

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

**Лабораторна робота №1**

з дисципліни  
«Алгоритми і структури даних»

Виконав:

студент групи ІМ-43

Олексійчук Станіслав Юрійович

номер у списку групи: 22

Перевірив:

Сергієнко А. М.

Київ 2025

## ***Постановка задачі***

Дане натуральне число  $n$ . Знайти суму перших  $n$  членів ряду чисел, заданого рекурентною формулою. Розв'язати задачу трьома способами:

- 1) у програмі використати рекурсивну функцію, яка виконує обчислення і членів ряду, і суми на рекурсивному спуску;
- 2) у програмі використати рекурсивну функцію, яка виконує обчислення і членів ряду, і суми на рекурсивному поверненні;
- 3) у програмі використати рекурсивну функцію, яка виконує обчислення членів ряду на рекурсивному спуску, а обчислення суми на рекурсивному поверненні.

При проектуванні програм слід врахувати наступне:

- 1) програми повинні працювати коректно для довільного цілого додатного  $n$  включно з  $n = 1$ ;
- 2) видимість змінних має обмежуватися тими ділянками, де вони потрібні;
- 3) функції повинні мати властивість модульності;
- 4) у кожному з трьох способів рекурсивна функція має бути одна (за потреби, можна також використати додаткову функцію-обгортку (wrapper function));
- 5) у другому способі можна використати запис (struct) з двома полями (але в інших способах у цьому немає потреби і це вважатиметься надлишковим);
- 6) програми мають бути написані мовою програмування C.

### ***Варіант 22:***

$$F_1 = x - 1; \quad F_{i+1} = -F_i \cdot i(x - 1)/(i + 1), \quad i > 0;$$

$$\sum_{i=1}^n F_i = \ln x, \quad 0 < x < 2.$$

# *Текст програми*

```
#include <stdio.h>
```

```
float recDescent(float x, unsigned int n, float Fi, float sum, int i) {  
    if (i > n) {  
        return sum - Fi;  
    }  
    Fi *= -i * (x - 1) / (i + 1);  
    sum += Fi;  
    return recDescent(x, n, Fi, sum, i + 1);  
}
```

```
float recAscent(float x, unsigned int n, float Fi, int i) {  
    float sum = -1;  
    if (i > n) {  
        sum = 0;  
    }  
    else {  
        float element = -Fi * i * (x - 1) / (i + 1);  
        sum = Fi + recAscent(x, n, element, i + 1);  
    }  
    return sum;  
}
```

```
float recMixed(float x, unsigned int n, float Fi, int i) {  
    if (i > n) {  
        return 0;  
    }  
    float element = -Fi * i * (x - 1) / (i + 1);  
    float sum = recMixed(x, n, element, i + 1);  
    return Fi + sum;  
}
```

```

float loopTesting (float x, unsigned int n, float Fi, float sum) {
    for (int i = 1; i <= n; i++) {
        Fi *= -i * (x - 1) / (i + 1);
        sum += Fi;
    }
    return sum - Fi;
}

int main() {
    float x;
    unsigned int n;
    printf("Choose the value of x: ");
    scanf("%f", &x);
    printf("Choose the value of n: ");
    scanf("%d", &n);
    if (x >= 2 || x <= 0 || n <= 0) {
        printf("You chose the incorrect input!");
        return 0;
    }
    float Fi = x - 1;
    printf("Descent recursion - the value of sum: %f\n", recDescent(x, n, Fi, x -
1, 1));
    printf("Ascent recursion - the value of sum: %f\n", recAscent(x, n, Fi, 1));
    printf("Mixed recursion - the value of sum: %f\n", recMixed(x, n, Fi, 1));
    printf("Loop testing - the value of sum: %f\n", loopTesting(x, n, Fi, x - 1));
    return 0;
}

