

**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 ІМ-43

Олексійчук Станіслав Юрійович

номер у списку групи: 23

Перевірила:

Молчанова А. А.

Київ 2024

# ***Постановка задачі***

1. Задане натуральне число  $n$ . Вирахувати значення заданої формули за варіантом.
2. Для вирішення задачі написати дві програми:
  - 1) перша програма повинна використовувати для обчислення формули вкладені цикли;
  - 2) друга програма повинна виконати обчислення формули за допомогою одного циклу з використанням методу динамічного програмування.
3. Виконати розрахунок кількості операцій для кожного з алгоритмів за методикою, викладеною на лекції, додавши до неї підрахунок кількості викликів стандартних функцій.
4. Програма має правильно вирішувати поставлену задачу при будь-якому заданому  $n$ , для якого результат обчислення може бути коректно представлений типом `double`.
5. Результуючі дані вивести у форматі з сімома знаками після крапки.

## ***Варіант 23:***

$$S = \sum_{i=1}^n \frac{\prod_{j=1}^i ((j+1) \cdot \sin(j))}{i \cdot (i+1)}$$

# *Тексти програм*

## **1) 1<sup>st</sup> task (not dynamic)**

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

int main()
{
    double S = 0.0;
    double P = 1.0;
    int n;
    printf("Input n: ");
    scanf("%d",&n);
    if (n>=0){
        int i = 1;
        int j = 1;
        while (i<=n){
            while (j<=i) {
                P *= (j+1)*sin(j);
                j++;
            }
            S+=P/(i*(i+1));
            i++;
        }
        printf("The sum is %.7f\n", S);
    }
    else {
        printf("Error...\n");
    }
}
```

```
    return 0;
}
```

## 2) 2<sup>nd</sup> task (dynamic)

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

int main()
{
    double S = 0.0;
    double P = 1.0;
    int n;
    printf("Input n: ");
    scanf("%d",&n);
    if (n>=0){
        int i = 1;
        while (i<=n) {
            P *= (i+1)*sin(i);
            S+=P/(i*(i+1));
            i++;
        }
        printf("The sum is %.7f\n", S);
    }
    else {
        printf("Error...");
    }
    return 0;
}
```

# ***Розрахунки кількості операцій***

Для зручності та правильності обчислень операцій було написано програму для двох варіантів розв'язку виразу:

## **1) 1<sup>st</sup> task\_operations (not dynamic)**

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

int main()
{
    double S = 0.0;
    double P = 1.0;
    int n;
    int ct = 0;
    int sin_ct = 0;
    int operations = 0;
    printf("Input n: ");
    scanf("%d",&n);
    if (n>=0){
        int i = 1;
        while (i<=n){
            int j = 1;
            while (j<=i) {
                P *= (j+1)*sin(j);
                j++;
                ct += 6;
                sin_ct++;
            }
            i++;
        }
    }
}
```

```

        S+=P/(i*(i+1));

        i++;

        ct += 8;

    }

    operations = ct + sin_ct + 3;

    printf("Operations = %d",operations);

}

else {

    printf("Error...\n");

}

return 0;

}

```

## 2) 2<sup>nd</sup> task\_operations (dynamic)

```

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

int main()

{

    double S = 0.0;

    double P = 1.0;

    int n;

    int ct = 0;

    int sin_ct = 0;

    int operations = 0;

    printf("Input n: ");

    scanf("%d",&n);

    if (n>=0){

        int i = 1;

        while (i<=n) {

```

```

        P *= (i+1)*sin(i);
        S+=P/(i*(i+1));
        i++;
        ct+=10;
        sin_ct+=1;
    }

