

**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-42

Лобань Михайло Юрійович

номер у списку групи: 20

Перевірила:

Сергієнко А. М.

Київ 2025

### Загальна постановка завдання

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  $i$  членів ряду, і суми на рекурсивному спуску;
- 2) у програмі використати рекурсивну функцію, яка виконує обчислення  $i$  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.

### Завдання за варіантом

#### Варіант № 20

$$F_1 = x; \quad F_{i+1} = F_i \cdot x^2(2i - 1)/(2i + 1), \quad i > 0;$$

$$\sum_{i=1}^n F_i = \operatorname{arth} x, \quad |x| < 1.$$

### Текст програми

```
#include <stdio.h>

#include <stdlib.h>

typedef struct {
    double sum;
    double F;
} Result;

double sum1(int n, double x, int i, double F, double sum) {
    if (x < 1 && x > -1) {
        sum += F;
        printf("F%d = %lf, sum = %lf\n", i, F, sum);

        if (i == n) {
```

```

        return sum;
    }

    return sum1(n, x, i + 1, F * (x * x * (2 * i - 1)) / (2 * i + 1), sum);
}

return sum;
}

```

```

Result sum2(int n, double x, int i) {

```

```

    Result result;

```

```

    if (x < 1 && x > -1) {

```

```

        if (i == 1) {

```

```

            result.F = x;

```

```

            result.sum = x;

```

```

            printf("F%d = %lf, sum = %lf\n", i, result.F, result.sum);

```

```

            return result;

```

```

        }

```

```

        Result prev = sum2(n, x, i - 1);

```

```

        result.F = prev.F * (x * x * (2 * i - 3)) / (2 * i - 1);

```

```

        result.sum = prev.sum + result.F;

```

```

        printf("F%d = %lf, sum = %lf\n", i, result.F, result.sum);

```

```

        return result;

```

```

    }

```

```

    return result;
}

```

```
}
```

```
double sum3(int n, double x, int i, double F) {  
    if (x < 1 && x > -1) {  
        if (i == n) {  
            printf("F%d = %lf, sum = %lf\n", i, F, F);  
            return F;  
        }  
  
        double sum = sum3(n, x, i + 1, F * (x * x * (2 * i - 1)) / (2 * i + 1));  
        sum += F;  
  
        printf("F%d = %lf, sum = %lf\n", i, F, sum);  
        return sum;  
    }  
    return 0;  
}
```

```
double loops(int n, double x, int i, double F, double sum) {  
    if (x < 1 && x > -1) {  
        for(int i = 1; i <= n; i++) {  
            sum += F;  
            printf("F%d = %lf, sum = %lf\n", i, F, sum);  
            F *= (x * x * (2 * i - 1)) / (2 * i + 1);  
        }  
        return sum;  
    }  
    return sum;
```

```
}
```

```
int main() {  
    int n = 5;  
    int i = 1;  
    double x = 0;  
    double F = x;  
  
    printf("=== First method ===\n");  
    double result1 = sum1(n, x, i, F, 0);  
    printf("Result: %lf\n\n", result1);  
  
    printf("=== Second method ===\n");  
    double result2 = sum2(n, x, n).sum;  
    printf("Result: %lf\n\n", result2);  
  
    printf("=== Third method ===\n");  
    double result3 = sum3(n, x, i, F);  
    printf("Result: %lf\n\n", result3);  
  
    printf("=== Loops ===\n");  
    double result_loops = loops(n, x, i, F, 0);  
    printf("Result: %lf\n\n", result_loops);  
  
    return 0;  
}
```

## Результати тестування програми

- $X = 0.5$

```
=== First method ===  
F1 = 0.500000, sum = 0.500000  
F2 = 0.041667, sum = 0.541667  
F3 = 0.006250, sum = 0.547917  
F4 = 0.001116, sum = 0.549033  
F5 = 0.000217, sum = 0.549250  
Result: 0.549250
```

```
=== Second method ===  
F1 = 0.500000, sum = 0.500000  
F2 = 0.041667, sum = 0.541667  
F3 = 0.006250, sum = 0.547917  
F4 = 0.001116, sum = 0.549033  
F5 = 0.000217, sum = 0.549250  
Result: 0.549250
```

```
=== Third method ===  
F5 = 0.000217, sum = 0.000217  
F4 = 0.001116, sum = 0.001333  
F3 = 0.006250, sum = 0.007583  
F2 = 0.041667, sum = 0.049250  
F1 = 0.500000, sum = 0.549250  
Result: 0.549250
```

```
=== Loops ===  
F1 = 0.500000, sum = 0.500000  
F2 = 0.041667, sum = 0.541667  
F3 = 0.006250, sum = 0.547917  
F4 = 0.001116, sum = 0.549033  
F5 = 0.000217, sum = 0.549250  
Result: 0.549250
```

- $X=0.4$

```
=== First method ===  
F1 = 0.400000, sum = 0.400000  
F2 = 0.021333, sum = 0.421333  
F3 = 0.002048, sum = 0.423381  
F4 = 0.000234, sum = 0.423615  
F5 = 0.000029, sum = 0.423645  
Result: 0.423645
```

```
=== Second method ===  
F1 = 0.400000, sum = 0.400000  
F2 = 0.021333, sum = 0.421333  
F3 = 0.002048, sum = 0.423381  
F4 = 0.000234, sum = 0.423615  
F5 = 0.000029, sum = 0.423645  
Result: 0.423645
```

```
=== Third method ===  
F5 = 0.000029, sum = 0.000029  
F4 = 0.000234, sum = 0.000263  
F3 = 0.002048, sum = 0.002311  
F2 = 0.021333, sum = 0.023645  
F1 = 0.400000, sum = 0.423645  
Result: 0.423645
```

```
=== Loops ===  
F1 = 0.400000, sum = 0.400000  
F2 = 0.021333, sum = 0.421333  
F3 = 0.002048, sum = 0.423381  
F4 = 0.000234, sum = 0.423615  
F5 = 0.000029, sum = 0.423645  
Result: 0.423645
```

