

**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  
Черепов Олександр Павлович  
номер у списку групи: 32

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 коми.

Варіант №2

$$P = \prod_{i=1}^n \frac{i^2 + 1}{\sum_{j=1}^i (j + \sin(j))}$$

# Тексти програм

## I програма:

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

int main(void) {

    double product=1;
    double sum=0;
    int j;
    int n;
    int i;

    printf("Welcome! Input n: ");
    scanf("%i", &n);

    if (n<=0){
        printf("Error: n should be greater than 0!\n");
    }
    else {
        for (i=1; i<=n; i++) {
            sum=0;
            for (j=1; j<=i; j++) {
                sum+=j+sin(j);
            }
            product*=((i*i)+1)/sum;
        }
        printf("Result: %.7lf\n", product);
    }

    return 0;
}
```

## II програма:

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

int main(void) {

    double product=1;
    double sum=0;
    int n;
    int i;

    printf("Welcome again! Input n: ");
    scanf("%i", &n);

    if (n<=0) {
        printf("Error: n should be greater than 0!\n");
    }
}
```

```

    }
    else {
        for (i=1; i<=n; i++) {
            sum+=i*sin(i);
            product*=((i*i)+1)/sum;
        }
        printf("Result: %.7lf\n", product);
    }

    return 0;
}

```

## Розрахунки кількості операцій

### I програма:

```

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

int main(void) {

    double product=1;
    double sum=0;
    int ct = 0;
    int j;
    int n;
    int i;

    printf("Welcome! Input n: ");
    scanf("%i", &n);

    if (n<=0){
        printf("Error: n should be greater than 0!\n");
    }
    else {
        for (i=1; i<=n; i++) {
            sum=0;
            for (j=1; j<=i; j++) {
                sum+=j*sin(j);
                ct+=6;
            }
            product*=((i*i)+1)/sum;
            ct+=9;
        }
        ct+=1;
        printf("Result: %.7lf\n", product);
    }
    ct+=4;
    printf("Operations done: %i\n", ct);

    return 0;
}

```

## II програма:

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

int main(void) {

    double product=1;
    double sum=0;
    int ct=0;
    int n;
    int i;

    printf("Welcome again! Input n: ");
    scanf("%i", &n);

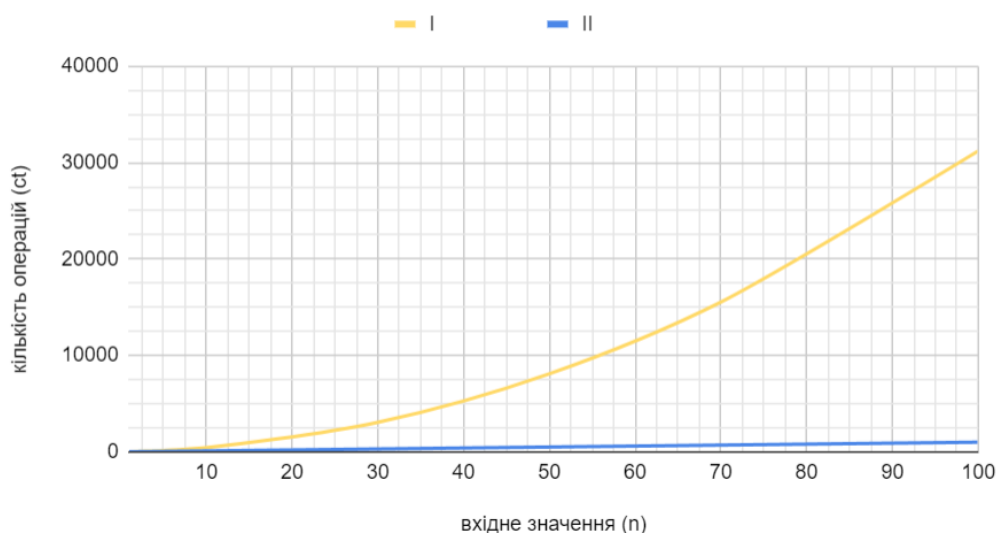
    if (n<=0) {
        printf("Error: n should be greater than 0!\n");
    }
    else {
        for (i=1; i<=n; i++) {
            sum+=i+sin(i);
            product*=((i*i)+1)/sum;
            ct+=10;
        }
        ct+=1;
        printf("Result: %.7lf\n", product);
    }
    ct+=4;
    printf("Operations done: %i\n", ct);

