

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 № 6 з дисципліни  
«Алгоритми та структури даних-1.  
Основи алгоритмізації»

«Дослідження рекурсивних алгоритмів»

Варіант 26

Виконав студент ІП-12, Саркісян Валерія Георгіївна

Перевірів

\_\_\_\_\_  
( прізвище, ім'я, по батькові)

Київ 2021

# Лабораторна робота 6

## Дослідження рекурсивних алгоритмів

**Мета** - дослідити особливості роботи рекурсивних алгоритмів та набути практичних навичок їх використання під час складання програмних специфікацій підпрограм.

### Задача 26.

Задано натуральне  $n$ . Обчислити

$$\sum_{k=m}^n \frac{(-1)^k}{k!} \left( \frac{a_k + 2}{3} \right)^k \quad a_0 = 1, a_k = \sqrt{|4a_{k-1} + 2|}$$

#### 1. Постановка задачі.

У результаті розв'язку отримаємо число, яке дорівнює сумі елементів з  $m$  по  $n$ . У виразі присутній факторіал числа, тому задача буде виконуватися тільки якщо  $m$  та  $n$  невід'ємні і  $m$  менше  $n$ . Для розв'язання задачі доцільно буде розбити виконання на підпрограми, використовуючи функції. Окремо пропишемо функцію для знаходження факторіалу числа, для обчислення значення параметра  $a$ . Суму також обчислимо в окремій функції з використанням двох попередніх функцій. Для обчислення кореня квадратного використаємо функцію `sqrt()`, а для знаходження модуля виразу – функцію `abs()`.

#### 2. Побудова математичної моделі.

Змінна	Тип	Ім'я	Призначення
Початкове значення	цілий	<code>m</code>	Початкове дане
Кінцеве значення	цілий (натуральне)	<code>n</code>	Початкове дане
Лічильник	цілий	<code>k</code>	Проміжне дане, лічильник
Факторіал числа	функція	<code>factorial()</code>	Знайти факторіал <code>k</code>
Параметр функції <code>factorial()</code>	цілий	<code>c</code>	Формальний параметр, проміжне дане
Факторіал числа <code>k</code>	цілий	<code>fact</code>	Результат функції <code>factorial()</code> , вихідне дане
Значення $a_k$	функція	<code>a_k()</code>	Знаходження поточного значення $a_k$
Параметр функції <code>a_k()</code>	цілий	<code>a1</code>	Формальний параметр, проміжне дане
Параметр функції <code>a_k()</code>	цілий	<code>a2</code>	Формальний параметр, проміжне дане
Значення $a_k$	дійсний	<code>a</code>	Результат функції <code>a_k()</code> , вихідне дане
Значення суми	функція	<code>sum()</code>	Обчислення кінцевого результату виразу
Параметр функції <code>sum()</code>	цілий	<code>from</code>	Формальний параметр, проміжне дане
Параметр функції <code>sum()</code>	цілий	<code>to</code>	Формальний параметр, проміжне дане
Значення суми	дійсний	<code>suma</code>	Результат, вихідне дане

Значення виразу	дійсний	dodanok	Поточне значення виразу, проміжне дане
Корінь з числа	функція	sqrt()	Знаходження кореня квадратного з виразу
Степінь числа	функція	pow()	Піднесення чисел(виразів) до степенів
Сума	дійсний	S	Результат

### 3. Розв'язання.

#### Основна програма:

Крок 1. Визначимо основні дії.

Крок 2. Деталізуємо дію перевірки відповідності введених даних умовам, необхідним для правильного функціонування програми.

Крок 3. Деталізуємо дію знаходження суми.

#### Підпрограми:

Крок 3.2. Деталізуємо дію обчислення факторіалу.

Крок 3.3. Деталізуємо дію обчислення значення числа  $a$ .

Крок 3.4. Деталізуємо дію обчислення суми.

