

Міністерство освіти і науки України  
Київський національний технічний університет України «Київський політехнічний  
інститут імені Ігоря Сікорського»  
Факультет інформатики та обчислювальної техніки  
Кафедра інформатики та програмної інженерії

3bit

з лабораторної роботи №5 з дисципліни  
«Алгоритми та структури даних-1.  
Основи алгоритмізації»

«Дослідження складних циклічних алгоритмів»

Вариант 18

Виконав студент ІП-12 Кушнір Ганна Вікторівна  
(шифр, прізвище, ім'я, по батькові)

Перевірів \_\_\_\_\_  
(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 18

*Задача.* Ввести натуральні числа  $m$  та  $n$ . Як результат вивести усі натуральні числа, що менші за  $m$ , сума цифр яких дорівнює  $n$ .

1. *Постановка задачі.* Початковими даними є натуральні числа  $m$  та  $n$ , які вводяться користувачем з клавіатури. Результатом розв'язку є множина натуральних чисел, які задовольняють умову задачі.
2. *Побудова математичної моделі.* Складемо таблицю імен змінних.

Змінна	Тип	Ім'я	Призначення
Натуральне число $m$	Цілий, додатній (натуральне число)	$m$	Початкове дане
Натуральне число $n$	Цілий, додатній (натуральне число)	$n$	Початкове дане
Лічильник ітераційного циклу	Цілий	$i$	Лічильник циклу
Натуральне число $k$	Цілий, додатній (натуральне число)	$k$	Результат

Таким чином, математичне формулювання задачі зводиться до задання зовнішнього арифметичного циклу з параметром  $k$  ( $k$  приймає значення від 1 до  $m$ , не включаючи  $m$ , з кроком 1), цей цикл буде перевіряти кожне натуральне число ( $k$ ), яке менше за  $m$ . Цей цикл включає в собі наступні дії:

- 1) Присвоєння початкових значень змінним:  $\text{Sum}_k := 0, i := k$ .
- 2) Далі обчислення суми цифр числа  $k$  за допомогою вкладеного циклу з передумовою (поки  $i > 0$ ), у якому будуть повторюватися такі дії:  $\text{Sum}_k := \text{Sum}_k + i \% 10, i := i / 10$ .
- 3) І останнє: перевірка, чи сума цифр числа  $k$  дорівнює числу  $n$  за допомогою умовної форми оператора вибору (умова:  $\text{Sum}_k == n$ ). Якщо результат істина, то виводиться  $k$ , інакше нічого не виводиться.

Програмні специфікації запишемо у псевдокодi та графічній формi у вигляді блок-схеми.

*Крок 1.* Визначимо основні дії.

Крок 2. Деталізуємо задання циклу, який перебирає всі натуральні числа від 1 до заданого числа  $m$ .

Крок 3. Деталізуємо дію присвоєння початкових значень змінним.

Крок 4. Деталізуємо дію обчислення суми  $Sum\_k$  цифр числа  $k$ .

Крок 5. Деталізуємо перевірку на рівність суми цифр числа  $k$  та самого числа  $n$ .

