

**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 №1.2**  
з дисципліни  
«Алгоритми і структури даних»

Виконала:

Студентка групи ІМ-41  
Куц Анна Василівна номер  
у списку групи: 11

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. Результируючі дані вивести у форматі з сімома знаками після крапки.

## Завдання

11.	$S = \sum_{i=1}^n \frac{i + \cos(i)}{\prod_{j=1}^i (j - \sin(j))}$
-----	--

## Текст програми

Програма 1

```
#include <stdio.h>
#include <math.h>
int main ()
{
    int n, i, j, ct;
    ct = 0;
    double S,P;
    S = 0;
```

```

printf("Enter n: \n");
scanf("%d", &n);
ct++; // S=0
if (n > 0)
{
    for (i = 1; i <= n; i++)
    {
        P = 1;
        for (j = 1; j <= i; j++)
        {
            P *= j - sin(j);
            ct+=6; // <=, j++, *=, -, sin(j), jmp
        }
        S += (i + cos(i)) / P;
        ct+=9; // <=, ++, +=, +, /, cos(i), P=1, j=1, jmp
    }
    printf("Your result = %.7f\n", S);
    ct++; // i=1
}
else
{
    printf("No value for n\n");
}
printf("Operations = %d\n", ct);

return 0;
}

```

## Програма 2

```

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

int main ()
{
    int n, i, ct;
    ct = 0;
    double S,P;
    S = 0;
    P = 1;
    printf("Enter n: \n");
    scanf("%d", &n);

```

```

ct+=2; // S=0, P=1
if (n > 0)
{
    for (i = 1; i <= n; i++)
    {
        P *= i - sin(i);
        S += (i + cos(i)) / P;
        ct+= 10; // <=, ++, *=, -, sin(i), +=, +, cos(i), / , jmp
    }
    printf("Your result = %.7f\n", S);
    ct++; // i=1
}
else
{
    printf("No value for n\n");
}
printf("Operations = %d\n", ct);

return 0;
}

```

### Тестування програми

$$\sum_{i=1}^3 \frac{i + \cos(i)}{\prod_{j=1}^i (j - \sin(j))} = \frac{1 + \cos(1) + \frac{2 + \cos(2)}{2 - \sin(2)} + \frac{3 + \cos(3)}{(\sin(2) - 2)(\sin(3) - 3)}}{1 - \sin(1)}$$

Sum

Exact

$$\sum_{i=1}^3 \frac{i + \cos(i)}{\prod_{j=1}^i (j - \sin(j))} \approx 22.94248907191821204396013636460064763665$$

Sum

$$\sum_{i=1}^2 \frac{i + \cos(i)}{\prod_{j=1}^i (j - \sin(j))} = \frac{4 - \sin(2) + \cos(2) - (\sin(2) - 2) \cos(1)}{(\sin(1) - 1)(\sin(2) - 2)}$$

Decimal approximation

18.876308565589008676529276027286014819141850328079460896279883995

Sum

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

Decimal approximation

9.7162169587795678759322497886811704955289894747331686394715058989  
...

## Програма 1

```
"/Users/annkuts/University/АСД/Лабораторні/lab 1.2/lab 1.2.1"
Enter n:
3
Your result = 22.9424891
Operations = 65

Process finished with exit code 0
```

```
"/Users/annkuts/University/АСД/Лабораторні/lab 1.2/lab 1.2.1"
Enter n:
2
Your result = 18.8763086
Operations = 38

Process finished with exit code 0
```

```
"/Users/annkuts/University/АСД/Лабораторні/lab 1.2/lab 1.2.1"
Enter n:
1
Your result = 9.7162170
Operations = 17

Process finished with exit code 0
```

## Програма 2

```
"/Users/annkuts/University/АСД/Лабораторні/lab 1.2/lab.1.2.2"
Enter n:
3
Your result = 22.9424891
Operations = 33

Process finished with exit code 0
|
```

```
"/Users/annkuts/University/АСД/Лабораторні/lab 1.2/lab.1.2.2"
Enter n:
2
Your result = 18.8763086
Operations = 23

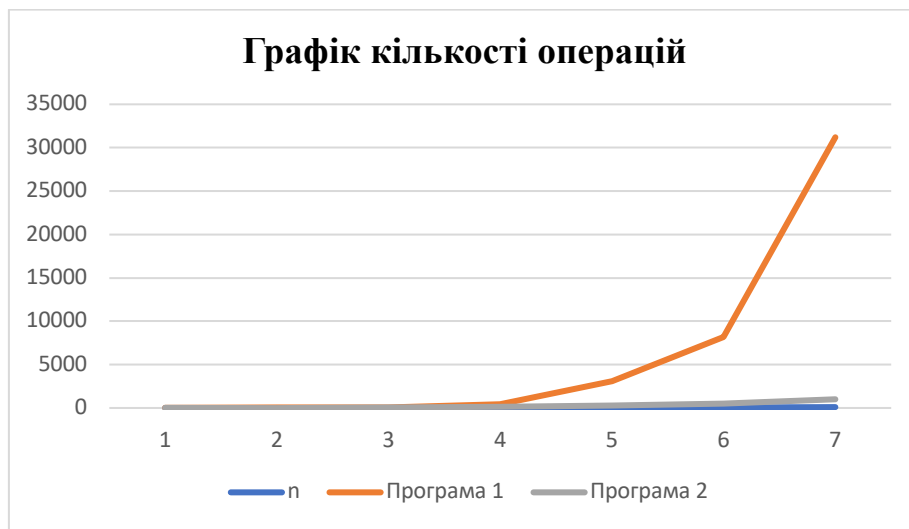
Process finished with exit code 0
|
```

```
"/Users/annkuts/University/АСД/Лабораторні/lab 1.2/lab.1.2.2"
Enter n:
1
Your result = 9.7162170
Operations = 13

Process finished with exit code 0
```

**Таблиця порівняння кількості операцій**

n	1	2	3	10	30	50	100
Програма 1	17	38	65	422	3062	8182	31202
Програма 2	13	23	33	103	303	503	1003



## Висновок

Завдяки виконанню лабораторної роботи №1.2 я засвоїла теоретичний матеріал та набула практичних навичок використання різних циклічних керуючих конструкцій, вкладених циклів, методу динамічного програмування та обчислення кількості операцій алгоритмів. Було створено дві програми для обчислення завдання: перша - з допомогою вкладених циклів, друга – за методом динамічного програмування. Проаналізувавши графік кількості операцій можна зрозуміти, що виконання другої програми є більш оптимізованим та зручним.