#### Псевдокод.

*Крок 1*

##### Початок

**Введення**  $n, m$

перевірка відповідності введених даних умовам

знаходження суми

**Виведення**  $S$

**Кінець**

*Крок 2*

##### Початок

**Введення**  $n, m$

**Поки**  $n \leq 0 \parallel m < 0 \parallel m > n$

**виконати**

Очистити екран

**Виведення** "Помилка"

**Введення**  $n, m$

**Все поки**

знаходження суми

**Виведення**  $S$

**Кінець**

*Крок 3*

##### Початок

**Введення**  $n, m$

**Поки**  $n \leq 0 \parallel m < 0 \parallel m > n$

**виконати**

Очистити екран

**Виведення** "Помилка"

**Введення**  $n, m$

**Все поки**

$S = \text{sum}(m, n)$

**Виведення**  $S$

**Кінець**

## ***Псевдокод підпрограми.***

*Крок 3.1. Функція **factorial** (c)*

**Початок** factorial (c)

**Якщо** c > 1

**То**

fact = c \* factorial (c-1)

**інакше**

fact = 1

**все якщо**

**повернути** fact

**Кінець** factorial

*Крок 3.2. Функція **a\_k** (a1, a2)*

**Початок** a\_k (a1, a2)

**Якщо** a1 == a2

**То**

a = 1

**інакше**

a = sqrt (abs (4 \* a\_k (a1 - 1, a2) + 2))

**все якщо**

**повернути** a

**Кінець** a\_k

*Крок 3.3. Функція **sum** (from, to)*

**Початок** sum (from, to)

**Для** k від from до to(включно) із кроком 1 **повторити**

dodanok = (pow (-1, k) / factorial (k)) \* pow ((a\_k (k, from) + 2) / 3, k)