### 3. Псевдокод алгоритму.

*Крок 1*

**початок**

введення  $m$  та  $n$

перебір натуральних чисел від 1 до  $m$

**кінець**

*Крок 2*

**початок**

введення  $m$  та  $n$

**повторити**

для  $k$  від 1 до  $m$

присвоєння початкових значень  
змінним

обчислення суми  $Sum\_k$  цифр  
числа  $k$

перевірка, чи сума цифр числа  $k$   
дорівнює числу  $n$

**все повторити**

**кінець**

*Крок 3*

**початок**

введення  $m$  та  $n$

**повторити**

для  $k$  від 1 до  $m$

$Sum\_k := 0$

$i := k$

обчислення суми  $Sum\_k$  цифр  
числа  $k$

перевірка, чи сума цифр числа  $k$   
дорівнює числу  $n$

**все повторити**

**кінець**

*Крок 4*

**початок**

введення  $m$  та  $n$

**повторити**

для  $k$  від 1 до  $m$

$Sum\_k := 0$

$i := k$

**повторити**

поки  $i > 0$

$Sum\_k := Sum\_k + i \% 10$

$i := i / 10$

**все повторити**

перевірка, чи сума цифр числа  $k$   
дорівнює числу  $n$

**все повторити**

**кінець**

*Крок 5*

**початок**

введення  $m$  та  $n$

**повторити**

для  $k$  від 1 до  $m$

$Sum\_k := 0$

$i := k$

**повторити**

**поки**  $i > 0$

$Sum\_k := Sum\_k + i \% 10$

$i := i / 10$

**все повторити**

**якщо**  $(Sum\_k == n)$

**то**

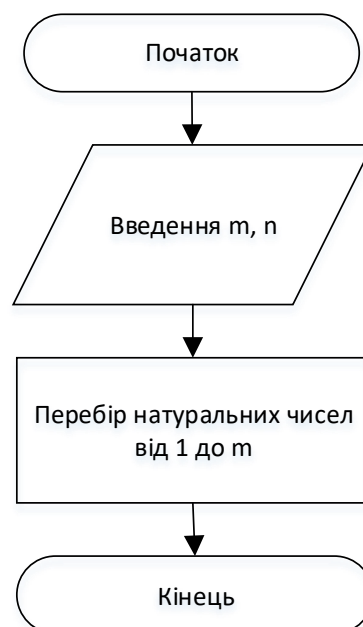
            виведення  $k$

**все повторити**

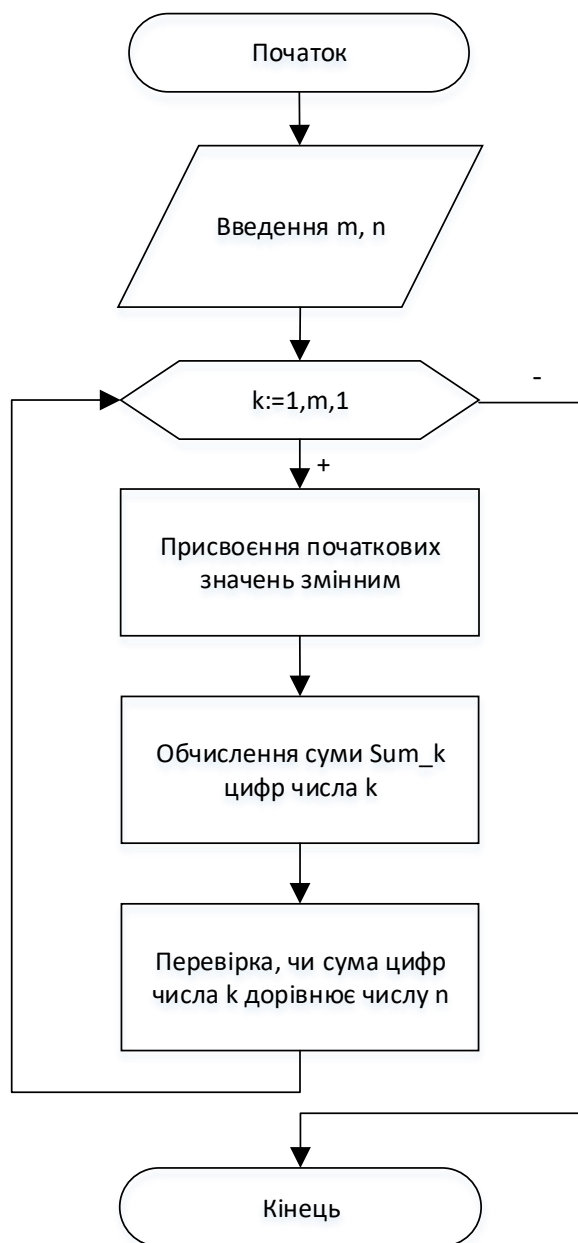
**кінець**

4. Блок-схема алгоритму.

