

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 ІІІ-15, Гуменюк Олександр Володимирович

(шифр, прізвище, ім'я, по батькові)

Перевірила Вєчерковська Анастасія Сергіївна

(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 цикли – 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 однакові – виводимо наше початкове число.

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

Побудова математичної моделі

Таблиця імен змінних

<i>Змінна</i>	<i>Тип</i>	<i>Ім'я</i>	<i>Призначення</i>
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 циклів задаємо значення змінної $limit = 1000$ (верхня границя паліндромів, які потрібно знайти) Далі йде зовнішній арифметичний цикл з лічильником i , за допомогою якого ми будемо перебирати кожне

число від 1 до 1000 та перевіряти чи це число є паліндромом. У середині зовнішнього циклу задаємо значення змінної $curNum = i$ (число, яке ми наразі перевіряємо). Далі ініціалізуємо змінні $opNum = 0$ (змінна «перевернутого числа») та $digits = 0$ (кількість цифр в числі). Використовуючи внутрішній цикл з передумою, знаходимо кількість цифр в числі: цілочисельно ділимо наше число на 10 та додаємо 1 до змінної $digits$. Для цілочисельного ділення використовуємо div ($curNum = curNum \div 10$). Після закінчення ітераційного циклу повертаємо значення $curNum = i$ та заходимо в внутрішній арифметичний цикл з лічильником j , який збудує «перевернуте» число до нашого. Починаючи з $j = digits - 1$ і до 0, програма буде додавати кожен цифру нашого числа помножену на 10 до j -ого степеню до змінної $opNum$; таким чином кожна цифра буде записуватися з кінця. Додаємо першу цифру з кінця таким чином: $opNum = opNum + ((curNum \bmod 10) * (10 \text{ pow } j))$, де \bmod – функція для знаходження остачі від ділення, а pow – функція для обчислення степені (у нашому випадку степені 10). Після додавання цілочисельно ділимо $curNum = curNum \div 10$, і повторюємо до кінця внутрішнього арифметичного циклу. Після цього порівнюємо $opNum$ з $curNum$, і якщо вони однакові, виводимо $curNum$ (паліндром знайдено).

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 оператора 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 оператора 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 оператора розгалуження та виведення паліндрому

все повторити

кінець

Крок 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 оператора розгалуження та виведення паліндрому

все повторити
кінець

Крок 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 оператора розгалуження та виведення паліндрому

все повторити
кінець

Крок 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 оператора розгалуження та виведення паліндрому

все повторити

кінець

Крок 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)

$\text{curNum} := \text{curNum} \text{ div } 10$

все повторити

якщо ($i == \text{opNum}$)

то

виведення i

все якщо

все повторити

кінець

Блок-схема

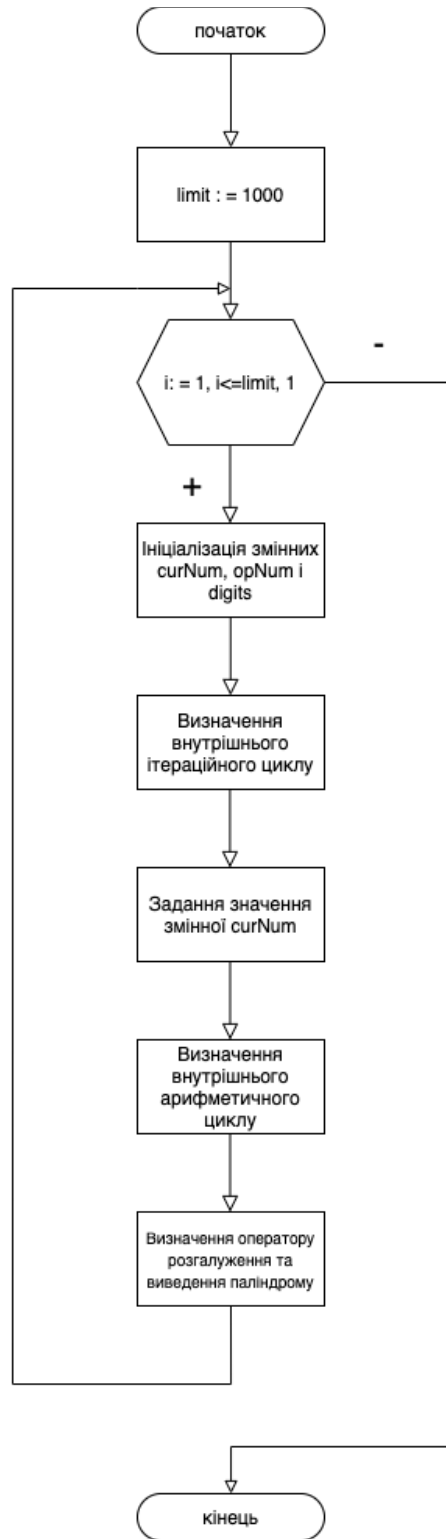
Крок 1.