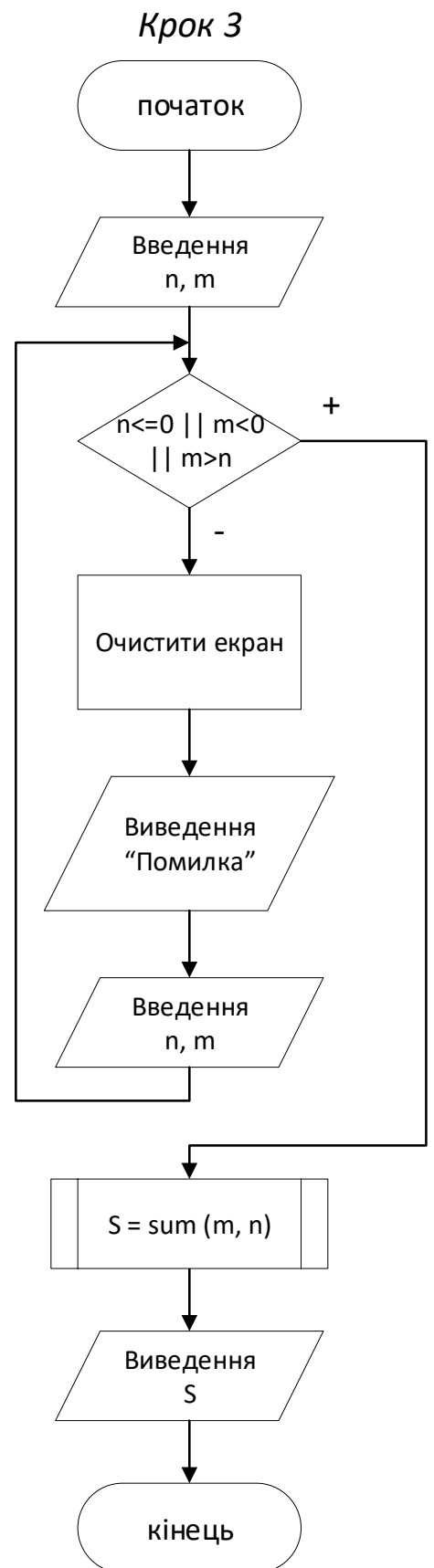
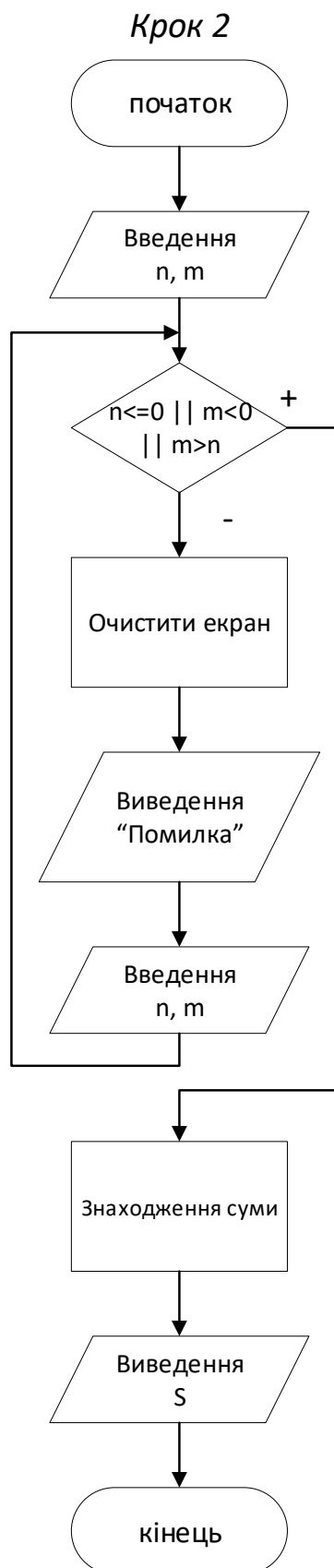
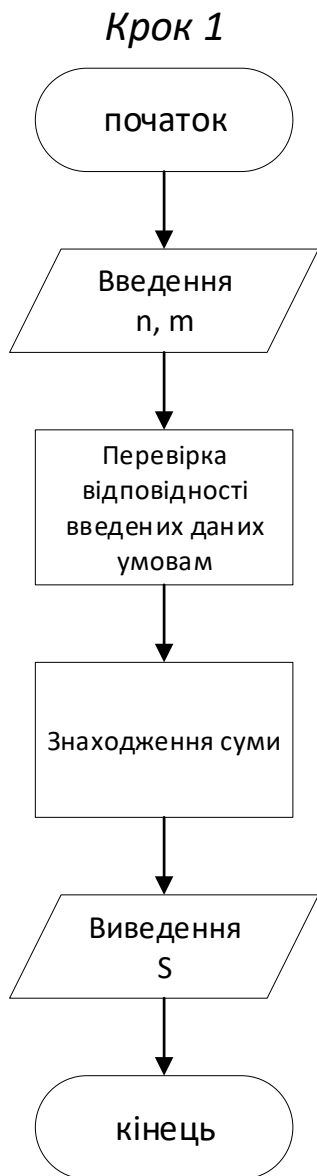
suma = suma + dodanok