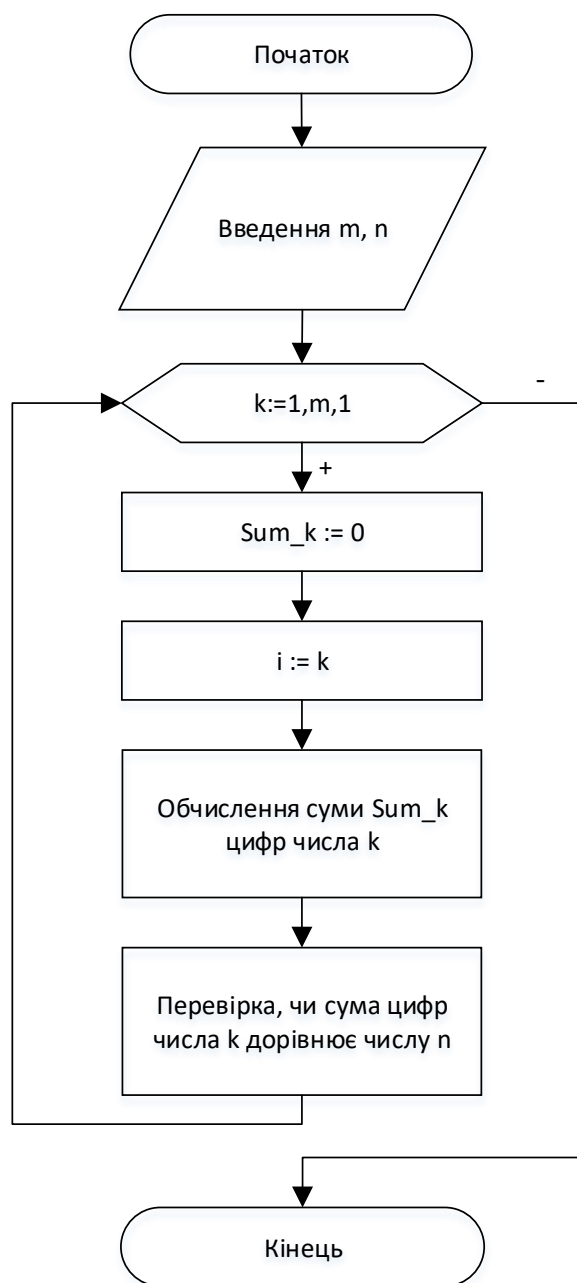
*Крок 1*



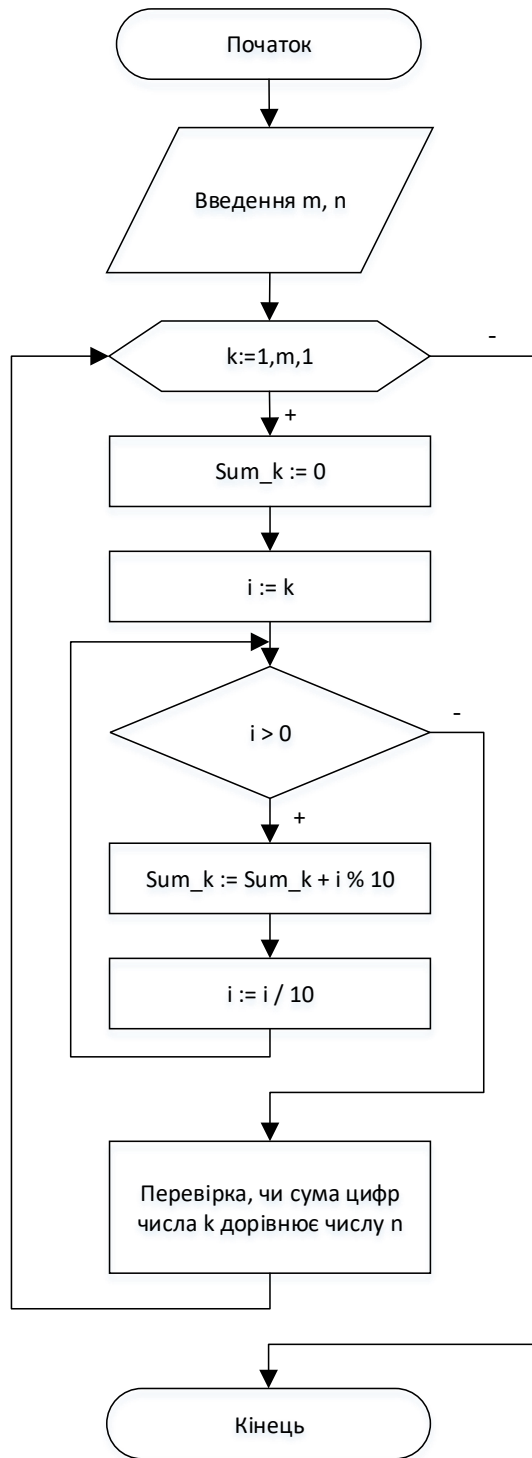
### Крок 2



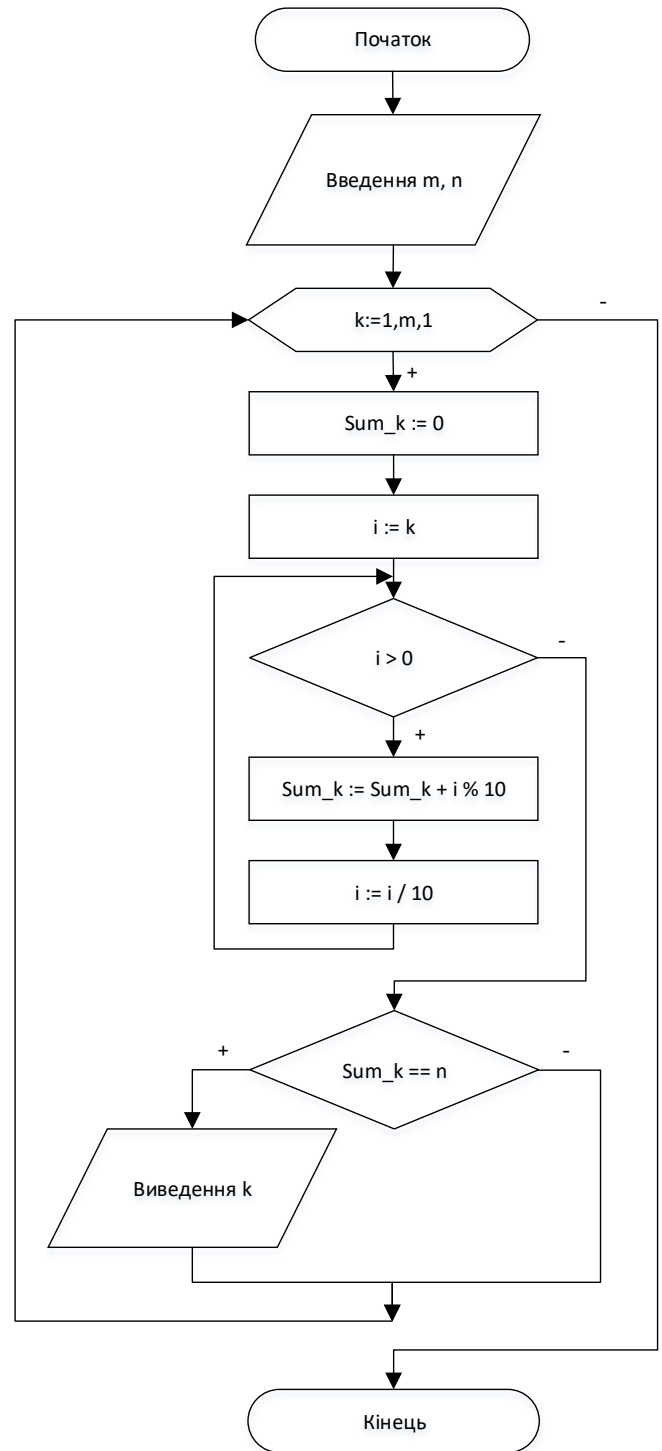
### Крок 3



#### Крок 4



#### Крок 5



5. *Випробування алгоритму.* Перевіримо правильність алгоритму на довільних конкретних значеннях початкових даних:

Блок	Дія
	<b>Початок</b>
<b>1</b>	Введення $m = 12, n = 2$
<b>2</b>	<b>повторити</b> для $k$ від 1 до 12
<b>2.1</b>	$k = 1 < 12$ - істина $Sum\_k = 0$ $i = 1$ <b>повторити поки (<math>i &gt; 0</math>):</b> 1. $i = 1 > 0$ – істина $Sum\_k = 0 + 1 \% 10 = 0 + 1 = 1$ $i = 1 / 10 = 0$ 2. $i = 0 > 0$ – хибність Вихід з циклу $1 == 2$ – хибність
