

**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-32
Король Олександр Володимирович
номер у списку групи: 15

Перевірила:

Молчанова А. А.

Київ 2024

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 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 № 15

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

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

Текст програми

1) програма, що рахує на спуску

```
#include <stdio.h>

double multiply (int i, double x) {

    return x*x*(2*i-1)/(2*i+1);

}

double recursion(double previous, int i, double sum, double x, unsigned int n) {

    double current;

    if (i == 1) {
```

```

        current = x;

    } else {

        current = -previous * multiply(i-1,x);

    }

    sum += current;

    if (i == n) {

        return sum;

    } else {

        return recursion(current,i+1, sum, x, n);

    }

}

double provideRecursion ( double x, int n) {

    return recursion(0,1,0,x,n);

}

int main() {

    double x;

    unsigned int n;

    printf("enter x (hint: it must be |x| < 1) = ");

    scanf("%lf", &x);

    if(x >= 1 || x <=-1) {

        printf("x must be |x| < 1");

    } else {

        printf("enter n = ");

        scanf("%d", &n);

        double result = provideRecursion(x,n);

        printf("Result: %lf\n", result);

    }

    return 0;

}

```

2) програма, що рахує при поверненні

```
#include <stdio.h>

double multiply (int i, double x) {

    return x*x*(2*i-1)/(2*i+1);

}

double recursion(double x, int n, double *sum) {

    double current;

    if (n == 1) {

        current = x;

    } else {

        current = -recursion( x,n-1, sum) * multiply(n-1, x);

    }

    *sum += current;

    return current;

}

double provideRecursion ( double x, int n) {

    double sum = 0;

    recursion(x, n, &sum);

    return sum;

}

int main() {

    double x;

    int n;

    printf("enter x (hint: it must be |x| < 1) = ");

    scanf("%lf", &x);

    if(x >= 1 || x <=-1) {

        printf("x must be |x| < 1");

    } else {
```

```

    printf("enter n = ");

    scanf("%d", &n);

    printf("Result: %.10lf\n", provideRecursion(x, n));

}

return 0;
}

```

3) програма, що рахує в змішаному форматі

```

#include <stdio.h>

double multiply (int i, double x) {

    return x*x*(2*i-1)/(2*i+1);

}

double recursion (double previous, int i, double x, int n) {

    double current;

    if (i == 1) {

        current = x;

    } else {

        current = -previous * multiply(i-1, x);

    }

    if (i == n) {

        return current;

    } else {

        double sum = current + recursion(current, i+1, x, n);

        return sum;

    }

}

double provideRecursion (double parameter, int amount) {

    return recursion(0, 1, parameter, amount);
}

```

```

}

int main() {

    double x;

    int n;

    printf("enter x (hint: it must be |x| < 1) = ");

    scanf("%lf", &x);

    if(x >= 1 || x <=-1) {

        printf("x must be |x| < 1");

    } else {

        printf("enter n = ");

        scanf("%d", &n);

        double sum = provideRecursion(x,n);

        printf("Result: %.10lf\n", sum);

    }

    return 0;

}

```

4) програма, що рахує циклом

```

#include <stdio.h>

double multiply (int i,double x) {

    return x*x*(2*i-1)/(2*i+1);

}

double loop(double x, int amount) {

    double sum = x;

    double current = x;

    for (int i = 2; i <= amount ; i++) {

        current *= -multiply(i-1,x);

        sum += current;

    }

}

```

```

    return sum;
}

int main() {

    double x;

    int n;

    printf("enter x (hint: it must be |x| < 1) = ");

    scanf("%lf", &x);

    if(x >= 1 || x <=-1) {

        printf("x must be |x| < 1");

    } else {

        printf("enter n = ");

        scanf("%d", &n);

        printf("Result: %.10lf\n", loop(x,n));

    }

    return 0;
}

```

Результати тестування програми

1. для $x = 0.5$

1)

```

D:\programming\university\labs\DSA-LABS\second_semester\Lab_1\onDivein\divein.exe
enter x (hint: it must be |x| < 1) =0.5
enter n =5
Result: 0.463684

Process finished with exit code 0

```

2)

```
D:\programming\university\labs\DSA-LABS\second_semester\Lab_1\onBack\back.exe
enter x (hint: it must be |x| < 1) =0.5
enter n =5
Result: 0.4636842758

Process finished with exit code 0
```

3)

```
D:\programming\university\labs\DSA-LABS\second_semester\Lab_1\onMixed\mixed.exe
enter x (hint: it must be |x| < 1) =0.5
enter n =5
Result: 0.4636842758

Process finished with exit code 0
```


4)

```
D:\programming\university\labs\DSA-LABS\second_semester\Lab_1\loop\loop.exe
enter x (hint: it must be |x| < 1) =0.5
enter n =5
Result: 0.4636842758

Process finished with exit code 0
```

калькулятор:

0.5

 **Результат:**

$\arctg(0.5) = 0.4636476090008061$

2. для $x = -0.145$

1)

```
D:\programming\university\labs\DSA-LABS\second_semester\Lab_1\onDivein\divein.exe
enter x (hint: it must be  $|x| < 1$ ) =-0.145
enter n =5
Result: -0.143996

Process finished with exit code 0
```

2)

```
D:\programming\university\labs\DSA-LABS\second_semester\Lab_1\onBack\back.exe
enter x (hint: it must be  $|x| < 1$ ) =-0.145
enter n =5
Result: -0.1439964218

Process finished with exit code 0
```

3)

```
D:\programming\university\labs\DSA-LABS\second_semester\Lab_1\onMixed\mixed.exe
enter x (hint: it must be  $|x| < 1$ ) =-0.145|
enter n =5
Result: -0.1439964218

Process finished with exit code 0
```


4)

```
D:\programming\university\labs\DSA-LABS\second_semester\Lab_1\loop\loop.exe
enter x (hint: it must be  $|x| < 1$ ) =-0.145
enter n =5
Result: -0.1439964218

Process finished with exit code 0
```

калькулятор:

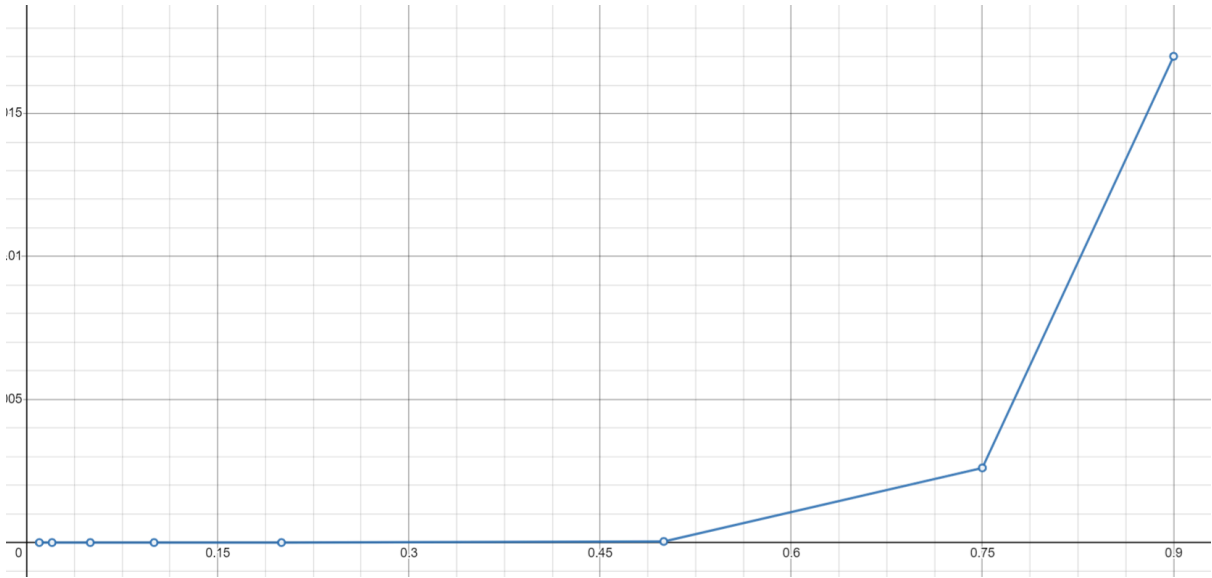
-0.145

 **Результат:**

$$\operatorname{arctg}(-0.145) = -0.14399642170889201$$

Розрахунки похибки

0.01	0.02	0.05	0.1	0.2	0.5	0.75	0.9
11.333476	22.684946						
2114×10^{-1}	3625×10^{-1}	2.1942767	8.8379581	1.8501192	0.0000366	0.0026084	0.01700149
1	1	048×10^{-11}	431×10^{-12}	245×10^{-9}	667991939	4490672	06135



Висновки: На цій лабораторній роботі ми навчилися розробляти програми з використанням рекурсивних функцій. Зокрема, ми вивчили теорію пов'язану з використанням рекурсивних функцій, методику обчислення значень різних функцій за допомогою рекурсії; також у результаті цієї лабораторної роботи ми здобули практичні навички з роботи з рекурсивними функціями та їх застосування на прикладі обчислення складних математичних послідовностей.

Під час лабораторної роботи ми порівняли ефективність трьох підходів: рекурсивного обчислення на спуску, на поверненні та змішаного. За результатами порівняння ми виявили, що підхід на спуску може бути більш ефективним з точки зору часу виконання, особливо для великих значень n , оскільки уникне накопичення додаткових значень на шляху до бази. Крім того, ми здійснили перевірку та тестування розроблених програм з рекурсивними функціями. Це дозволило нам переконатися в правильності реалізації алгоритмів та в їхній здатності до точних обчислень.

У підсумку, ця лабораторна робота дала нам можливість поглибити знання про рекурсивні функції та їх використання в обчисленнях, а також розвинула практичні навички програмування та аналізу алгоритмів.