**все повторити**

**повернути** suma

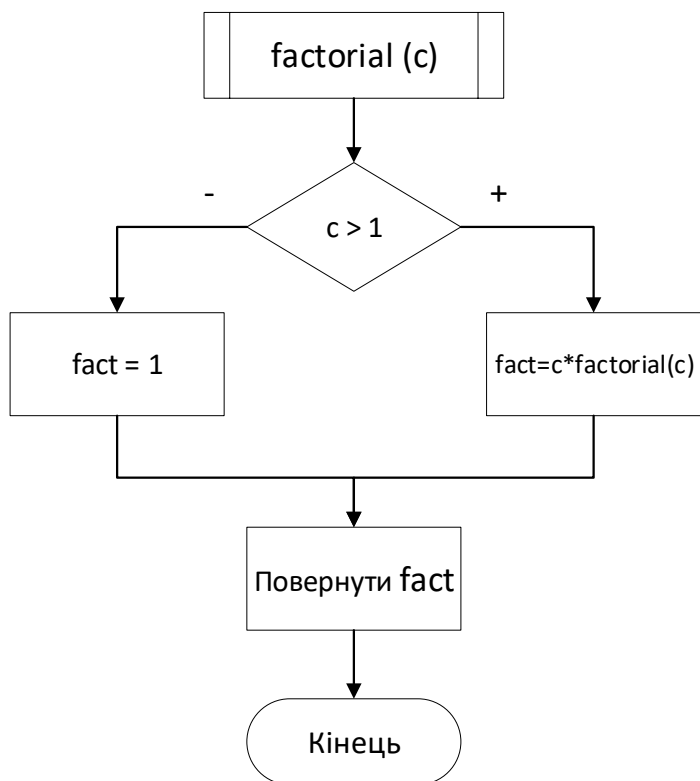
**Кінець** sum

## ***4. Блок-схема.***

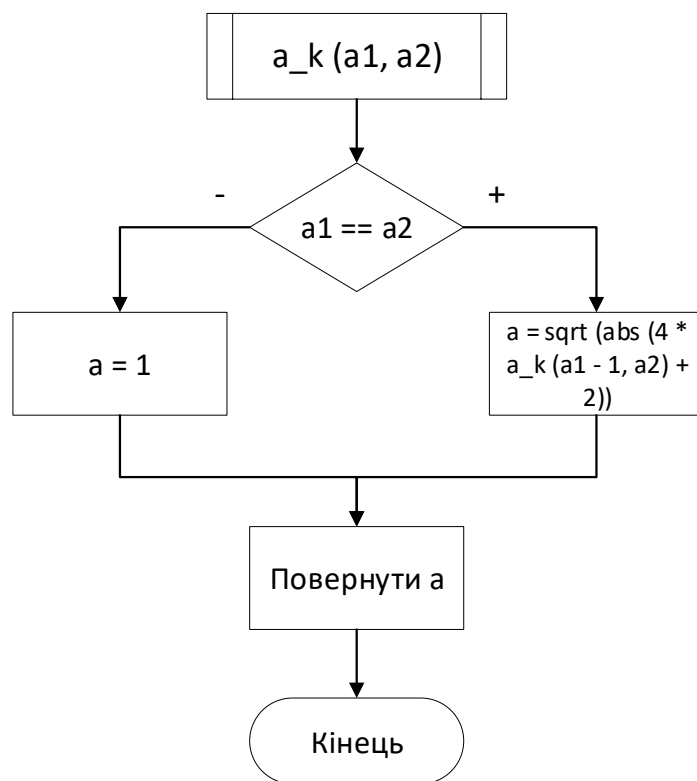


## Підпрограми:

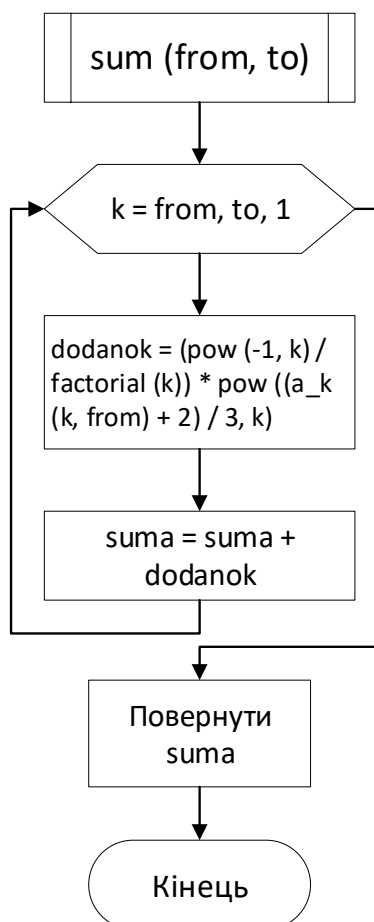
Крок 3.1



Крок 3.2



Крок 3.3



## 5. Код програми (C++):

```
#include <iostream>
#include <cmath>
using namespace std;

int factorial(int k);
double a_k(int a1, int a2);
double sum(int from, int to);

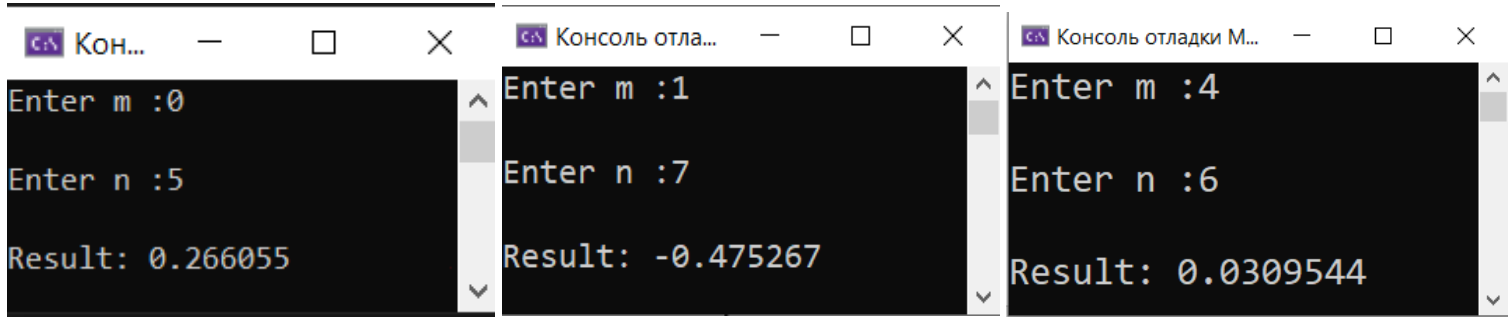
int main(){
    int n, m;
    cout << "Enter m :"; cin >> m;
    cout << "\nEnter n :"; cin >> n;
    while (n <= 0 or m < 0 or m > n) {
        system("cls");
        cout << "Please enter correct numbers\n" << "Enter m :"; cin >> m;
        cout << "\nEnter n :"; cin >> n;
    }
    double S = sum(m, n);
    cout << "\nResult: " << S;
    return 0;
}

int factorial(int c) {
    int fact;
    if (c > 1) {
        fact = c * factorial(c - 1);
    }
    else
        fact = 1;
    return fact;
}

double a_k(int a1, int a2) {
    double a;
    if (a1 == a2)