<b>2.2</b>	$k = 2 < 12$ - істина $Sum\_k = 0$ $i = 2$ <b>повторити поки (<math>i &gt; 0</math>):</b> 1. $i = 2 > 0$ – істина $Sum\_k = 0 + 2 \% 10 = 0 + 2 = 2$ $i = 2 / 10 = 0$ 2. $i = 0 > 0$ – хибність Вихід з циклу $2 == 2$ – істина <b>Виведення <math>k = 2</math></b>
<b>2.3</b>	$k = 3 < 12$ - істина $Sum\_k = 0$ $i = 3$ <b>повторити поки (<math>i &gt; 0</math>):</b> 1. $i = 3 > 0$ – істина $Sum\_k = 0 + 3 \% 10 = 0 + 3 = 3$ $i = 3 / 10 = 0$ 2. $i = 0 > 0$ – хибність Вихід з циклу $3 == 2$ – хибність
<b>2.4</b>	$k = 4 < 12$ - істина $Sum\_k = 0$ $i = 4$ <b>повторити поки (<math>i &gt; 0</math>):</b> 1. $i = 4 > 0$ – істина $Sum\_k = 0 + 4 \% 10 = 0 + 4 = 4$ $i = 4 / 10 = 0$ 2. $i = 0 > 0$ – хибність Вихід з циклу $4 == 2$ – хибність

