

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 6

Виконав студент ІП-13 Вдовиченко Станіслав Юрійович  
(шифр, прізвище, ім'я, по батькові)

Перевірів Вечерковська Анастасія Сергіївна  
(прізвище, ім'я, по батькові)

Київ 20211

## Лабораторна робота 5

### Дослідження складних циклічних алгоритмів

• **Мета** – дослідити особливості роботи складних циклів та набути практичних навичок їх використання під час складання програмних специфікацій.

• **Постановка задачі (6 варіант)**

*Дано натуральне число  $n$ . Перевірити, чи можна подати  $n!$  у вигляді добутку трьох послідовних цілих чисел.*

Тобто треба знайти факторіал натурального числа  $n$  за допомогою арифметичного циклу, потім дослідити чи можна подати це значення у вигляді добутку трьох послідовних цілих чисел також використовуючи арифметичний цикл.

• **Математична модель**

| Змінна          | Тип         | Ім'я        | Призначення       |
|-----------------|-------------|-------------|-------------------|
| Число           | Натуральний | number      | Вхідні дані       |
| Лічильник циклу | Натуральний | i           | Ітераційна змінна |
| Лічильник циклу | Натуральний | n           | Ітераційна змінна |
| Добуток         | Натуральний | composition | Проміжні дані     |
| Рівність        | Логічний    | definition  | Проміжні дані     |
| Факторіал       | Натуральний | factorial   | Вихідні дані      |
| Множник         | Натуральний | a           | Вихідні дані      |
| Множник         | Натуральний | b           | Вихідні дані      |
| Множник         | Натуральний | c           | Вихідні дані      |

Розв'язання.

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

Крок 1. Визначимо основні дії.

Крок 2. Деталізуємо дію перевірки введеного числа(повинно бути  $>0$ ).

Крок 3. Деталізуємо дію знаходження значення факторіала.

Крок 4. Деталізуємо знаходження добутку.

Крок 5. Деталізуємо перевірку добутку.

Факторіал знаходимо за допомогою арифметичного циклу.

Для цього застосуємо цикл з параметром, де:

- $i := \text{number}$ ; (початкове значення ітераційної змінної, змінну `number` задаємо вручну).
- $i > 0$ ; (умова повторення, поки більше нуля)
- $i--$ ; (крок циклу, -1)

Для знаходження добутку також застосуємо цикл з параметром, де

- $n := 1$ ; (початкове значення ітераційної змінної).
- $n \leq \text{factorial}$ ; (умова повторення, поки змінна `n` не більша за змінну `factorial`).
- $n++$ ; (крок циклу, 1).

Також перед знаходженням добутку задаємо значення ( $a:=b:=c:=1$ ) трьох змінних (множників), які ймовірно будуть послідовними натуральними числами.

При кожній ітерації циклу перезадано значення множників (`a,b,c`) таким чином:

`c := b;`

`b := a;`

`a := n;`

Тобто числа будуть послідовно збільшуватись на 1 (так як `n` збільшується на 1 з кожною ітерацією).

Потім перевіряємо добуток (`composition`):

`composition := a*b*c;`

І якщо добуток (`composition`) буде дорівнювати значенню факторіала (`factorial`), то виходимо з циклу і виводимо значення добутку.

Якщо ж добуток так і не буде дорівнювати значенню факторіала, то виходимо з циклу дивлячись на умову повторення.

- **Псевдокод**

Крок 1.

**Початок**

Введення даних

Декларування змінних

Перевірка введеного значення

Знаходження значення факторіала

Знаходження добутку

Перевірка значення добутку

Виведення даних

**Кінець**

Крок 2.

**Початок**

Введення number;

factorial := 1, definition := false, a := b := c := 1;

Перевірка введеного значення

Знаходження значення факторіала

Знаходження добутку

Перевірка значення добутку

Виведення даних

**Кінець**

Крок 3.