```

## *Результати тестування програми*

```
Choose the value of x: 0.5
Choose the value of n: 5
Descent recursion - the value of sum: -0.688542
Ascent recursion - the value of sum: -0.688542
Mixed recursion - the value of sum: -0.688542
Loop testing - the value of sum: -0.688542
```

```
Choose the value of x: 0.9
Choose the value of n: 5
Descent recursion - the value of sum: -0.105360
Ascent recursion - the value of sum: -0.105360
Mixed recursion - the value of sum: -0.105360
Loop testing - the value of sum: -0.105360
```

```
Choose the value of x: 1
Choose the value of n: 5
Descent recursion - the value of sum: 0.000000
Ascent recursion - the value of sum: 0.000000
Mixed recursion - the value of sum: 0.000000
Loop testing - the value of sum: 0.000000
```

```
Choose the value of x: 1.2
Choose the value of n: 5
Descent recursion - the value of sum: 0.182331
Ascent recursion - the value of sum: 0.182331
Mixed recursion - the value of sum: 0.182331
Loop testing - the value of sum: 0.182331
```

```
Choose the value of x: 1.6
Choose the value of n: 5
Descent recursion - the value of sum: 0.475152
Ascent recursion - the value of sum: 0.475152
Mixed recursion - the value of sum: 0.475152
Loop testing - the value of sum: 0.475152
```

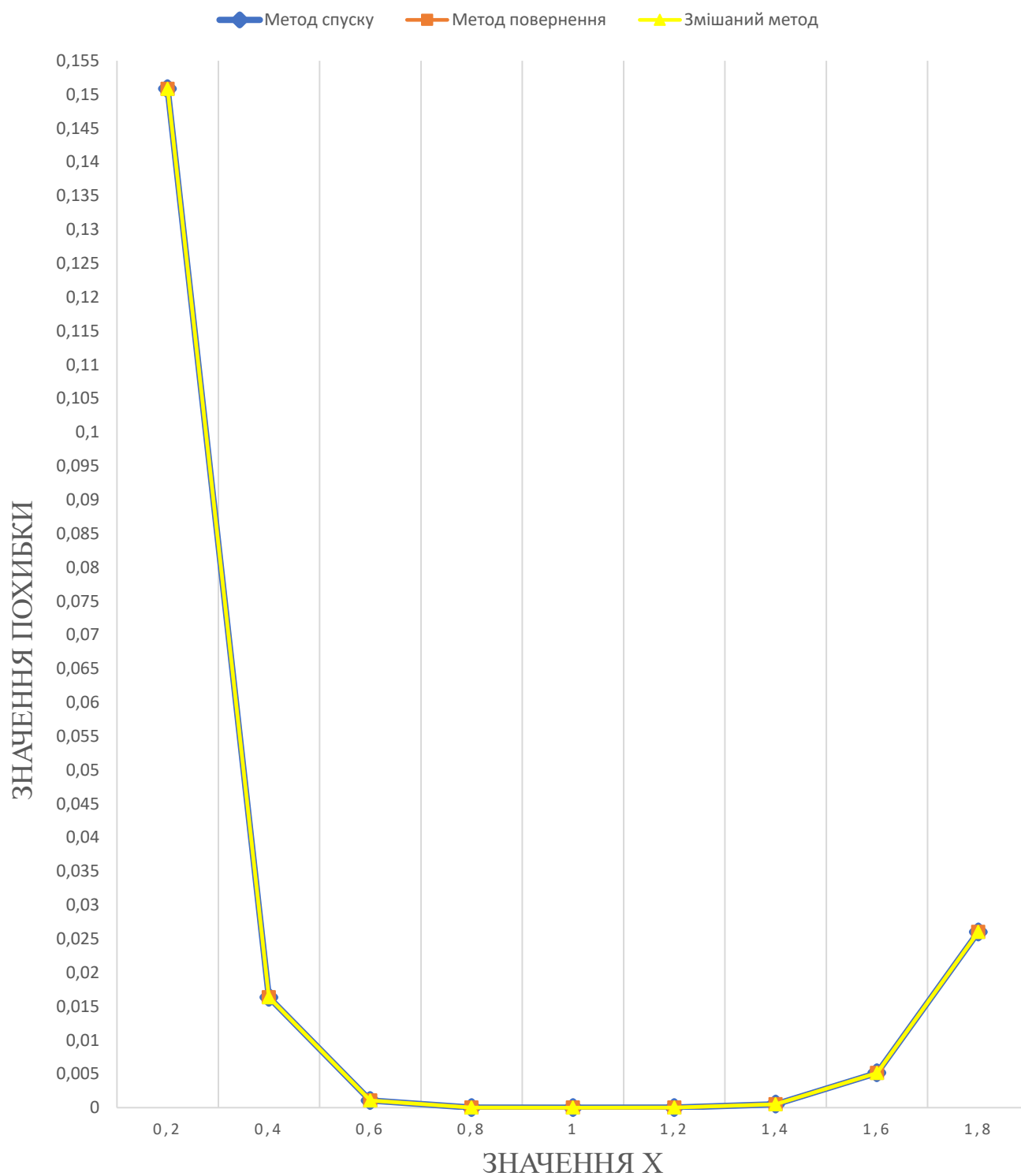
Перевірка даних тестування з допомогою калькулятора WolframAlpha ( $\log(x)$  – натуральний логарифм):

