

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-12
Миць Вікторія Ігорівна
Номер у списку групи: 19

Перевірила:
Молчанова А. А

Київ 2022

Завдання:

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 i членів ряду, i суми на рекурсивному спуску; 2) в програмі використати рекурсивну процедуру або функцію, яка виконує обчислення i членів ряду, i 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:

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

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

Код алгоритму на C:

1) обчислення і членів ряду, і суми на рекурсивному спуску

```
#include <stdio.h>

double x;
int n;

double get_multiplier(int n)
{
    return - x * x * (2 * n - 1) * (2 * n - 1) / (4 * n * n + 2 * n);
}

double recursion_sum(double last, int current, double sum)
{
    double element = (current != 1) ? last * get_multiplier(current - 1) : x;
    sum += element;
    if (current == n)
    {
        return sum;
    }
    else
    {
        return recursion_sum(element, current + 1, sum);
    }
}

double get_sum(double argument, int count)
{
    x = argument;
    n = count;
    return recursion_sum(0, 1, 0);
}

int main()
{
    double argument = -1;
    while (argument >= 1 || argument <= -1)
    {
        printf("Input real number x, |x| < 1\n");
        scanf("%lf", &argument);
    }

    unsigned int count;
    printf("Input integer n\n");
    scanf("%d", &count);

    double sum = get_sum(argument, count);

    printf("The result is %.30lf", sum);

    return 0;
}
```

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

```
#include <stdio.h>

double x;

double get_multiplier(int n)
{
    return - x * x * (2 * n - 1) * (2 * n - 1) / (4 * n * n + 2 * n);
}

double recursion_sum(int n, double *sum)
{
    if (n == 1)
    {
        *sum += x;
        return x;
    }
    double element = recursion_sum(n - 1, sum) * get_multiplier(n - 1);
    *sum += element;
    return element;
}

void get_sum(double argument, int count, double *sum)
{
    *sum = 0;
    x = argument;
    recursion_sum(count, sum);
}

int main()
{
    double argument = -1;
    while (argument >= 1 || argument <= -1)
    {
        printf("Input real number x, |x| < 1\n");
        scanf("%lf", &argument);
    }

    unsigned int count;
    printf("Input integer n\n");
    scanf("%d", &count);

    double sum;
    get_sum(argument, count, &sum);

    printf("The result is %.30lf", sum);

    return 0;
}
```

- 3) обчислення членів ряду на рекурсивному спуску, а обчислення суми на рекурсивному поверненні

```
#include <stdio.h>

double x;
int n;

double get_multiplier(int n)
{
    return - x * x * (2 * n - 1) * (2 * n - 1) / (4 * n * n + 2 * n);
}

double recursion_sum(double last, int cur)
{
    double element = (cur != 1) ? last * get_multiplier(cur - 1) : x;
    if (cur == n)
    {
        return element;
    }
    else
    {
        double sum = element + recursion_sum(element, cur + 1);
        return sum;
    }
}

double get_sum(double argument, int count)
{
    x = argument;
    n = count;
    return recursion_sum(0, 1);
}

int main()
{
    double argument = -1;
    while (argument >= 1 || argument <= -1)
    {
        printf("Input real number x, |x| < 1\n");
        scanf("%lf", &argument);
    }

    unsigned int count;
    printf("Input integer n\n");
    scanf("%d", &count);

    double sum = get_sum(argument, count);

    printf("The result is %.30lf", sum);

    return 0;
}
```

4) циклічний алгоритм для тестування

```
#include <stdio.h>

double x;

double get_multiplier(int n)
{
    return - x * x * (2 * n - 1) * (2 * n - 1) / (4 * n * n + 2 * n);
}

double get_sum(double argument, int count)
{
    x = argument;

    double sum = argument;
    double element = argument;
    for (int i = 2; i <= count; i++) {
        element *= get_multiplier(i - 1);
        sum += element;
    }
    return sum;
}

int main()
{
    double argument = -1;
    while (argument >= 1 || argument <= -1)
    {
        printf("Input real number x, |x| < 1\n");
        scanf("%lf", &argument);
    }

    unsigned int count;
    printf("Input integer n\n");
    scanf("%d", &count);

    double sum = get_sum(argument, count);

    printf("The result is %.30lf", sum);

    return 0;
}
```

Результати тестувань:

