, які потрібно знайти	Ціле	limit	Початкові дані
Лічильник в зовнішньому арифметичному циклі	Ціле	i	Проміжні дані
Лічильник в внутрішньому арифметичному циклі	Ціле	j	Проміжні дані
Число, яке ми наразі розглядаємо	Натуральне	curNum	Проміжні дані/ Результат
«Перевернуте» число	Ціле	opNum	Проміжні дані
Кількість цифр в числі	Ціле	digits	Проміжні дані

Перед початком циклу ми встановлюємо значення ліміту змінної на 1000 (верхня межа паліндрому, яку потрібно знайти). Усередині зовнішнього циклу встановіть значення змінної curNum = i (число, яке ми зараз

перевіряємо). Далі ініціалізуємо змінні opNum = 0 (змінна «взаємна») і digits = 0 (кількість цифр у числі). Використовуйте внутрішній цикл з передумовою, щоб знайти кількість цифр у числі: розділіть наше число на 10 цілих чисел і додайте 1 до змінної числа. Для цілого поділу ми використовуємо div ($curNum = curNum \text{ div } 10$). В кінці ітераційного циклу ми повертаємо значення $curNum = i$ і використовуємо лічильник j, щоб увійти у внутрішній арифметичний цикл, який побудує число, «протилежне» нашему. Від $j = \text{число}-1$ до 0 програма помножить кожну цифру нашого числа на 10 в степені j і додасть її до змінної opNum; отже, кожне число буде записане з кінця. Додайте перше число в кінці наступним чином: $opNum = opNum + (curNum \text{ mod } 10) * (10^j)$, де mod — це функція для знаходження залишку від ділення, а pow — функція для обчислення ступеня (u нашему випадку ступінь 10). Додайте ціле число і поділіть на $curNum = curNum \text{ div } 10$ і повторіть до кінця внутрішнього арифметичного циклу. Потім ми порівнюємо opNum з curNum, і якщо вони однакові, виводимо curNum (знаходимо паліндром).

Розв'язання

Програмні специфікації запишемо у псевдокоді та графічній формі у вигляді блок-схеми.

Крок 1. Визначимо основні дії.

Крок 2. Ініціалізація змінної limit

Крок 3. Визначення зовнішнього арифметичного циклу

Крок 4. Ініціалізація змінних curNum, opNum і digits

Крок 5. Визначення внутрішнього ітераційного циклу

Крок 6. Задання значення змінної curNum

Крок 7. Визначення внутрішнього арифметичного циклу

Крок 8. Визначення оператору розгалуження та виведення паліндрому

Псевдокод

Крок 1

початок

Ініціалізація змінної limit

Визначення зовнішнього
арифметичного циклу

кінець

Крок 2

початок

limit : = 1000

Визначення зовнішнього
арифметичного циклу

кінець

Крок 3

початок

limit : = 1000

повторити

для i від 1 до limit

Ініціалізація змінних curNum, opNum і digits

Визначення внутрішнього ітераційного циклу

Задання значення змінної curNum

Визначення внутрішнього арифметичного циклу

Визначення оператору розгалуження та виведення паліндрому

все повторити

кінець

Крок 4 початок

limit : = 1000 **повторити**

для i від 1 до limit

curNum : = i

opNum := 0
digits := 0

Визначення внутрішнього ітераційного циклу

Задання значення змінної curNum

Визначення внутрішнього арифметичного циклу

Визначення оператору розгалуження та виведення паліндрому

все повторити

кінець

Krok 5 початок

limit := 1000 **повторити**
для i від 1 до limit
curNum := i
opNum := 0
digits := 0

повторити

поки (curNum != 0)

digits := digits + 1

curNum := curNum div 10

все повторити

Задання значення змінної curNum

Визначення внутрішнього арифметичного циклу

Визначення оператору розгалуження та виведення паліндрому

все повторити

кінець

Krok 6 початок

limit := 1000 **повторити**
для i від 1 до limit
curNum := i

opNum := 0

digits := 0

повторити

поки (curNum != 0)

 digits := digits + 1

 curNum := curNum div 10

все повторити

 curNum := i

Визначення внутрішнього арифметичного циклу

Визначення оператору розгалуження та виведення паліндрому

все повторити

кінець

Krok 7 початок

limit := 1000 **повторити**

 для i від 1 до limit

 curNum := i

 opNum := 0

 digits := 0

повторити

поки (curNum != 0)

 digits := digits + 1

 curNum := curNum div 10

все повторити

 curNum := i

повторити

 для j від (digits -1) до 0, з кроком -1

 opNum = opNum + (curNum mod 10) * (10 pow j)

