

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 №2.1
з дисципліни
«Алгоритми і структури даних»

Виконала:

студентка групи ІМ-21

Рабійчук Дар'я Олександрівна

номер у списку групи: 18

Перевірила:

Молчанова А.А.

Київ 2023

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 № 18

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

$$\sum F_i = \operatorname{ch} x, |x| < 10^6;$$

Текст програм:

№1

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

int main() {

    double x;
    unsigned int n = 0;

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

    printf("n: ");
    scanf("%u", &n);

    clock_t begin = clock();

    double recursion(double x, double F, double sum, unsigned int i){

        F = i == 0 ? x : F * ((x * x) / (4 * i * i - 2 * i));
        sum += F;
        i++;

        return i < n ? recursion(x, F, sum, i) : sum;

    }

    double sum = recursion(x, 0, 0, 0);
    printf("Sum = %.15lf\n", sum);
    printf("Delta: %.15lf\n", fabs((double) (cosh(x) - sum)));

    clock_t end = clock();
    printf("Time: %lf seconds\n", (double) (end - begin) / CLOCKS_PER_SEC);

    return 0;
}
```

№2

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

int main() {

    double x;
    unsigned int n = 0;

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

    printf("n: ");
    scanf("%u", &n);

    clock_t begin = clock();

    double sum = 0;
    double recursion(double x, unsigned int i){
        double F;

        if (i == 0){
            F = 1;
        } else {
            double m = (x * x) / (4 * i * i - 2 * i);
            F = m * recursion(x, i - 1);
        }
        sum += F;
        return F;
    }

    double F = recursion(x, n);

    printf("Sum = %.15lf\n", sum);
    printf("Delta: %.15lf\n", fabs((double)(cosh(x) - sum)));

    clock_t end = clock();
    printf("Time: %lf seconds\n", (double)(end - begin) / CLOCKS_PER_SEC);

    return 0;
}
```

№3

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

int main() {

    double x;
    unsigned int n = 0;

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

    printf("n: ");
    scanf("%u", &n);

    clock_t begin = clock();

    double recursion(double x, double F, unsigned int i){

        F = i == 0 ? x : F * ((x * x) / (4 * i * i - 2 * i));
        i++;
        return i < n ? F + recursion(x, F, i) : F;
    }

    double sum = recursion(x, 0, 0);

    printf("Sum = %.15lf\n", sum);
    printf("Delta: %.15lf\n", fabs((double)(cosh(x) - sum)));

    clock_t end = clock();
    printf("Time: %lf seconds\n", (double)(end - begin) / CLOCKS_PER_SEC);

    return 0;
}
```

№4 (циклічна для перевірки)

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

int main(){

    double x;
    unsigned int n = 0;

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

    printf("n: ");
    scanf("%u", &n);

    clock_t begin = clock();

    double F = 1;
    double sum = F;
    for (int i = 1; i < n; i++){
        F = F * ((x * x) / (4 * i * i - 2 * i));
        sum += F;
    }

    printf("Sum = %.15lf\n", sum);
    printf("Delta: %.15lf\n", fabs((double)(cosh(x) - sum)));

    clock_t end = clock();
    printf("Time: %lf seconds\n", (double)(end - begin) / CLOCKS_PER_SEC);

    return 0;
}
```

Результат циклічної програми:

```
x: 2
n: 5
Sum = 3.761904761904762
Delta: 0.000290929178870
Time: 0.048000 seconds

Process returned 0 (0x0)   execution time : 6.005 s
Press any key to continue.
```

```
x: 4
n: 1
Sum = 1.0000000000000000
Delta: 26.308232836016487
Time: 0.000000 seconds

Process returned 0 (0x0)   execution time : 1.172 s
Press any key to continue.
```

```
x: 1
n: 1
Sum = 1.0000000000000000
Delta: 0.543080634815244
Time: 0.000000 seconds

Process returned 0 (0x0)   execution time : 0.949 s
Press any key to continue.
```

Результат програми виконаної способом №1:

```
x: 2
n: 5
Sum = 3.761904761904762
Delta: 0.000290929178870
Time: 0.001000 seconds

Process returned 0 (0x0)   execution time : 2.109 s
Press any key to continue.
```

```
x: 4
n: 1
Sum = 1.0000000000000000
Delta: 26.308232836016487
Time: 0.001000 seconds

Process returned 0 (0x0)   execution time : 1.746 s
Press any key to continue.
```

```
x: 1
n: 1
Sum = 1.0000000000000000
Delta: 0.543080634815244
Time: 0.001000 seconds

Process returned 0 (0x0)   execution time : 1.016 s
Press any key to continue.
```

Результат програми виконаної способом №2:

```
x: 2
n: 5
Sum = 3.761904761904762
Delta: 0.000290929178870
Time: 0.000000 seconds

Process returned 0 (0x0)   execution time : 1.898 s
Press any key to continue.
```

```
x: 4
n: 1
Sum = 1.0000000000000000
Delta: 26.308232836016487
Time: 0.001000 seconds

Process returned 0 (0x0)   execution time : 1.167 s
Press any key to continue.
```

```
x: 1
n: 1
Sum = 1.0000000000000000
Delta: 0.543080634815244
Time: 0.001000 seconds

Process returned 0 (0x0)   execution time : 6.328 s
Press any key to continue.
```

Результат програми виконаної способом №3:

```
x: 2
n: 5
Sum = 3.761904761904762
Delta: 0.000290929178870
Time: 0.001000 seconds

Process returned 0 (0x0)   execution time : 2.370 s
Press any key to continue.
```

```
x: 4
n: 1
Sum = 1.0000000000000000
Delta: 26.308232836016487
Time: 0.001000 seconds

Process returned 0 (0x0)   execution time : 2.580 s
Press any key to continue.
```

```
x: 1
n: 1
Sum = 1.0000000000000000
Delta: 0.543080634815244
Time: 0.000000 seconds

Process returned 0 (0x0)   execution time : 0.692 s
Press any key to continue.
```

Результат обчислень на калькуляторі:

x

5

Точность вычисления

Знаков после запятой: 20

РАССЧИТАТЬ

Значения гиперболических функций

↓

| Обозначение | Наименование | Значение |
|-------------|-------------------------|-------------------|
| sh | Гиперболический синус | 74.20321057778874 |
| ch | Гиперболический косинус | 74.20994852478783 |

x

4

Точность вычисления

Знаков после запятой: 20

РАССЧИТАТЬ

Значения гиперболических функций

↓

| Обозначение | Наименование | Значение |
|-------------|-------------------------|--------------------|
| sh | Гиперболический синус | 27.289917197127746 |
| ch | Гиперболический косинус | 27.308232836016483 |

x

1

Точность вычисления

Знаков после запятой: 20

РАССЧИТАТЬ

Значения гиперболических функций

↓

| Обозначение | Наименование | Значение |
|-------------|-------------------------|--------------------|
| sh | Гиперболический синус | 1.1752011936438014 |
| ch | Гиперболический косинус | 1.5430806348152437 |