

**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»

Виконала:  
студентка групи ІМ-43:  
Козаченко Софія Олександрівна  
номер у списку групи: 14

Перевірів:  
Сергієнко А. М.

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

**Варіант 14:**

$$F_0 = x; \quad F_i = F_{i-1} \cdot (2i - 1)^2 \cdot x^2 / (2i(2i + 1)), \quad i > 0;$$

$$\pi/2 - \sum_{i=0}^n F_i = \arccos x.$$

**Тексти програм:****1) обчислення  $i$  членів ряду,  $i$  суми на рекурсивному спуску:**

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

```
#include <math.h>
```

```
double recursiveDecent(double x, int n, int i, double Fi, double sum){  
    double res = 0;  
    if(n==0){  
        res = sum;  
    }else{  
        Fi = Fi*x*x*(2*i-1)*(2*i-1)/(2*i*(2*i+1));  
        res = recursiveDecent(x, n-1, i+1, Fi, sum + Fi);  
    }  
    return res;  
}
```

```
int main(){  
    int n = 0;  
    double x = 0;  
    printf("Enter n: ");  
    scanf("%d", &n);
```

```

printf("Enter x: ");
scanf("%lf", &x);

double F0 = x;
double sum = F0;
double result = M_PI_2 - recursiveDecent(x, n, 1, F0, sum);
printf("Result_1 = %1.8f\n", result);
}

```

## 2) обчислення і членів ряду, і суми на рекурсивному поверненні:

```

#include <stdio.h>
#include <math.h>

typedef struct{
    double sum;
    double Fi;
} str;

str recursiveReturn(double x, int i){
    str res = {
        .sum=0,
        .Fi=0
    };
    str temp;
    if(i==0){
        res.Fi = x;
        res.sum= res.Fi;
    }else{
        temp = recursiveReturn(x,i-1);
        res.Fi = temp.Fi * x*x*(2*i-1)*(2*i-1)/(2*i*(2*i+1));
        res.sum = temp.sum + res.Fi;
    }
    return res;
}

int main(){
    int n = 0;
    double x = 0;

```

```

printf("Enter n: ");
scanf("%d", &n);
printf("Enter x: ");
scanf("%lf", &x);

double result = M_PI_2 - recursiveReturn(x, n).sum;
printf("Result_2 = %1.8f\n", result);
}

```

### **3) обчислення членів ряду на рекурсивному спуску, а обчислення суми на рекурсивному поверненні:**

```

#include <stdio.h>
#include <math.h>

double recursiveMix(double x, int n, int i, double Fi){
    double sum = 0;
    if(n==0){
        sum = x;
    }else{
        Fi = Fi*x*x*(2*i-1)*(2*i-1)/(2*i*(2*i+1));
        sum = Fi + recursiveMix(x, n-1, i+1, Fi);
    }
    return sum;
}

int main(){
    int n = 0;
    double x = 0;

    printf("Enter n: ");
    scanf("%d", &n);
    printf("Enter x: ");
    scanf("%lf", &x);

    double F0 = x;
    double result = M_PI_2 - recursiveMix(x, n, 1, F0);
    printf("Result_3 = %1.8f\n", result);
}

```

#### 4) циклічний варіант рішення задачі:

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

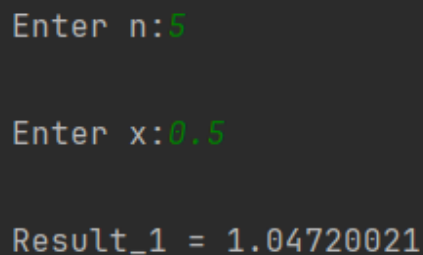
```
#include <math.h>
```

```
double result(double x, unsigned n) {  
    double res = x;  
    double prev = res;  
    for (unsigned i = 1; i <= n; i++) {  
        prev *= ((2 * i - 1) * (2 * i - 1) * x * x) / (2 * i * (2 * i + 1));  
        res += prev;  
    }  
    return M_PI/2 - res;  
}
```

```
int main() {  
    int n = 0;  
    double x = 0;  
  
    printf("Enter n : ");  
    scanf("%d", &n);  
    printf("Enter x : ");  
    scanf("%lf", &x);  
  
    double RESULT = result (x, n);  
    printf("Result = %1.8lf\n", RESULT);  
    return 0;  
}
```