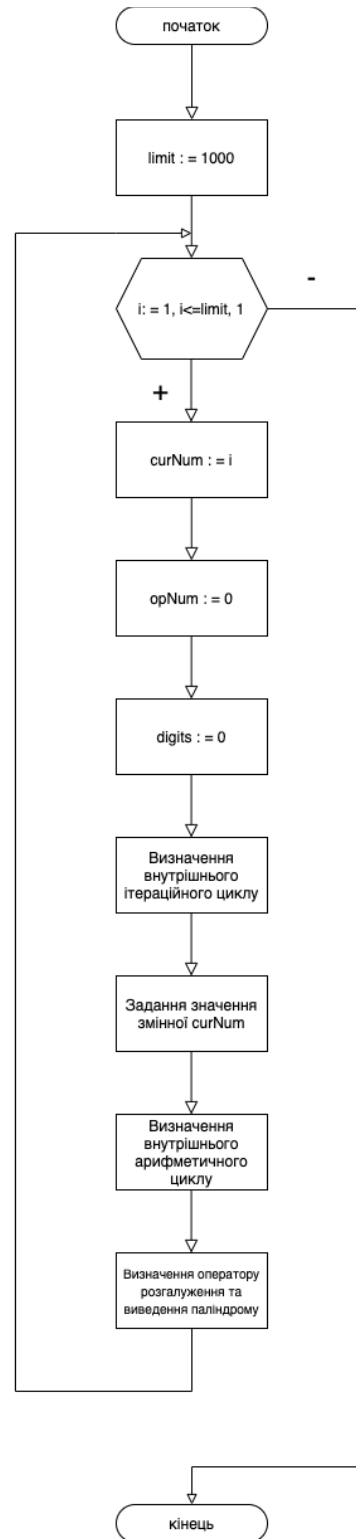
Крок 2.



Крок 3.