    return 0;
}
```

Виконуємо програми для деяких  $n$ . Кількості операцій записуємо в таблицю до відповідного значення  $n$ . Будуємо графік залежності кількості операцій ( $ct$ ) від  $n$ .

I програма		II програма	
$n$	$ct$	$n$	$ct$
1	20	1	15
3	68	3	35
5	140	5	55
10	425	10	105
30	3065	30	305
50	8105	50	505
70	15545	70	705
100	31205	100	1005

Графік залежності  $ct$  від  $n$



## Перевірка калькулятором

$n = 3$ :

Підстановка:

$$P_3 = \prod_{i=1}^3 \frac{i^2 + 1}{\sum_{j=1}^i (j + \sin(j))} = \frac{1^2 + 1}{1 + \sin(1)} \times \frac{2^2 + 1}{1 + \sin(1) + 2 + \sin(2)} \times \frac{3^2 + 1}{1 + \sin(1) + 2 + \sin(2) + 3 + \sin(3)} =$$
$$= \frac{100}{(1 + \sin(1))(3 + \sin(1) + \sin(2))(6 + \sin(1) + \sin(2) + \sin(3))} \approx 1.4484060$$

Результат WolframAlpha:

$$\prod_{i=1}^3 \frac{2(1 + i^2)}{i + i^2 + \cot\left(\frac{1}{2}\right) - \cos(i) \cot\left(\frac{1}{2}\right) + \sin(i)} \approx$$

1.448406026941595080357055902886210171667

$n = 5$ :

Підстановка:

$$P_5 = \prod_{i=1}^5 \frac{i^2 + 1}{\sum_{j=1}^i (j + \sin(j))} = \frac{1^2 + 1}{1 + \sin(1)} \times \frac{2^2 + 1}{1 + \sin(1) + 2 + \sin(2)} \times \frac{3^2 + 1}{1 + \sin(1) + 2 + \sin(2) + 3 + \sin(3)} \times$$
$$\times \frac{4^2 + 1}{1 + \sin(1) + 2 + \sin(2) + 3 + \sin(3) + 4 + \sin(4)} \times \frac{5^2 + 1}{1 + \sin(1) + 2 + \sin(2) + 3 + \sin(3) + 4 + \sin(4) + 5 + \sin(5)} =$$
$$= \frac{44200}{(1 + \sin(1))(3 + \sin(1) + \sin(2))(6 + \sin(1) + \sin(2) + \sin(3))(10 + \sin(1) + \sin(2) + \sin(3) + \sin(4))(15 + \sin(1) + \sin(3) + \sin(4) + \sin(5))} \approx 3.7884109$$

Результат WolframAlpha:

$$\prod_{i=1}^5 \frac{2(1 + i^2)}{i + i^2 + \cot\left(\frac{1}{2}\right) - \cos(i) \cot\left(\frac{1}{2}\right) + \sin(i)} \approx$$

3.788410888572291746793694890019476764959

# Тестування програм

## I програма:

```
D:\education\EducationalWork\EducationalWork\АСД\000LABS\lab2\programme\1>main.exe
Welcome! Input n: 1
Result: 1.0860883
Operations done: 20

D:\education\EducationalWork\EducationalWork\АСД\000LABS\lab2\programme\1>main.exe
Welcome! Input n: 3
Result: 1.4484060
Operations done: 68

D:\education\EducationalWork\EducationalWork\АСД\000LABS\lab2\programme\1>main.exe
Welcome! Input n: 5
Result: 3.7884109
Operations done: 140
```

## II програма:

```
D:\education\EducationalWork\EducationalWork\АСД\000LABS\lab2\programme\2>main.exe
Welcome again! Input n: 1
Result: 1.0860883
Operations done: 15

D:\education\EducationalWork\EducationalWork\АСД\000LABS\lab2\programme\2>main.exe
Welcome again! Input n: 3
Result: 1.4484060
Operations done: 35

D:\education\EducationalWork\EducationalWork\АСД\000LABS\lab2\programme\2>main.exe
Welcome again! Input n: 5
Result: 3.7884109
Operations done: 55
```

## Висновок

На лабораторній роботі я розв'язував задачу по знаходженню значення виразу, означеного циклічною формулою, за допомогою мови програмування С. Виконав завдання двома способами: з використанням вкладеного циклу та методом динамічного програмування. Також для кожної програми було створено лічильник кількості операцій для подальшого порівняння складності та ефективності обох алгоритмів.

Порівнявши значення, переконався, що метод динамічного програмування є набагато ефективнішим, адже у ході виконання алгоритму, створеного за цим методом, операцій виконується в рази менше. Особливо це помітно при більшій кількості ітерацій циклу. Це також зумовлено лінійною складністю цього алгоритму ( $O(n)$ ) у порівнянні з квадратичною складністю алгоритму із вкладеним циклом ( $O(n^2)$ ).

Отже, в подальшому використовуватиму саме метод динамічного програмування там, де це можливо.