        a = 1;
    else
        a = sqrt(abs(4 * a_k(a1 - 1, a2) + 2));
    return a;
}

double sum(int from, int to) {
    double suma = 0;
    for (int k = from; k <= to; k++) {
        double dodanok = (pow(-1, k) / factorial(k)) * pow((a_k(k, from) + 2) / 3, k);
        suma = suma + dodanok;
    }
    return suma;
}
```

**Тестування:**



6. *Випробування алгоритму.* Перевіримо правильність роботи алгоритму для довільних значень  $m$  і  $n$ :

Блок	Дія
	Початок
1	Введення $m=-3, n=5$
2	$m < 0$
3	Очистити екран
4	Виведення “Please enter correct numbers”
5	Введення $m=0, n=5$
6	$k=0$ factorial ( $c=0$ ): $c! > 1$ $fact=1$ $a\_k(a1=0, a2=0)$ : $a1 == a2$ $a=1.0$ $dodanok = (pow(-1, 0) / 1) * pow((1+2)/3, 0) = 1$ $suma = 0 + 1 = 1$
7	$k=1$ factorial ( $c=1$ ): $c! > 1$ $fact=1$ $a\_k(a1=1, a2=0)$ : $a1 != a2$ $a = \sqrt{abs(4 * a\_k(a1 - 1, a2) + 2)}$ $a\_k(a1 - 1, a2) = 1$ $a = \sqrt{abs(4 * 1 + 2)} \approx 2,45$ $dodanok = (pow(-1, 1) / 1) * pow((2,45+2)/3, 1) \approx -1,483$ $suma = 1 - 1,483 \approx -0,483$