- $X = 0.2$

```
=== First method ===  
F1 = 0.200000, sum = 0.200000  
F2 = 0.002667, sum = 0.202667  
F3 = 0.000064, sum = 0.202731  
F4 = 0.000002, sum = 0.202732  
F5 = 0.000000, sum = 0.202733  
Result: 0.202733
```

```
=== Second method ===  
F1 = 0.200000, sum = 0.200000  
F2 = 0.002667, sum = 0.202667  
F3 = 0.000064, sum = 0.202731  
F4 = 0.000002, sum = 0.202732  
F5 = 0.000000, sum = 0.202733  
Result: 0.202733
```

```
=== Third method ===  
F5 = 0.000000, sum = 0.000000  
F4 = 0.000002, sum = 0.000002  
F3 = 0.000064, sum = 0.000066  
F2 = 0.002667, sum = 0.002733  
F1 = 0.200000, sum = 0.202733  
Result: 0.202733
```

```
=== Loops ===  
F1 = 0.200000, sum = 0.200000  
F2 = 0.002667, sum = 0.202667  
F3 = 0.000064, sum = 0.202731  
F4 = 0.000002, sum = 0.202732  
F5 = 0.000000, sum = 0.202733  
Result: 0.202733
```

- Перевірка за допомогою калькулятора:

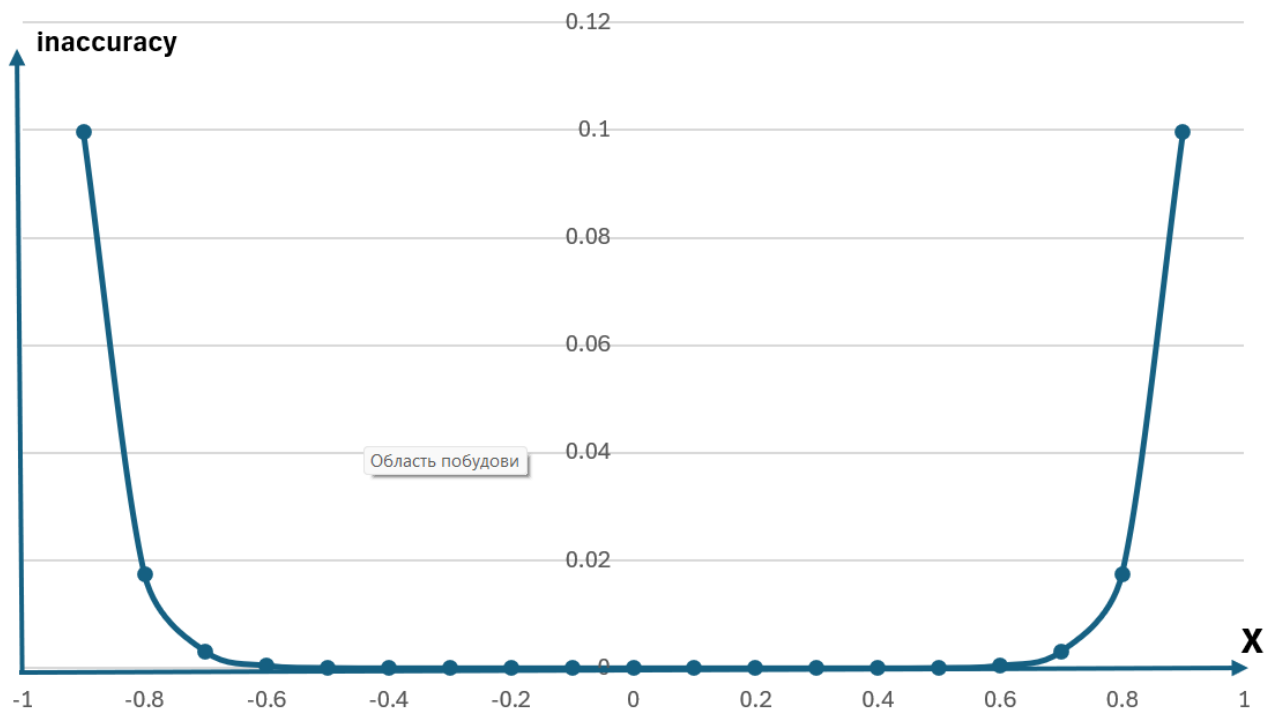


$\tanh^{-1}(0.2) =$   
**0.2027325540540821909890065**  
**5773217**

$\tanh^{-1}(0.4) =$   
**0.4236489301936018068550537**  
**5326033**

$\tanh^{-1}(0.5) =$   
**0.5493061443340548456976226**  
**1846126**

**Графік залежності похибки обчислення заданої функції від значення  $x$  при фіксованому значенні  $n=5$ .**



### **Висновки**

Після виконання лабораторної роботи, засвоїв теоретичний матеріал та набв практичного досвіду створення рекурсивних алгоритмів та написання відповідних їм програм.