) curNum := curNum div 10 **все повторити**

Визначення оператору розгалуження та виведення паліндрому

все повторити

кінець

Krok 8 початок

```
limit := 1000 повторити
    для i від 1 до limit
        curNum := i
        opNum := 0
        digits := 0
        повторити
            поки (curNum != 0)
                digits := digits + 1
                curNum := curNum div 10
            все повторити
            curNum := i
        повторити
            для j від (digits -1) до 0, з кроком -1
                opNum = opNum + (curNum mod 10) * (10 pow j )
                curNum := curNum div 10
            все повторити
            якщо (i == opNum)
                то
                    виведення i
                все якщо
            все повторити
```

кінець

Блок-схема

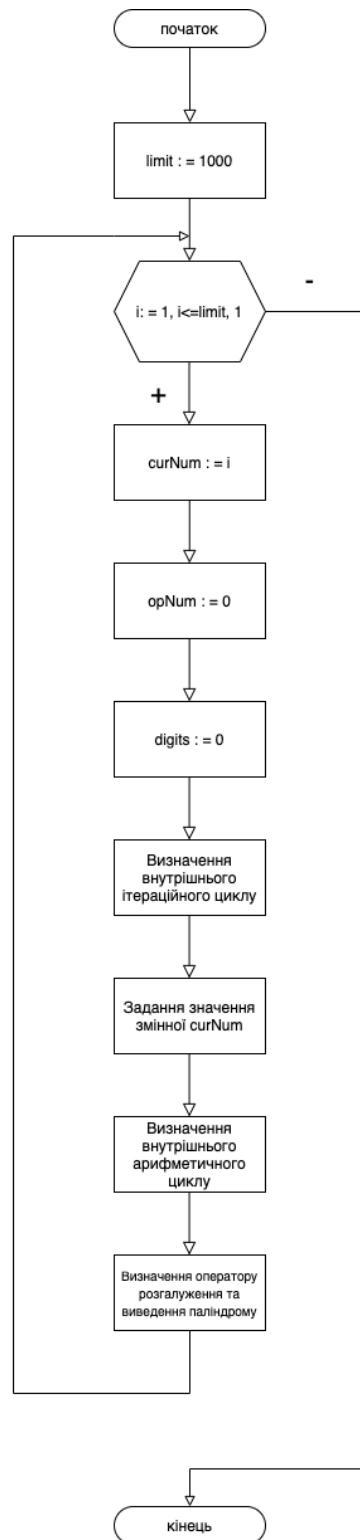
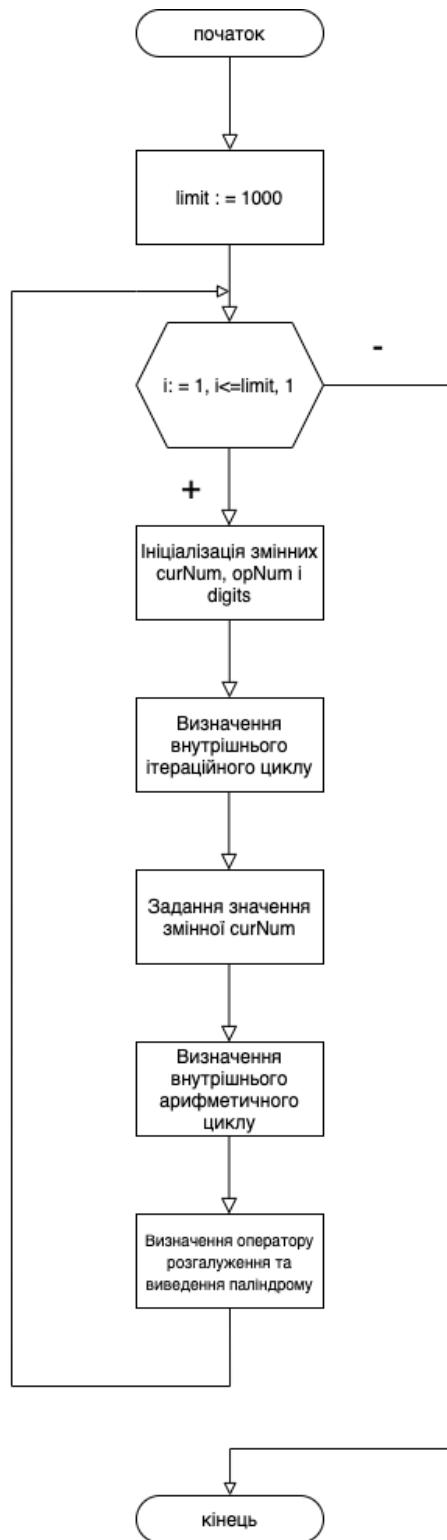
Krok 1.