    operations = ct + sin_ct + 3;
    printf("Operations = %d",operations);
}

else {
    printf("Error...");
}

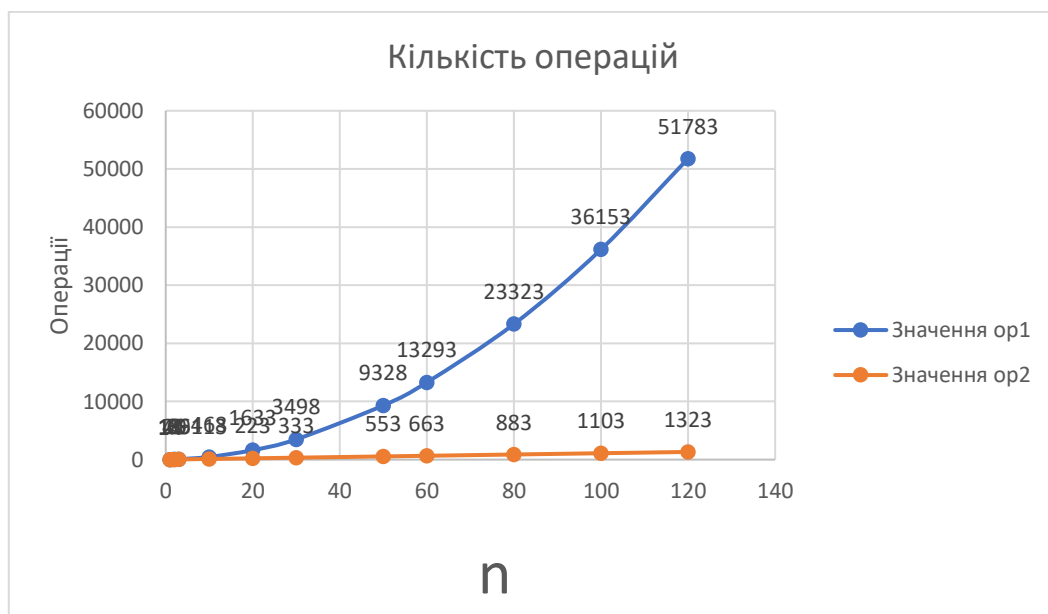
return 0;
}

```

Виберемо деякі значення n, за яких обчислимо кількість операцій для кожного способу. Занесемо ці дані в таблицю:

n		1	2	3	10	20	30	50	60	80	100	120
Кількість операцій	1 спосіб	18	40	69	468	1633	3498	9328	13293	23323	36153	51783
	2 спосіб	14	25	36	113	223	333	553	663	883	1103	1323

Відобразимо ці дані у вигляді графіку:



Ось перевірка кількості операцій з допомогою програми:

1) 

```
Input n: 1
Operations = 18
PS C:\Users\Dell\Desktop> .\1st_task.exe
Input n: 2
Operations = 40
PS C:\Users\Dell\Desktop> .\1st_task.exe
Input n: 3
Operations = 69
PS C:\Users\Dell\Desktop> .\1st_task.exe
Input n: 10
Operations = 468
PS C:\Users\Dell\Desktop> .\1st_task.exe
Input n: 20
Operations = 1633
PS C:\Users\Dell\Desktop> .\1st_task.exe
Input n: 30
Operations = 3498
```

```
Input n: 50
Operations = 9328
PS C:\Users\Dell\Desktop> .\1st_task.exe
Input n: 60
Operations = 13293
PS C:\Users\Dell\Desktop> .\1st_task.exe
Input n: 80
Operations = 23323
PS C:\Users\Dell\Desktop> .\1st_task.exe
Input n: 100
Operations = 36153
PS C:\Users\Dell\Desktop> .\1st_task.exe
Input n: 120
Operations = 51783
PS C:\Users\Dell\Desktop> .\1st_task.exe
```

2) 

```
Input n: 1
Operations = 14
PS C:\Users\Dell\Desktop> .\2nd_task.exe
Input n: 2
Operations = 25
PS C:\Users\Dell\Desktop> .\2nd_task.exe
Input n: 3
Operations = 36
PS C:\Users\Dell\Desktop> .\2nd_task.exe
Input n: 10
Operations = 113
PS C:\Users\Dell\Desktop> .\2nd_task.exe
Input n: 20
Operations = 223
PS C:\Users\Dell\Desktop> .\2nd_task.exe
Input n: 30
Operations = 333
```