Крок 4.



```

graph TD
    Start([початок]) --> Limit[limit := 1000]
    Limit --> Init{ i := 1, k := limit, 1 }
    Init -- "+" --> CurNum[curNum := i]
    CurNum --> OpNum[opNum := 0]
    OpNum --> Digits[digits := 0]
    Digits --> Loop{curNum != 0}
    Loop -- "+" --> DigitsInc[digits := digits + 1]
    DigitsInc --> Div[curNum := curNum div 10]
    Div --> Loop
    Loop -- "-" --> AssignCurNum[Задання значення змінної curNum]
    AssignCurNum --> FindCycle[Визначення внутрішнього арифметичного циклу]
    FindCycle --> AssignOp[Визначення оператора розгортання та виведення паліндрому]
    AssignOp --> End([кінець])
    End --> Start

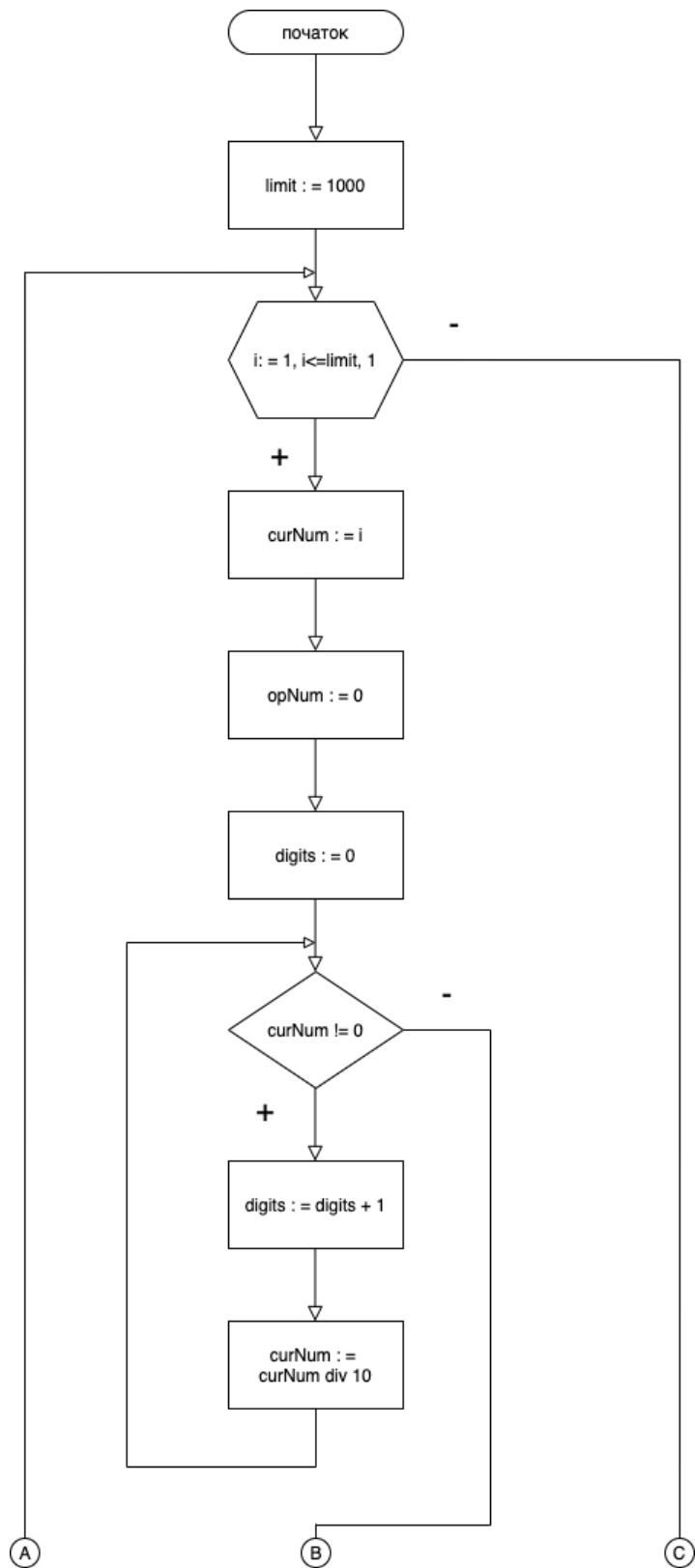
```