Krok 2.



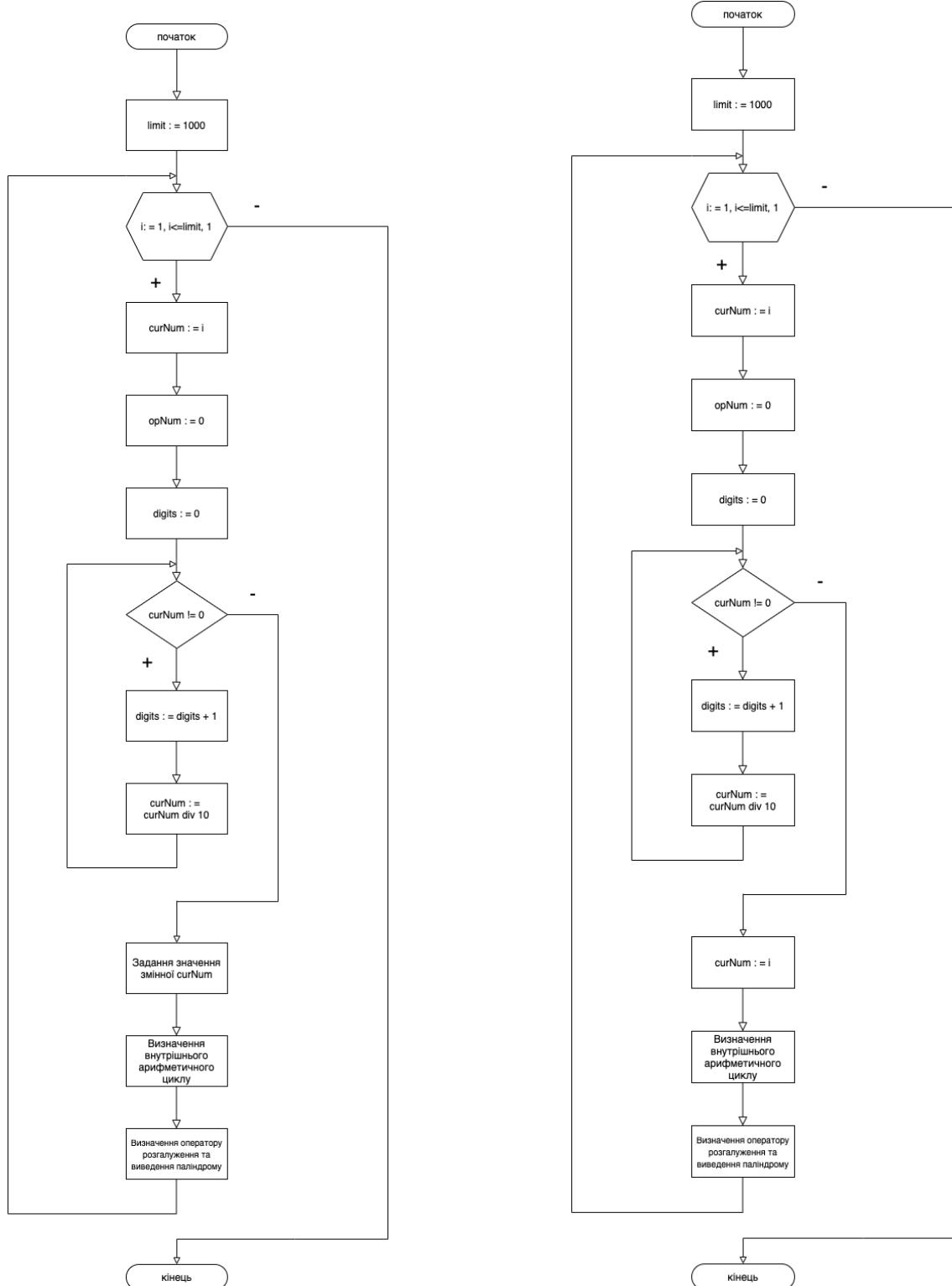
Крок 3.