**Початок**

Введення number;

factorial := 1, definition := false, a := b := c := 1;

**якщо** (number <= 0)

**то**

вивід “Значення задано невірно”

**все якщо**

**інакше**

Знаходження значення факторіала

Знаходження добутку

Перевірка значення добутку

**все інакше**

Виведення даних

**Кінець**

Крок 4.

**Початок**

Введення number;

factorial := 1, definition := false, a := b := c := 1;

**якщо** (number <= 0)

**то**

вивід “Значення задано невірно”

**все якщо**

**інакше**

для i від number до 0 з кроком -1 повторити

factorial = factorial\*i;

**все повторити**

Знаходження добутку

Перевірка значення добутку

**все інакше**

Виведення даних

**Кінець**

Крок 5.

**Початок**

Введення number;

factorial := 1, definition := false, a := b := c := 1;

**якщо** (number <= 0)

**то**

вивід “Значення задано невірно”

**все якщо**

**інакше**

для i від number до 0 з кроком -1 повторити

factorial = factorial\*i;

**все повторити**

для n від 1 до factorial з кроком 1 повторити

c := b;

b := a;

a := n;

composition := a\*b\*c;

Перевірка значення добутку

**все повторити**

**все інакше**

Виведення даних

**Кінець**

Крок 6.

**Початок**

Введення number;

factorial := 1, definition := false, a := b := c := 1;

**якщо** (number <= 0)

**то**

вивід “Значення задано невірно”

**все якщо**

**інакше**

**для i від number до 0 з кроком -1 повторити**

factorial = factorial\*i;

**все повторити**

**для n від 1 до factorial з кроком 1 повторити**

c := b;

b := a;

a := n;

composition := a\*b\*c;

**якщо** (composition == factorial)

definition := true;

break;

**все якщо**

**все повторити**

**все інакше**

Виведення даних

**Кінець**

Крок 7.

**Початок**

Введення number;

factorial := 1, definition := false, a := b := c := 1;

**якщо** (number <= 0)

**то**

вивід “Значення задано невірно”

**все якщо**

**інакше**

для i від number до 0 з кроком -1 повторити

factorial = factorial\*i;

**все повторити**

для n від 1 до factorial з кроком 1 повторити

c := b;

b := a;

a := n;

composition := a\*b\*c;

**якщо** (composition == factorial)

definition := true;

break;

**все якщо**

**все повторити**

**все інакше**

Виведення factorial;