$\log(x)$  is the natural logarithm

<table><tr><th>Input</th></tr><tr><td><math>\log(0.5)</math></td></tr><tr><th>Result</th></tr><tr><td><u><math>-0.693147\dots</math></u></td></tr></table>	Input	$\log(0.5)$	Result	<u><math>-0.693147\dots</math></u>	<table><tr><th>Input</th></tr><tr><td><math>\log(0.9)</math></td></tr><tr><th>Result</th></tr><tr><td><math>-0.105361\dots</math></td></tr></table>	Input	$\log(0.9)$	Result	$-0.105361\dots$	<table><tr><th>Input</th></tr><tr><td><math>\log(1)</math></td></tr><tr><th>Result</th></tr><tr><td>0</td></tr></table>	Input	$\log(1)$	Result	0
Input														
$\log(0.5)$														
Result														
<u><math>-0.693147\dots</math></u>														
Input														
$\log(0.9)$														
Result														
$-0.105361\dots$														
Input														
$\log(1)$														
Result														
0														
<table><tr><th>Input</th></tr><tr><td><math>\log(1.2)</math></td></tr><tr><th>Result</th></tr><tr><td>0.182322...</td></tr></table>	Input	$\log(1.2)$	Result	0.182322...	<table><tr><th>Input</th></tr><tr><td><math>\log(1.6)</math></td></tr><tr><th>Result</th></tr><tr><td>0.470004...</td></tr></table>	Input	$\log(1.6)$	Result	0.470004...					
Input														
$\log(1.2)$														
Result														
0.182322...														
Input														
$\log(1.6)$														
Result														
0.470004...														

Деякі значення, обчислені на калькуляторі, дещо відрізняються від обчислення рекурсивним способом, оскільки присутня похибка обчислення. Графік цього буде зображено в наступному розділі.

## ГРАФІК ПОХИБКИ ОБЧИСЛЕННЯ ФУНКЦІЇ ( $n=5$ )



## ***Висновки***

У ході виконання лабораторної роботи я навчився використовувати рекурсивні методи мови програмування C для розв'язування певних математичних задач, а саме обчислення суми рекурсивно заданих доданків; зрозумів, що таке рекурсивні спуск та повернення і використав їх під час виконання роботи: як окремо, так і разом; розглянув похибку обчислення функції, а також представив це на графіку, використовуючи кожний з методів: метод спуску, повернення та змішаний; закріпив знання про модульність, підпрограми, функції та вивчив нову теоретичну базу, пов'язану з рекурсією: зрозумів її особливості та недоліки; у результаті практичних та теоретичних завдань зрозумів важливість лабораторної роботи та здобув нові знання з рекурсії.