Крок 4.

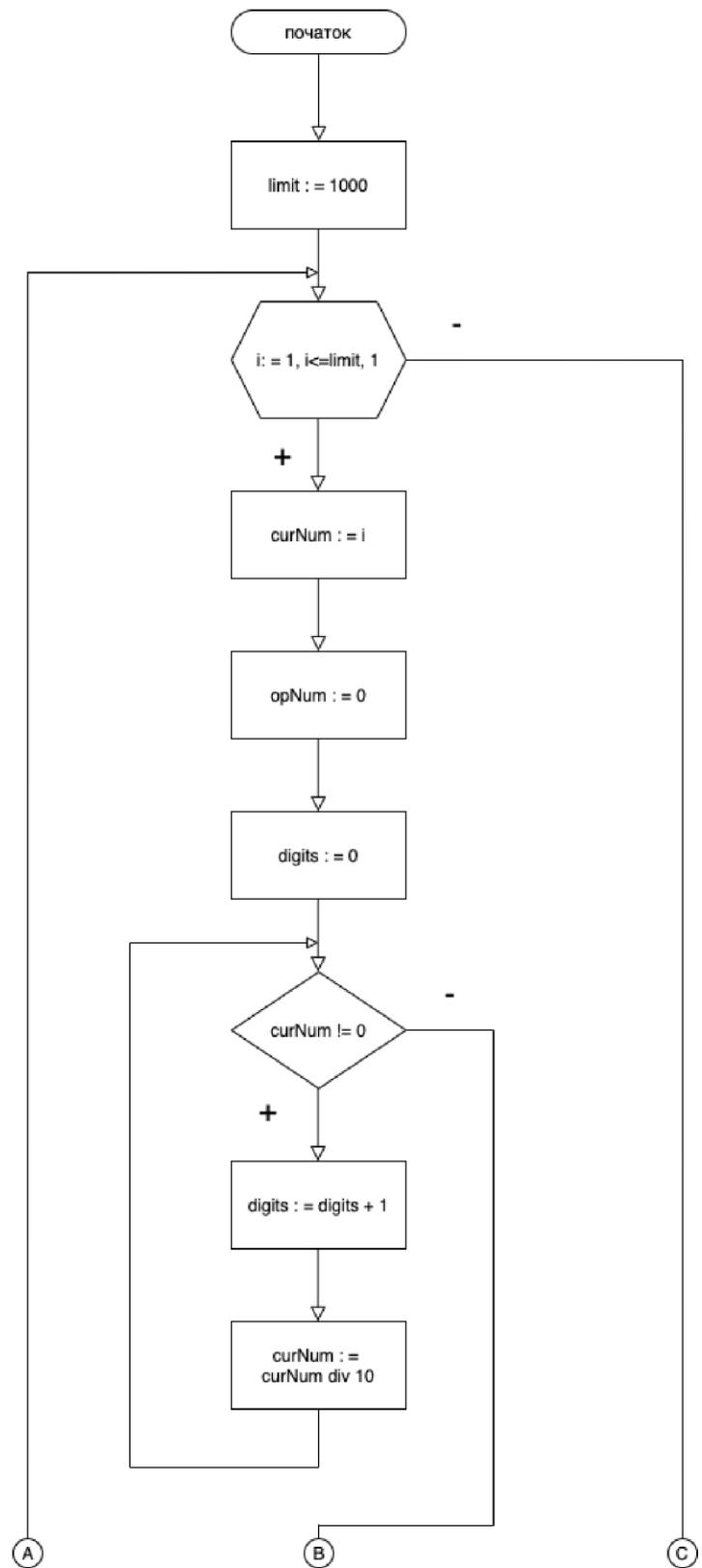


Крок 5.

Крок 6.