**якщо** (definition == true)

Виведення a,b,c

**все якщо**

**інакше**

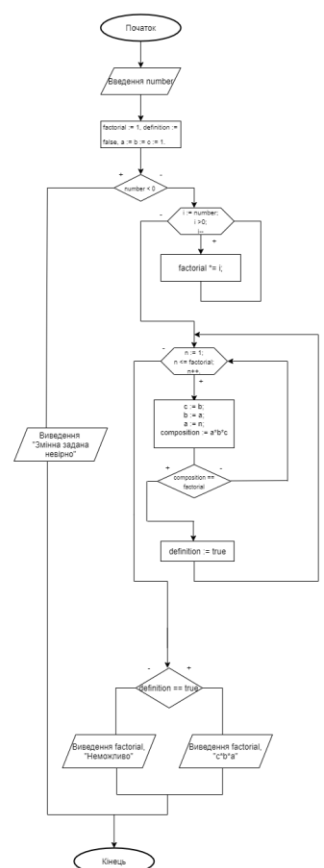
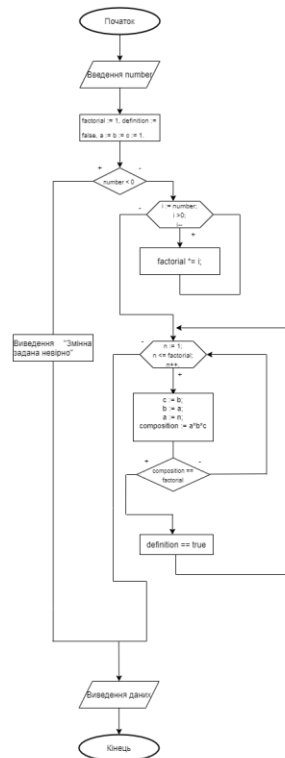
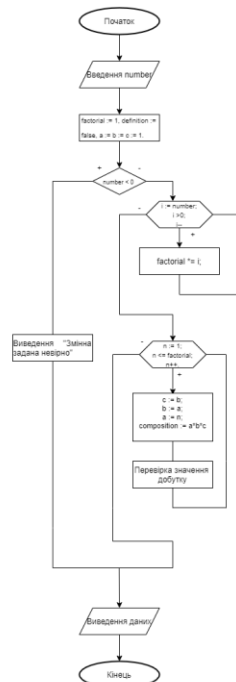
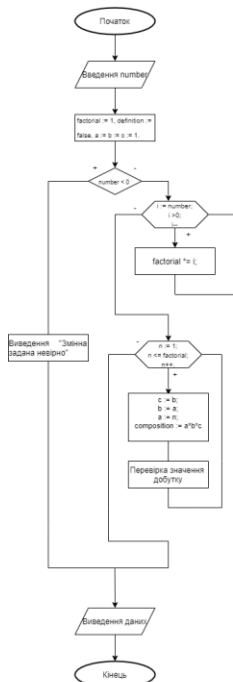
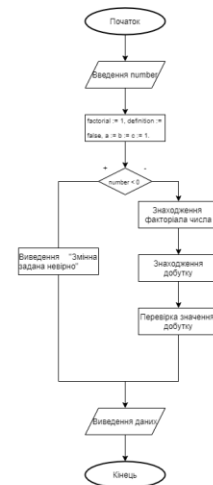
Виведення “Неможливо розкласти”

**все інакше**

**Кінець**



## • Блок-схема



- Випробування алгоритму

| Блок | Дія  |
|------|--|
|      | Початок  |
| 1    | Введення $\text{number} := 4$  |
| 2    | $4 > 0$  |
| 3    | $(i := 4; 4 > 0; 4-1) \text{ factorial} := 1*4=4;$   |
| 4    | $(i := 3; 3 > 0; 3-1) \text{ factorial} := 4*3=12;$  |
| 5    | $(i := 2; 2 > 0; 2-1) \text{ factorial} := 12*2=24;$   |
| 6    | $(i := 1; 1 > 0; 1-1) \text{ factorial} := 24*1=24;$   |
| 7    | $(0 = 0)$  |
| 8    | $(n := 1; 1 < 24; 1+1)$<br>$c := 1, b:=1 ; a := 1$<br>$\text{composition} := 1*1*1=1;$<br>$1 != 24;$   |
| 9    | $(n := 2; 2 < 24; 2+1)$<br>$c := 1, b:=1 ; a := 2;$<br>$\text{composition} := 1*1*2=2;$<br>$2 != 24;$  |
| 10   | $(n := 2; 2 < 24; 2+1)$<br>$c := 1, b:=1 ; a := 2;$<br>$\text{composition} := 1*1*2=2;$<br>$2 != 24;$  |
| 11   | $(n := 3; 3 < 24; 3+1)$<br>$c := 1, b:=2 ; a := 3;$<br>$\text{composition} := 1*2*3=6;$<br>$6 != 24;$  |
| 12   | $(n := 4; 4 < 24; 4+1)$<br>$c := 2, b:=4 ; a := 4;$<br>$\text{composition} := 2*3*4=24;$<br>$24 == 24;$<br>$\text{definition} = \text{true};$<br>$\text{break};$ |
| 13   | Виведення factorial  |
| 14   | $\text{definition} == \text{true}$<br>Виведення a,b,c  |
|      | Кінець   |

- **Висновок**

Під час виконання лабораторної роботи я дослідив особливості роботи складних циклів та набув практичних навичок їх використання під час складання програмних специфікацій. Розв'язав поставлену задачу за допомогою складних циклів. Розробив псевдокод, блок-схему, протестував алгоритм.