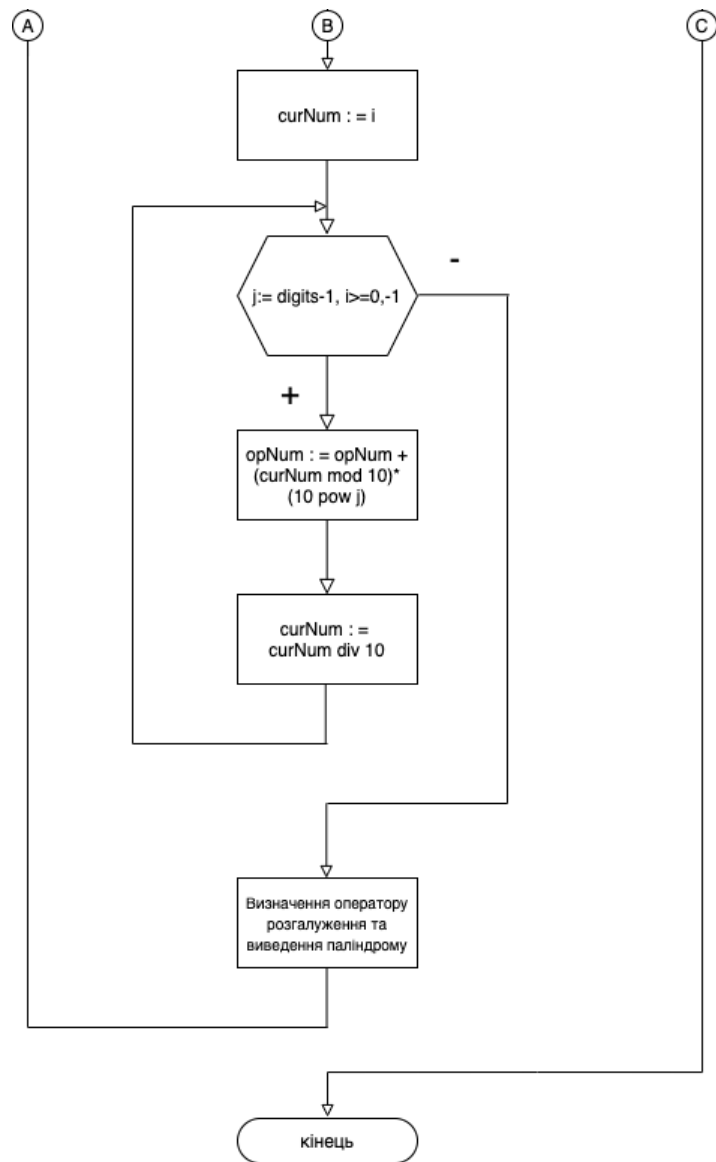
```

graph TD
    Start([початок]) --> Init[limit := 1000]
    Init --> LoopStart{ }
    LoopStart -- "+" --> AssignI[i := 1, ic := limit, 1]
    AssignI --> AssignCurNum[curNum := i]
    AssignCurNum --> AssignOpNum[opNum := 0]
    AssignOpNum --> AssignDigits[digits := 0]
    AssignDigits --> Decision{curNum != 0}
    Decision -- "+" --> IncrementDigits[digits := digits + 1]
    IncrementDigits --> DivCurNum[curNum := curNum div 10]
    DivCurNum --> Decision
    Decision -- "-" --> AssignCurNumI[curNum := i]
    AssignCurNumI --> CalcCycle[Визначення внутрішнього арифметичного циклу]
    CalcCycle --> CalcOp[Визначення оператора розгалуження та виведення паліндрому]
    CalcOp --> End([кінець])
    LoopStart -- "-" --> End
    DivCurNum --> LoopStart

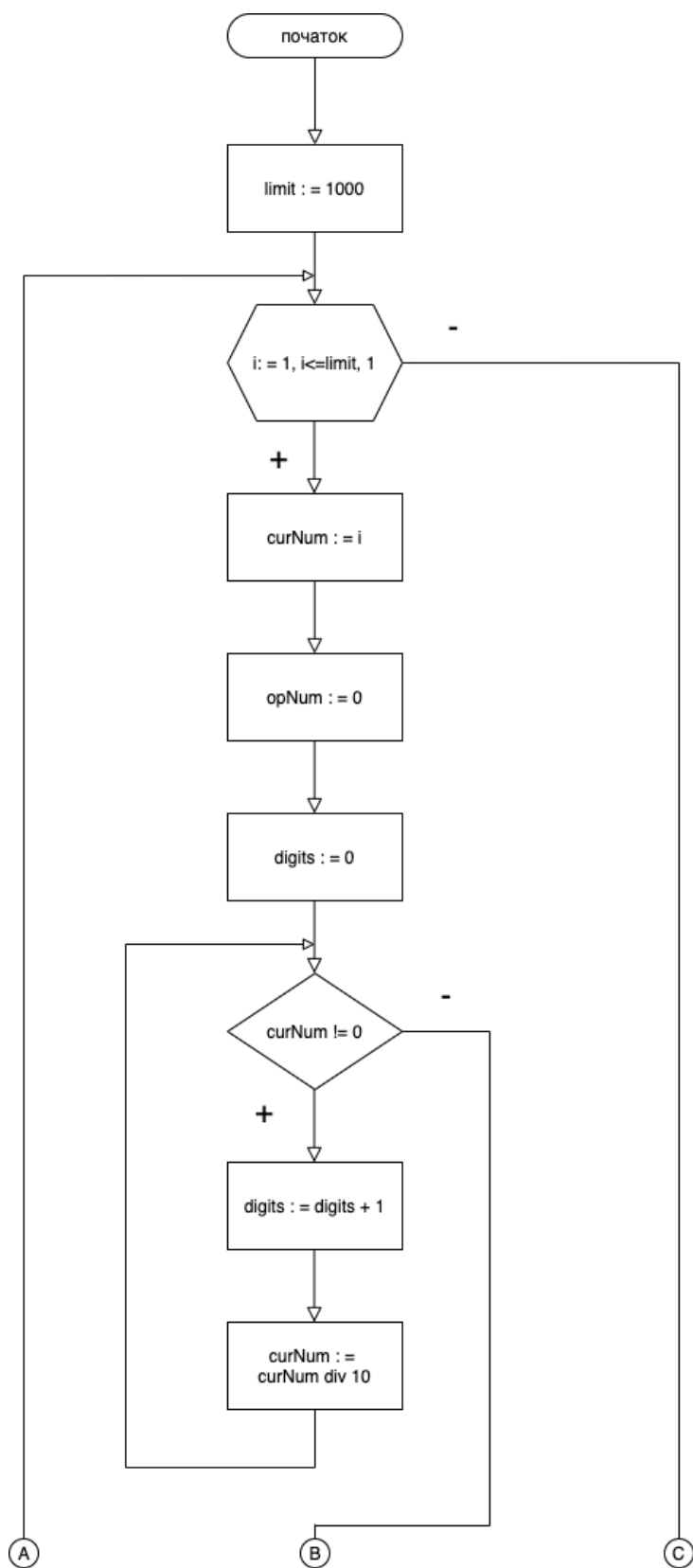
```