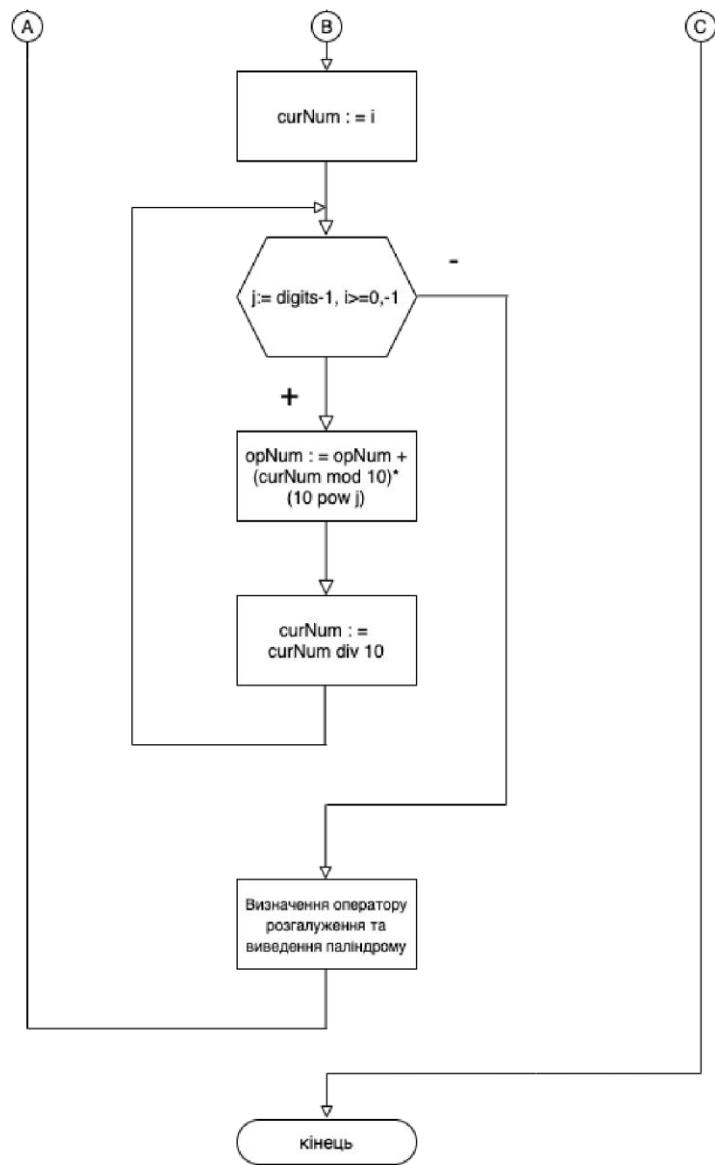
Крок 7.