#### Скриншоти тестування програм:

##### 1)Спуск:



```
Enter n:5  
  
Enter x:0.5  
  
Result_1 = 1.04720021
```

## 2)Повернення:

```
Enter n:5  
  
Enter x:0.5  
  
Result_2 = 1.04720021
```

## 3)Мікс:

```
Enter n:5  
  
Enter x:0.5  
  
Result_3 = 1.04720021
```

## 4)Циклічний:

```
Enter n :5  
  
Enter x :0.5  
  
Result = 1.04720021
```

## 5)Калькулятор:

1	$F_0(x) = x_0$	×
2	$F_5(x) = \frac{F_4(x) \cdot (2 \cdot 5 - 1)^2 \cdot x^2}{10(10 + 1)}$	×
3	$F_4(x) = \frac{F_3(x) \cdot (2 \cdot 4 - 1)^2 \cdot x^2}{8(8 + 1)}$	×
4	$F_3(x) = \frac{F_2(x) \cdot (2 \cdot 3 - 1)^2 \cdot x^2}{6(6 + 1)}$	×
5	$F_2(x) = \frac{F_1(x) \cdot (2 \cdot 2 - 1)^2 \cdot x^2}{4(4 + 1)}$	×
6	$F_1(x) = \frac{F_0(x) \cdot (2 \cdot 1 - 1)^2 \cdot x^2}{2(2 + 1)}$	×
7	$\frac{\pi}{2} - (F_0(x_0) + F_1(x_0) + F_2(x_0) + F_3(x_0) + F_4(x_0) + F_5(x_0)) =$ $= 1.0472002075$	×
8	$x_0 = 0.5$	×

0 

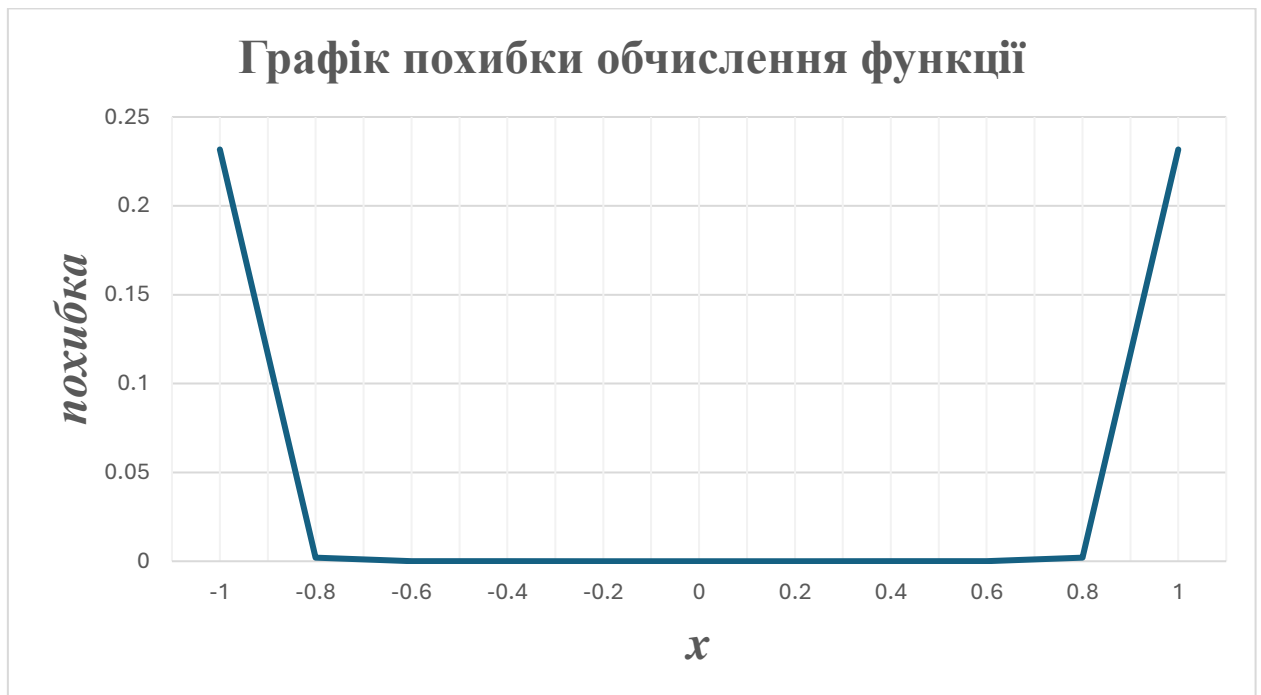
↺

↻

 1

**Графік залежності похибки обчислення заданої функції від значення  $x$  при фіксованому значенні  $n$ :**

**$N=5$**



**Висновок:**

Виконавши цю лабораторну роботу, я засвоїла теоретичний матеріал з теми «Рекурсивні алгоритми», а також набула практичних навичок їх використання. Реалізувала рекурсивні алгоритми трьома способами для знаходження суми перших  $n$  членів ряду чисел, заданого формулою. Для перевірки правильності роботи програм було розроблено ітераційний варіант розв’язку. Отримані результати та калькулятор підтвердили коректність реалізованих алгоритмів. Також, я проаналізувала похибку обчислень. Вона виникає через накопичення округлень при використанні рекурсії. Графік показує залежність похибки обчислення від значення  $X$ , абсолютне значення похибки збільшується при збільшенні абсолютного значення  $X$ .