1) Значення x : 0.875

Бажаний результат: $\operatorname{arcsinh}(0.875)$

Input		
$\sinh^{-1}(0.875)$	$\sinh^{-1}(x)$ is the inverse hyperbolic sine function	
Result	Fewer digits	More digits
0.7901687254123969767786401184854033556023470437228822942219953873 ...		

a. Значення n : 5

Алгоритм 1:

```
vika@MacBook-Pro-vika code % ./first
Input real number x, |x| < 1
0.875
Input integer n
5
The result is 0.793418025816532090921384678950%
```

Алгоритм 2:

```
vika@MacBook-Pro-vika code % ./second
Input real number x, |x| < 1
0.875
Input integer n
5
The result is 0.793418025816532090921384678950%
```

Алгоритм 3:

```
vika@MacBook-Pro-vika code % ./third
Input real number x, |x| < 1
0.875
Input integer n
5
The result is 0.793418025816532090921384678950%
```

Циклічний алгоритм:

```
vika@MacBook-Pro-vika code % ./cyclic
Input real number x, |x| < 1
0.875
Input integer n
5
The result is 0.793418025816532090921384678950%
```

На калькуляторі:

The calculator interface shows a sequence of 7 steps:

- Step 1: $t = 0.875$. A slider below the input ranges from -10 to 10, with the value set at 0.875.
- Step 2: $f(x, i) = -\frac{x \cdot t^2 \cdot (2i - 1)^2}{(4i^2 + 2i)}$
- Step 3: $F_2 = f(0.875, 1)$. The result is $F_2 = -0.111653645833$.
- Step 4: $F_3 = f(F_2, 2)$. The result is $F_3 = 0.038468170166$.
- Step 5: $F_4 = f(F_3, 3)$. The result is $F_4 = -0.0175310671329$.
- Step 6: $F_5 = f(F_4, 4)$. The result is $F_5 = 0.0091345686168$.
- Step 7: $S = t + F_2 + F_3 + F_4 + F_5$. The result is $S = 0.793418025817$.

b. Значення n: 50

Алгоритм 1:

```
vika@MacBook-Pro-vika code % ./first
Input real number x, |x| < 1
0.875
Input integer n
50
The result is 0.790168724784345410938612985774%
```

Алгоритм 2:


```
vika@MacBook-Pro-vika code % ./second
Input real number x, |x| < 1
0.875
Input integer n
50
The result is 0.790168724784345410938612985774%
```

Алгоритм 3:

```
vika@MacBook-Pro-vika code % ./third
Input real number x, |x| < 1
0.875
Input integer n
50
The result is 0.790168724784345410938612985774%
```

Циклічний алгоритм:

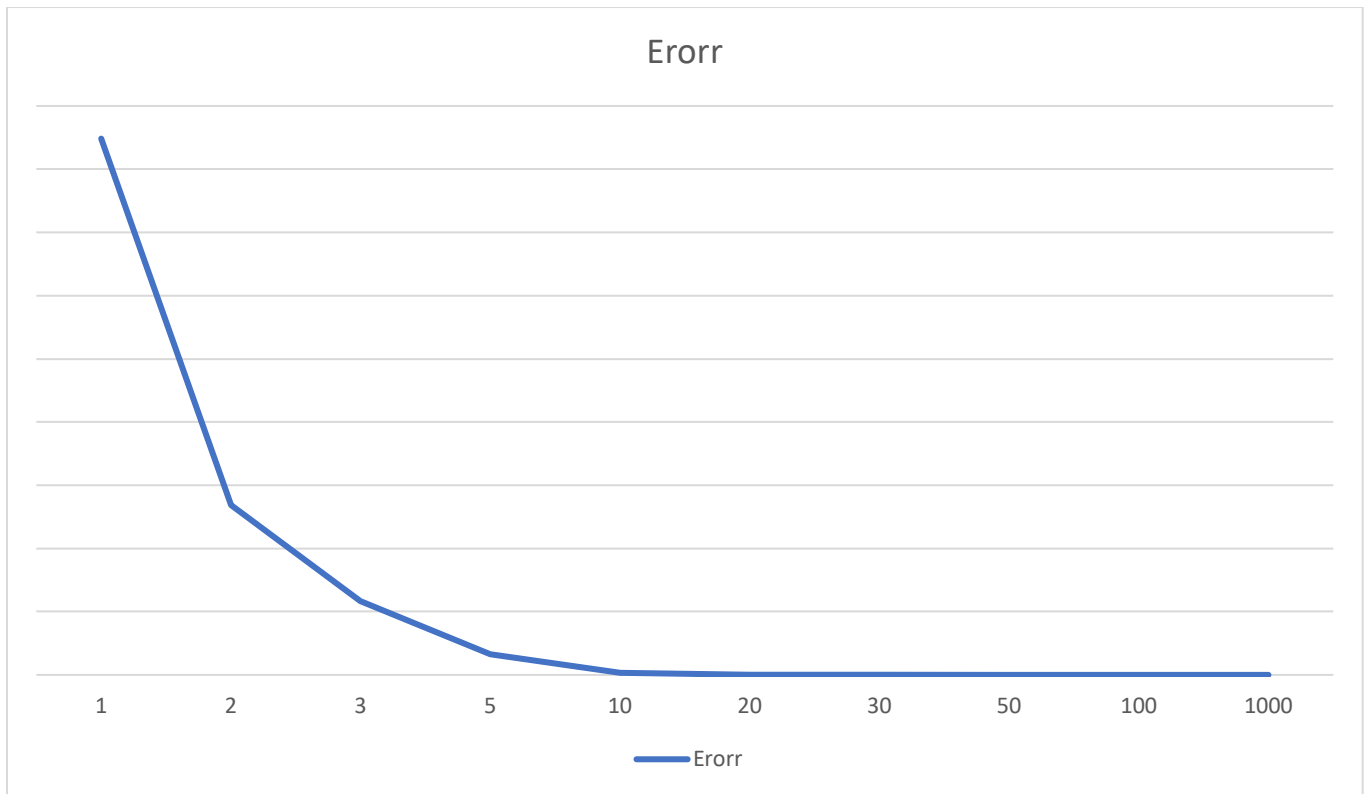
```
vika@MacBook-Pro-vika code % ./cyclic
Input real number x, |x| < 1
0.875
Input integer n
50
The result is 0.790168724784345410938612985774%
```

Похибка

$|\text{result} - \text{arcsh}(0.875)|$ (не вираховуємо ділення на $\text{arcsh}(0.875)$, оскільки при побудові графіка ним можна знехтувати)

1	2	3	5	10	20	30	50	100	1000
0.08483	0.02682	0.01164	0.00324	0.000305	7.48744	2.82131	6.28052	3.33067	2.22045
13	24	58	93	349	e-06	e-07	e-10	e-16	e-16

$\text{arcsh}(0.875)$



2) Значення x : -0.32
 Бажаний результат: $\operatorname{arcsinh}(-0.32)$

Input

$\sinh^{-1}(-0.32)$

$\sinh^{-1}(x)$ is the inverse hyperbolic sine function

Result

-0.314775980018790209460221332273736193181029013834072211758411467
...

Fewer digits
More digits

а. Значення n : 5

Алгоритм 1:

```
vika@MacBook-Pro-vika code % ./first
Input real number x, |x| < 1
-0.32
Input integer n
5
The result is -0.314776054707390562370505904255%
```

Алгоритм 2:

```
vika@MacBook-Pro-vika code % ./second
Input real number x, |x| < 1
-0.32
Input integer n
5
The result is -0.314776054707390562370505904255%
```

Алгоритм 3:

```
vika@MacBook-Pro-vika code % ./third
Input real number x, |x| < 1
-0.32
Input integer n
5
The result is -0.314776054707390562370505904255%
```

Циклічний алгоритм:

```
vika@MacBook-Pro-vika code % ./cyclic
Input real number x, |x| < 1
-0.32
Input integer n
5
The result is -0.314776054707390562370505904255%
```

На калькуляторі:

The calculator interface displays a sequence of 7 steps for calculating the sum S :