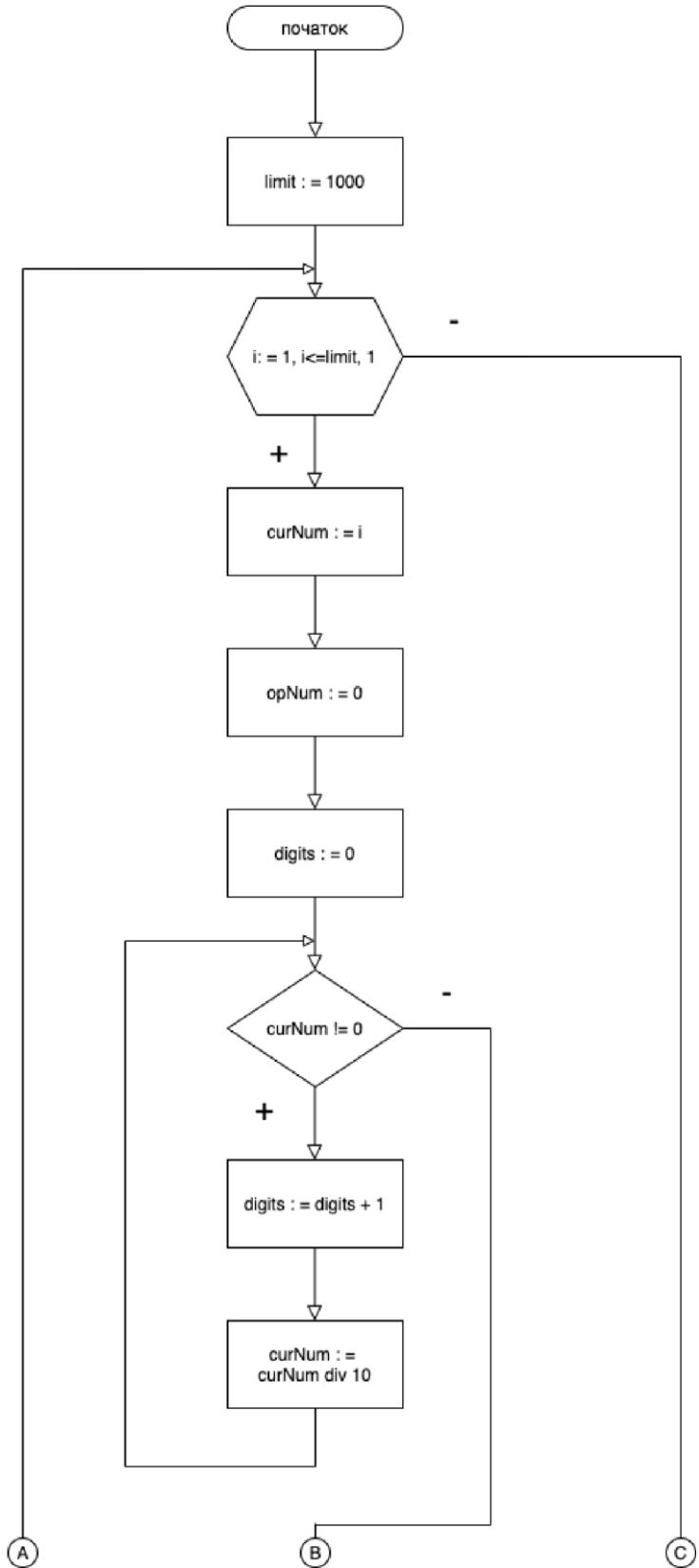
A

B

C



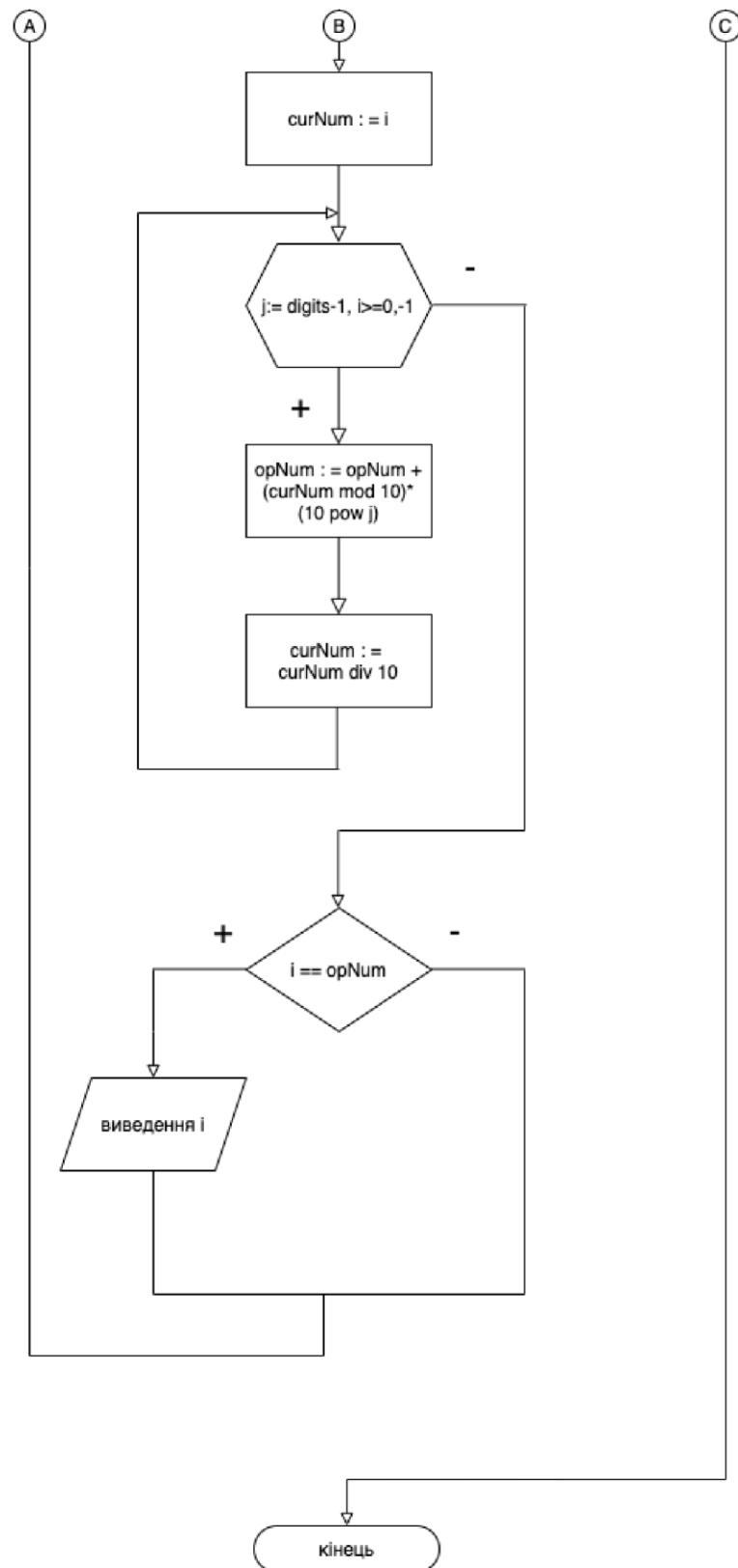
Krok 8.