8	<p>k=2</p> <p>factorial (c=2):</p> <p>    c&gt;1</p> <p>    fact=2*1=2</p> <p>a_k(a1=2, a2=0):</p> <p>    a1!=a2</p> <p>        a=sqrt (abs (4 * a_k (a1 - 1, a2) + 2))</p> <p>        a_k (a1 - 1, a2) <math>\approx</math> 2,45</p> <p>        a=sqrt (abs (4 *2,45 + 2)) <math>\approx</math>3,435</p> <p>dodanok=(pow (-1, 2) / 2) * pow ((3,435+ 2) / 3, 2) <math>\approx</math> 1,64</p> <p>suma=-0,483+1,64<math>\approx</math>1,158</p>
9	<p>k=3</p> <p>factorial (c=3):</p> <p>    c&gt;1</p> <p>    fact=3*2*1=6</p> <p>a_k(a1=3, a2=0):</p> <p>    a1!=a2</p> <p>        a=sqrt (abs (4 * a_k (a1 - 1, a2) + 2))</p> <p>        a_k (a1 - 1, a2) <math>\approx</math> 3,435</p> <p>        a=sqrt (abs (4 *3,435 + 2)) <math>\approx</math>3,97</p> <p>dodanok=(pow (-1, 3) / 6) * pow ((3,97+ 2) / 3, 3) <math>\approx</math> -1,31</p> <p>suma=1,158-1,31<math>\approx</math>-0,154</p>
10	<p>k=4</p> <p>factorial (c=4):</p> <p>    c&gt;1</p> <p>    fact=4*3*2*1=24</p> <p>a_k(a1=4, a2=0):</p> <p>    a1!=a2</p> <p>        a=sqrt (abs (4 * a_k (a1 - 1, a2) + 2))</p> <p>        a_k (a1 - 1, a2) <math>\approx</math> 3,97</p> <p>        a=sqrt (abs (4 *3,97 + 2)) <math>\approx</math> 4,23</p> <p>dodanok=(pow (-1, 4) / 24) * pow ((4,23+ 2) / 3, 4) <math>\approx</math> 0,77</p> <p>suma=-0,154+0,77<math>\approx</math>0,62</p>
11	<p>k=5</p> <p>factorial (c=5):</p> <p>    c&gt;1</p> <p>    fact=5*4*3*2*1=120</p> <p>a_k(a1=5, a2=0):</p> <p>    a1!=a2</p> <p>        a=sqrt (abs (4 * a_k (a1 - 1, a2) + 2))</p>

	$a_k(a_{l-1}, a_2) \approx 4,23$ $a = \sqrt{\text{abs}(4 * 4,23 + 2)} \approx 4,35$ $\text{dodanok} = (\text{pow}(-1, 5) / 120) * \text{pow}((4,35 + 2) / 3, 5) \approx -0,35$ $\text{suma} = 0,62 - 0,35 \approx 0,266055$
12	$S = \text{sum}(0, 5) \approx 0,266055$
13	Виведення $S = 0,266055$
	Кінець

## 7. Висновки.

На цій лабораторній роботі було досліджено особливості роботи рекурсивних алгоритмів та набуто практичних навичок їх використання під час складання програмних специфікацій підпрограм. Для розв'язання поставленої задачі було використано 3 допоміжних алгоритми-функції, у 2 з яких були використовували просту рекурсію (функції `factorial()` та `a_k()`), а в третій використовуючи складну рекурсію викликано 2 попередні функції.