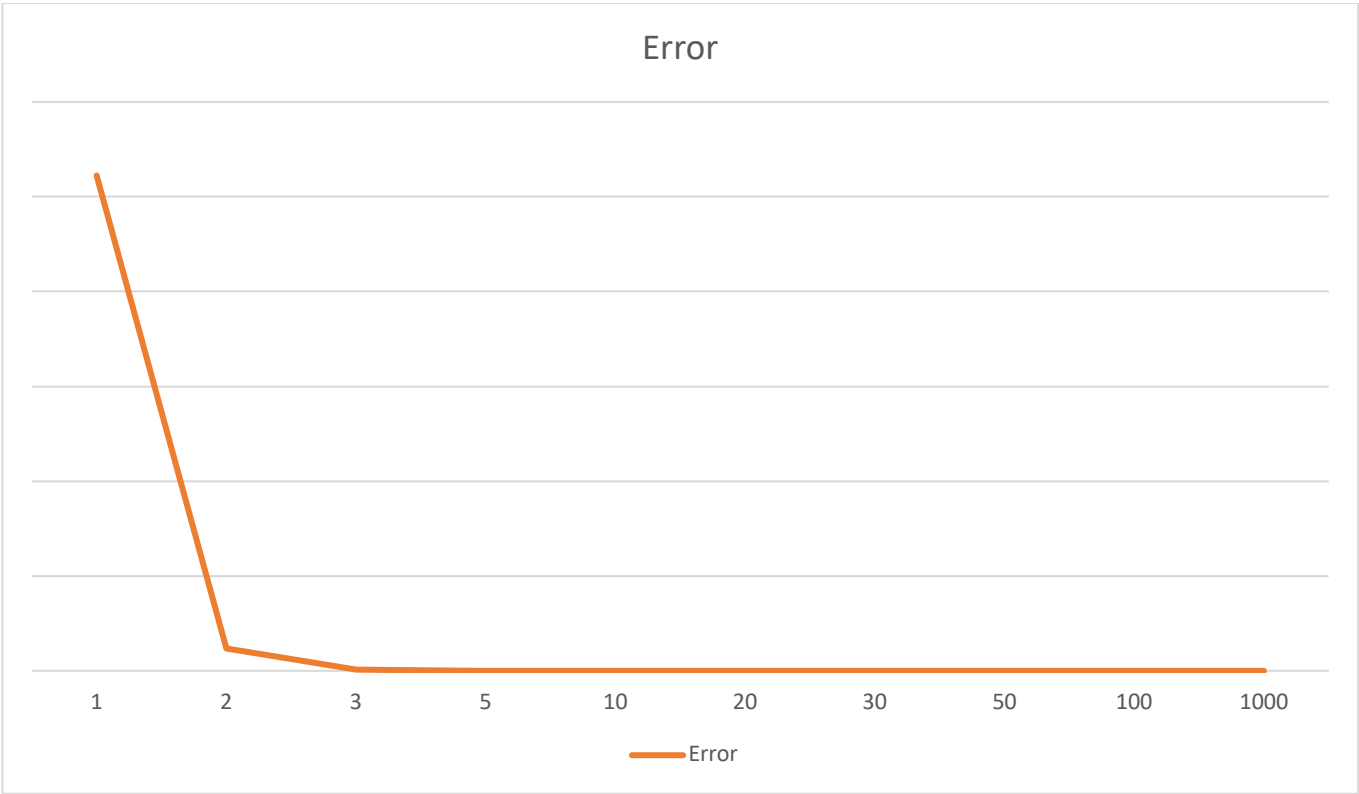
- Step 1: Variable assignment $t = -0.32$. A slider below the input ranges from -10 to 10.
- Step 2: Formula for $f(x, i)$:
$$f(x, i) = -\frac{x \cdot t^2 \cdot (2i - 1)^2}{(4i^2 + 2i)}$$
- Step 3: Calculation of $F_2 = f(-0.32, 1)$. The result is $F_2 = 0.00546133333333$.
- Step 4: Calculation of $F_3 = f(F_2, 2)$. The result is $F_3 = -0.00025165824$.
- Step 5: Calculation of $F_4 = f(F_3, 3)$. The result is $F_4 = 0.0000153391689143$.
- Step 6: Calculation of $F_5 = f(F_4, 4)$. The result is $F_5 = -0.00000106896963812$.
- Step 7: Final sum calculation $S = t + F_2 + F_3 + F_4 + F_5$. The result is $S = -0.314776054707$.

Похибка

$|result - \operatorname{arcsh}(-0.32)|$ (не вираховуємо ділення на $\operatorname{arcsh}(-0.32)$, оскільки при побудові графіка ним можна знехтувати)

1	2	3	5	10	20	30	50	100	1000
0.00522402	0.000237313	1.43449e-05	7.46886e-08	3.12417e-13	~0	~0	~0	~0	~0

$\operatorname{arcsh}(-0.32)$



Залежність похибки від x

