

**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 №2
з дисципліни
«Алгоритми і структури даних»

Виконав:

студент групи ІП-04
Пашенко Дмитро Олексійович
номер у списку групи: 19

Перевірила:

Сергієнко А. А.

Постановка задачі

Дане натуральне число n . Знайти суму перших n членів ряду чисел, заданого рекурентною формулою. Розв'язати задачу трьома способами (написати три програми): 1) в програмі використати рекурсивну процедуру або функцію, яка виконує обчислення і членів ряду, і суми на рекурсивному спуску; 2) в програмі використати рекурсивну процедуру або функцію, яка виконує обчислення і членів ряду, і суми на рекурсивному поверненні; 3) в програмі використати рекурсивну процедуру або функцію, яка виконує обчислення членів ряду на рекурсивному спуску, а обчислення суми на рекурсивному поверненні.

Програми повинні працювати коректно для довільного натурального n включно з $n = 1$.

Варіант 19

$$F_1 = x; F_{i+1} = -F_i \cdot x^2 (2i-1)^2 / (4i^2 + 2i); i > 0;$$

$$\sum F_i = \operatorname{arcsch} x, |x| < 1;$$

Текст програми 1 (спуск)

```
#include <stdio.h>
#include <math.h>
int N = 1;

double first(double x, double F, int n, double s) {
    double i = -1;
    if (n == -1)
        i = s;
    else {
        s = s + F;
        double t = (2 * (N - n) - 1), p = (4 * (N - n) * (N - n) +
2 * (N - n)), k = -1 * t * t / p;
        F = x * k * x * F;
        i = first(x, F, n - 1, s);
    }
    return i;
}

int main() {
    double S = 0, F, x;
    N = 5;
    x = 0.2;

    if (fabs(x) < 1) {
        F = x;
        S = first(x, F, N - 1, S);
        printf("%.10f\n", S);
    }
    else {
        printf("Module x must be less than 1\n");
    }
}
```

Тестування

```
"C:\C lang\lab
0.1986901108
```

Текст програми 2 (повернення)

```
#include <stdio.h>
#include <math.h>
int N = 1;
double S = 0;

double second(double x, int n) {
    double i;
    if (n == 1) {
        i = x;
        S = S + i;
    }
    else {
        double t = (2 * (n - 1) - 1), p = (4 * (n - 1) * (n - 1) +
2 * (n - 1)), k = -1 * t * t / p;
        i = x * k * x * second(x, n - 1);
        S = S + i;
    }
    return i;
}

int main() {
    double F, x;
    S = 0;
    N = 5;
    x = 0.2;

    if (fabs(x) < 1) {
        F = x;
        x = second(x, N);
        printf("%.10f\n", S);
    }
    else {
        printf("Module x must be less than 1\n");
    }
}
```

Тестування

```
"C:\C lang\lab
0.1986901108
```

Текст програми 3 (спуск + повернення)

```
#include <stdio.h>
#include <math.h>
int N = 1;

double third(double x, double F, int n) {
    double i = -1;
    if (n == -1)
        return x;
    else {
        double t = (2 * (N - n) - 1), p = (4 * (N - n) * (N - n) +
2 * (N - n)), k = -1 * t * t / p;
        F = x * k * x * F;
        return F + third(x, F, n - 1);
    }
}

int main() {
    double S = 0, F, x;
    N = 5;
    x = 0.2;

    if (fabs(x) < 1) {
        F = x;
        S = third(x, x, N - 1);
        printf("%.10f\n", S);
    }
    else {
        printf("Module x must be less than 1\n");
    }
}
```

Тестування

```
"C:\C lang\lab
0.1986901108
```

Текст програми 4 (цикл)

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

int main() {
    int N;
    double S = 0, x, F;
    N = 5;
    x = 0.2;

    if (fabs(x) < 1) {
        for (unsigned int i = 1; i <= N; i++) {
            if (i == 1) {
                F = x;
                S += F;
            } else {
                F *= (-1) * x * x * (2 * (i-1) - 1) * (2 * (i-1) -
1) / (4 * (i-1) * (i-1) + 2 * (i-1));
                S += F;
            }
        }
        printf("%.10f\n", S);
    }
    else {
        printf("Module x must be less than 1\n");
    }
}
```

Тестування

```
"C:\C lang\lab
0.1986901108
```

Тестування (калькулятор)

N = 5, x = 0.2

F1 = 0.2

F2 = -0.0013

$$-0.2 \cdot 0.2 \cdot 0.2 \cdot (2-1)^2 / (4+2)$$

 Extended Keyboard Upload

Input:

$$-0.2 \times 0.2 \times 0.2 \times \frac{(2-1)^2}{4+2}$$

Result:

[illegible]

F3 = 0.0000234

$$0.0013 \cdot 0.2 \cdot 0.2 \cdot (2^2 - 1)^2 / (4 \cdot 2^2 + 2 \cdot 2)$$

Extended Keyboard Upload

Input:



$$0.0013 \times 0.2 \times 0.2 \times \frac{(2 \times 2 - 1)^2}{4 \times 2^2 + 2 \times 2}$$

Result:

0.0000234

$$F4 = -5.571428571428571428571428571428571428571428571428571 \times 10^{-7}$$

$$-0.0000234 \cdot 0.2 \cdot 0.2 \cdot (2 \cdot 3 - 1)^2 / (4 \cdot 3^2 + 2 \cdot 3)$$

 Extended Keyboard  Upload

Input:

$$-0.0000234 \times 0.2 \times 0.2 \times \frac{(2 \times 3 - 1)^2}{4 \times 3^2 + 2 \times 3}$$

Result:

$$-5.571428571428571428571428571428571428571428571428571... \times 10^{-7}$$

[illegible]

$$71428571428571428571428571428571428571428571428571 \times 10^{-7} * 0.2 * 0.2$$

 Sigma Pi

 Extended Keyboard

 Upload

An attempt was made to fix mismatched parentheses, brackets, or braces.

Input interpretation:

$$\frac{5.571428571428571428571428571428571428571428571428571 \times 10^{-7} \times 0.2 \times 0.2 \times \frac{(2 \times 4 - 1)^2}{4 \times 4 \times 4 + 2 \times 4}}{}$$

Result:

[illegible]

sum = ~ 0.1987228580

$571428571 \times 10^{-7} + 1.51666666666666666666666666666666666666666666666$

$\int_{\Sigma} \pi_{\Sigma} \theta$

$\int_{\Sigma^{\pi}}^{\pi}$ Extended Keyboard

 Upload

Exan

Input interpretation:

[illegible]

Result:

0.1987228580238095238095238095238095238095238095238095
...

Графік похибки