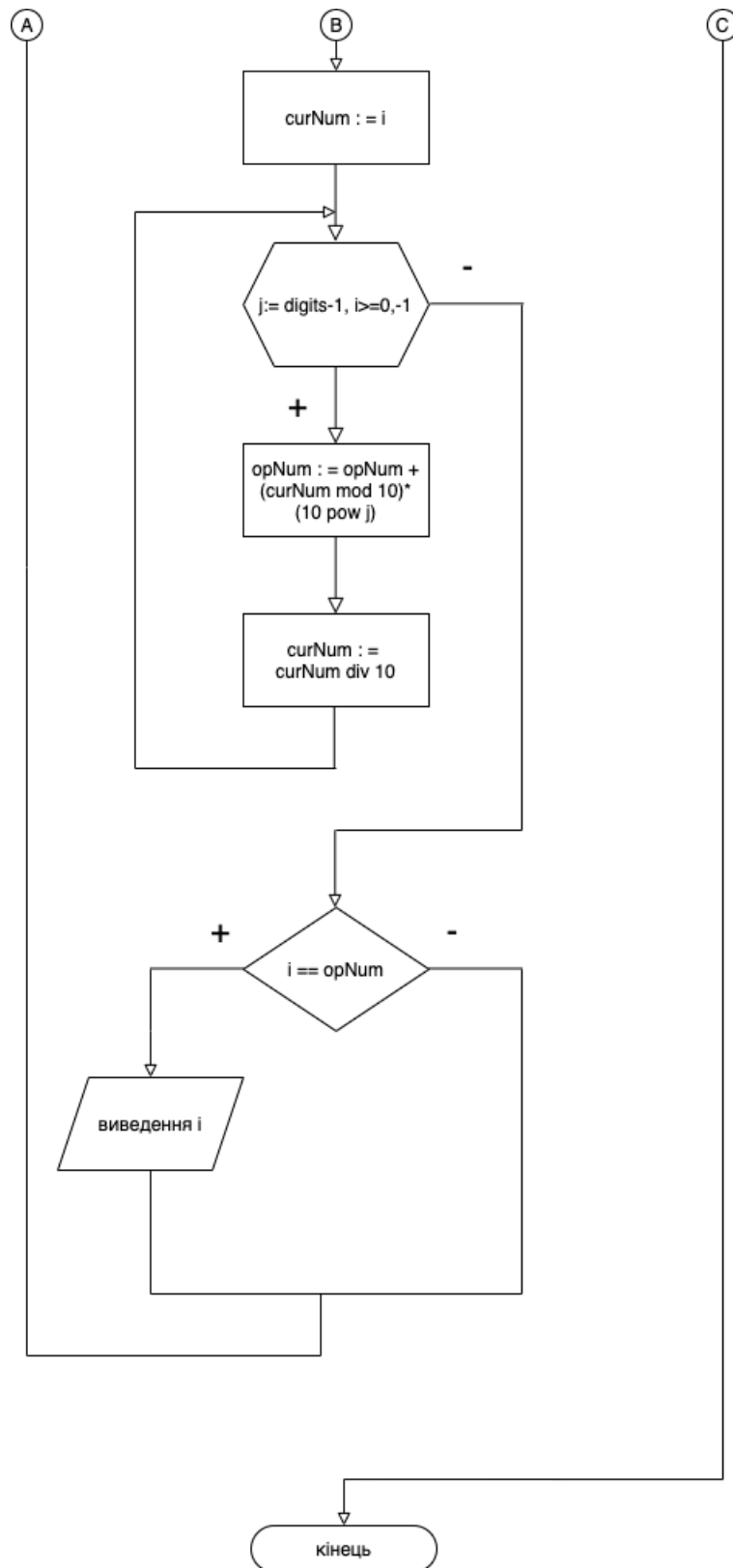
Крок 7.