2.5	$k = 5 < 12$ - істина $Sum\_k = 0$ $i = 5$ <b>повторити поки (<math>i &gt; 0</math>):</b> 1. $i = 5 > 0$ – істина $Sum\_k = 0 + 5\%10 = 0 + 5 = 5$ $i = 5/10 = 0$ 2. $i = 0 > 0$ – хибність Вихід з циклу $5 == 2$ – хибність
2.6	$k = 6 < 12$ - істина $Sum\_k = 0$ $i = 6$ <b>повторити поки (<math>i &gt; 0</math>):</b> 1. $i = 6 > 0$ – істина $Sum\_k = 0 + 6\%10 = 0 + 6 = 6$ $i = 6/10 = 0$ 2. $i = 0 > 0$ – хибність Вихід з циклу $6 == 2$ – хибність
2.7	$k = 7 < 12$ - істина $Sum\_k = 0$ $i = 7$ <b>повторити поки (<math>i &gt; 0</math>):</b> 1. $i = 7 > 0$ – істина $Sum\_k = 0 + 7\%10 = 0 + 7 = 7$ $i = 7/10 = 0$ 2. $i = 0 > 0$ – хибність Вихід з циклу $7 == 2$ – хибність
2.8	$k = 8 < 12$ - істина $Sum\_k = 0$ $i = 8$ <b>повторити поки (<math>i &gt; 0</math>):</b> 1. $i = 8 > 0$ – істина $Sum\_k = 0 + 8\%10 = 0 + 8 = 8$ $i = 8/10 = 0$ 2. $i = 0 > 0$ – хибність Вихід з циклу $8 == 2$ – хибність
2.9	$k = 9 < 12$ - істина $Sum\_k = 0$ $i = 9$ <b>повторити поки (<math>i &gt; 0</math>):</b> 1. $i = 9 > 0$ – істина $Sum\_k = 0 + 9\%10 = 0 + 9 = 9$ $i = 9/10 = 0$ 2. $i = 0 > 0$ – хибність Вихід з циклу $9 == 2$ – хибність
2.10	$k = 10 < 12$ - істина $Sum\_k = 0$



	$i = 10$ <b>повторити поки (<math>i &gt; 0</math>):</b> 1. $i = 10 > 0$ – істина $Sum\_k = 0 + 10 \% 10 = 0 + 0 = 0$ $i = 10 / 10 = 1$ 2. $i = 1 > 0$ – істина $Sum\_k = 0 + 1 \% 10 = 0 + 1 = 1$ $i = 1 / 10 = 0$ 3. $i = 0 > 0$ – хибність Вихід з циклу $1 == 2$ – хибність
2.11	$k = 11 < 12$ - істина $Sum\_k = 0$ $i = 11$ <b>повторити поки (<math>i &gt; 0</math>):</b> 1. $i = 11 > 0$ – істина $Sum\_k = 0 + 11 \% 10 = 0 + 1 = 1$ $i = 11 / 10 = 1$ 2. $i = 1 > 0$ – істина $Sum\_k = 1 + 1 \% 10 = 1 + 1 = 2$ $i = 1 / 10 = 0$ 3. $i = 0 > 0$ – хибність Вихід з циклу $2 == 2$ – істина <b>Виведення <math>k = 11</math></b>
2.12	$k = 12 < 12$ – хибність Вихід з циклу
	<b>Кінець</b>

6. *Висновки.* На цій лабораторній роботі було досліджено особливості роботи складних циклів та було набуто практичних навичок їх використання під час складання програмних специфікацій.

Готовий алгоритм було випробувано на довільних конкретних значеннях початкових даних:  $m = 12$ ,  $n = 2$ . Після виконання алгоритму було виведено два числа: 2 та 11. Перевіримо правильність роботи алгоритму.

$2 < 12$  і  $11 < 12$  – перша умова виконується ( $k < m$ ).

Якщо ми додамо цифри отриманих чисел, то отримаємо:

для 2:  $2 = 2$ , для 11:  $1 + 1 = 2$ . Маємо, що сума цифр отриманих чисел дорівнює заданому числу  $n = 2$ .

Отже, алгоритм працює правильно.