(A)

(B)

(C)



Тестування

Блок	Дія (зовнішній цикл)	Дія (внутрішні цикли)
	Початок	
1	limit = 1000	
2	i = 1, $1 \leq 1000 \rightarrow \text{true}$	
3	curNum = 1, opNum = 0, digits = 0	
4		$1 \neq 0 \rightarrow \text{true}$
5		$\text{digits} = 0 + 1 = 1$
6		$\text{curNum} = 1 \text{ div } 10 = 0$
7		$0 \neq 0 \rightarrow \text{false}$
8	curNum = 1	
9		$j = 0, 0 \geq 0 \rightarrow \text{true}$
10		$\text{opNum} = 0 + 1 * (10 \text{ pow } 0) = 1$
11		$\text{curNum} = 1 \text{ div } 10 = 0$
12		$j = -1, -1 \geq 0 \rightarrow \text{false}$
13	$1 == 1 \rightarrow \text{true}$	
14	Виведення 1	
15	i = 2, $1 \leq 1000 \rightarrow \text{true}$	
	...	
16	i = 1000, $1000 \leq 1000 \rightarrow \text{true}$	
17	curNum = 1000, opNum = 0, digits = 0	

18		$1000 \neq 0 \rightarrow \text{true}$
19		$\text{digits} = 0 + 1 = 1$

21		$\text{curNum} = 1000 \text{ div } 10 = 100$
22		$100 \neq 0 \rightarrow \text{true}$
23		$\text{digits} = 1 + 1 = 2$
24		$\text{curNum} = 100 \text{ div } 10 = 10$
25		$10 \neq 0 \rightarrow \text{true}$
26		$\text{digits} = 2 + 1 = 3$
27		$\text{curNum} = 10 \text{ div } 10 = 1$
28		$1 \neq 0 \rightarrow \text{true}$
29		$\text{digits} = 3 + 1 = 4$
30		$\text{curNum} = 1 \text{ div } 10 = 0$
31		$0 \neq 0 \rightarrow \text{false}$
32	$\text{curNum} = 1000$	
33		$j = 3, 3 \geq 0 \rightarrow \text{true}$
34		$\text{opNum} = 0 + 0 * (10 \text{ pow } 3) = 0$
35		$\text{curNum} = 1000 \text{ div } 10 = 100$
36		$j = 2, 2 \geq 0 \rightarrow \text{true}$
37		$\text{opNum} = 0 + 0 * (10 \text{ pow } 2) = 0$
38		$\text{curNum} = 100 \text{ div } 10 = 10$
39		$j = 1, 1 \geq 0 \rightarrow \text{true}$
40		$\text{opNum} = 0 + 0 * (10 \text{ pow } 1) = 0$

41		$\text{curNum} = 10 \text{ div } 10 = 1$
42		$j = 0, 0 \geq 0 \rightarrow \text{true}$
43		$\text{opNum} = 0 + 1 * (10 \text{ pow } 0) = 1$
44		$\text{curNum} = 1 \text{ div } 10 = 0$
45		$j = -1, -1 \geq 0 \rightarrow \text{false}$
46	$1000 == 1 \rightarrow \text{false}$	
	Кінець	

Висновки

Протягом п`ятогї лабораторної роботи я дослідив особливості роботи складних циклів та набув практичних навичок їх використання під час складання програмних специфікацій. В результат ми отримали алгоритм, що використовує складені циклі та знаходить числа-паліндроми, які не більше 1000.