Крок 8.





Тестування

Блок	Дія (зовнішній цикл)	Дія (внутрішні цикли)
	Початок	
1	limit = 1000	
2	i = 1, 1 <= 1000 → true	
3	curNum = 1, opNum = 0, digits = 0	
4		1 != 0 → true
5		digits = 0 + 1 = 1
6		curNum = 1 div 10 = 0
7		0 != 0 → false
8	curNum = 1	
9		j = 0, 0 >= 0 → true
10		opNum = 0 + 1*(10 pow 0) = 1
11		curNum = 1 div 10 = 0
12		j = -1, -1 >= 0 → false
13	1 == 1 → true	
14	Виведення 1	
15	i = 2, 1 <= 1000 → true	
	...	
16	i = 1000, 1000 <= 1000 → true	
17	curNum = 1000, opNum = 0, digits = 0	
18		1000 != 0 → true
19		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		$\text{opNum} = 0 + 0 \cdot (10 \text{ pow } 3) = 0$
35		$\text{curNum} = 1000 \text{ div } 10 = 100$
36		$j = 2, 2 \geq 0 \rightarrow \text{true}$
37		$\text{opNum} = 0 + 0 \cdot (10 \text{ pow } 2) = 0$
38		$\text{curNum} = 100 \text{ div } 10 = 10$
39		$j = 1, 1 \geq 0 \rightarrow \text{true}$
40		$\text{opNum} = 0 + 0 \cdot (10 \text{ pow } 1) = 0$
41		$\text{curNum} = 10 \text{ div } 10 = 1$
42		$j = 0, 0 \geq 0 \rightarrow \text{true}$
43		$\text{opNum} = 0 + 1 \cdot (10 \text{ pow } 0) = 1$
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'ятої 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.