```
Input n: 50
Operations = 553
PS C:\Users\Dell\Desktop> .\2nd_task.exe
Input n: 60
Operations = 663
PS C:\Users\Dell\Desktop> .\2nd_task.exe
Input n: 80
Operations = 883
PS C:\Users\Dell\Desktop> .\2nd_task.exe
Input n: 100
Operations = 1103
PS C:\Users\Dell\Desktop> .\2nd_task.exe
Input n: 120
Operations = 1323
PS C:\Users\Dell\Desktop> .\2nd_task.exe
```



# Обчислення виразу

## 1) Скорочення виразу без використання калькулятора

$$S = \sum_{i=1}^n \frac{\prod_{j=1}^i ((j+1) \cdot \sin(j))}{i \cdot (i+1)}$$

Нехай  $n=3$ , тоді

$$\begin{aligned} S &= \sum_{i=1}^3 \frac{\prod_{j=1}^i ((j+1) \cdot \sin(j))}{i \cdot (i+1)} = \frac{((1+1) \cdot \sin(1))}{1 \cdot (1+1)} + \frac{((1+1) \cdot \sin(1)) \cdot ((2+1) \cdot \sin(2))}{2 \cdot (2+1)} + \\ &+ \frac{((1+1) \cdot \sin(1)) \cdot ((2+1) \cdot \sin(2)) \cdot ((3+1) \cdot \sin(3))}{3 \cdot (3+1)} = \frac{2 \sin(1)}{2} + \frac{2 \sin(1) \cdot 3 \sin(2)}{6} + \\ &+ \frac{2 \sin(1) \cdot 3 \sin(2) \cdot 4 \sin(3)}{12} = \frac{12 \sin(1) + 12 \sin(1) \cdot \sin(2) + 24 \sin(1) \cdot \sin(2) \cdot \sin(3)}{12} = \\ &= \frac{12 \sin(1) (1 + \sin(2) + 2 \sin(2) \sin(3))}{12} = \sin(1) (1 + \sin(2) + 2 \sin(2) \sin(3)) = 1,8225736 \end{aligned}$$

## 2) Рішення виразу з використанням графічного калькулятора

[Desmos](#):

The screenshot shows the Desmos calculator interface. In the first input field, the formula 
$$\sum_{i=1}^n \frac{\prod_{j=1}^i ((j+1) \sin(j))}{i(i+1)}$$
 is entered. To the right of the formula, the result is displayed as  $= 1.8225736009$ . In the second input field, the variable  $n$  is set to 3. Below this, there is a slider for  $n$  ranging from -10 to 10, with a blue dot indicating the current value of 3. The interface includes standard calculator controls like a play button, a refresh button, and a close button (X) for each input field.

# Результати тестування програми

1)

```
PS C:\Users\Dell\Desktop>
Input n: 1
The sum is 0.8414710
PS C:\Users\Dell\Desktop>
Input n: 2
The sum is 1.6066184
PS C:\Users\Dell\Desktop>
Input n: 3
The sum is 1.8225736
PS C:\Users\Dell\Desktop>
Input n: 5
The sum is 3.2129339
PS C:\Users\Dell\Desktop>
Input n: 10
The sum is 839.8907095
PS C:\Users\Dell\Desktop>
Input n: -10
Error...
```

$$\sum_{i=1}^n \frac{\prod_{j=1}^i ((j+1) \cdot \sin(j))}{i \cdot (i+1)}$$

$n = 1$  = 0.841470984808

$$\sum_{i=1}^n \frac{\prod_{j=1}^i ((j+1) \cdot \sin(j))}{i \cdot (i+1)}$$

$n = 3$  = 1.8225736009

$$\sum_{i=1}^n \frac{\prod_{j=1}^i ((j+1) \cdot \sin(j))}{i \cdot (i+1)}$$

$n = 2$  = 1.60661838604

$$\sum_{i=1}^n \frac{\prod_{j=1}^i ((j+1) \cdot \sin(j))}{i \cdot (i+1)}$$

$n = 5$  = 3.21293385659

$$\sum_{i=1}^n \frac{\prod_{j=1}^i ((j+1) \cdot \sin(j))}{i \cdot (i+1)}$$

$n = 10$  = 839.890709525

2)

Оскільки  $n=-10$  не входить у проміжок значень натуральних чисел, то немає сенсу його перевіряти, оскільки на калькуляторі покаже 0, а в програмі від'ємні числа виведуть помилку «Error...»

**Висновок:** я навчився використовувати цикли та вкладені цикли для розв'язування суми та добутку певних операцій у формулі, використовуючи середовище CodeBlocks; зрозумів, як використовувати динамічне програмування для оптимізації програми та відповідно зменшення кількості операцій програми, внаслідок чого зрозумів його перевагу; навчився рахувати кількість операцій у програмі, включаючи jump; зрозумів, що є операцією, а що – ні (що потрібно враховувати під час підрахунку кількості операцій, а що – ні); навчився виводити значення (представляти output) у вигляді числа з точністю до певної кількості знаків після коми (у цій роботі – до 7).