

**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»**

**Виконала:**

**студентка групи ІМ-43  
Козаченко Софія Олександрівна  
номер у списку групи: 14**

**Перевірила:**

**Молчанова А. А**

**Київ 2024**

### Завдання :

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.

### Варіант 14 :

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

### Текст програми :

1)

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

int main()
{
    int n;
```

```

double mult = 1.0;

printf("Enter n: ");

scanf("%d", &n);


    for (int i = 1; i <= n; i++)
    {

        double sum = 0.0;

        for (int j = 1; j <= i; j++)
        {

            sum += sin(j);

        }

        mult *= (cos(i) + 1) / sum;

    }

printf("%.7lf\n", mult);
}

```

2)

```

#include <stdio.h>

#include <math.h>


int main() {

    int n;

    printf("Enter n: ");

    scanf("%d", &n);

```

```
double sum = 0.0;

double mult = 1.0;

for (int i = 1; i <= n; i++) {
    sum += sin(i);

    double result = (cos(i) + 1) / sum;

    mult *= result;
}

printf("%.71f\n", mult);
}
```

Розрахунки кількості операцій :

1)

```
#include <stdio.h>

#include <math.h>

int main()
{
    int n;

    double mult = 1.0;

    int oper_counter = 0;

    int sin_counter = 0;

    int cos_counter = 0;
```

```
printf("Enter n: ");  
scanf("%d", &n);  
  
oper_counter += 3;  
  
for (int i = 1; i <= n; i++)  
{  
    double sum = 0.0;  
    for (int j = 1; j <= i; j++)  
    {  
        sum += sin(j);  
        oper_counter += 4;  
        sin_counter += 1;  
    }  
    mult *= (cos(i) + 1) / sum;  
    oper_counter += 8;  
    cos_counter += 1;  
}  
  
int oper_result = oper_counter + sin_counter +  
cos_counter;  
  
printf("the number of operations: %u\n",  
oper_result);  
  
printf("%.7lf\n", mult);  
}
```

2)

```
#include <stdio.h>

#include <math.h>

int main() {

    int n;

    int oper_counter = 0;

    int sin_counter = 0;

    int cos_counter = 0;

    printf("Enter n: ");

    scanf("%d", &n);

    double sum = 0.0;

    double mult = 1.0;

    oper_counter +=4;

    for (int i = 1; i <= n; i++) {

        sum += sin(i);

        double result = (cos(i) + 1) / sum;

        mult *= result;

        oper_counter +=8;

        sin_counter += 1;

        cos_counter += 1;

    }
```

```
    int oper_result = oper_counter + sin_counter +  
cos_counter;  
  
    printf("the number of operations: %u\n",  
oper_result);  
  
    printf("%.7lf\n", mult);  
}
```

## Скріншоти тестування програм :

1)

```
"C:\Users\kozac\Desktop\unik\acd\laba 2\2.1.1\cmake-build-debug\untitled.exe"  
Enter n:2  
    the number of operations: 36  
0.6104383
```

```
"C:\Users\kozac\Desktop\unik\acd\laba 2\2.1.1\cmake-build-debug\untitled.exe"  
Enter n:3  
    the number of operations: 60  
0.0032290
```

```
"C:\Users\kozac\Desktop\unik\acd\laba 2\2.1.1\cmake-build-debug\untitled.exe"  
Enter n:7  
    the number of operations: 206  
-0.4317164
```

2)

```
"C:\Users\kozac\Desktop\unik\acd\laba 2\2.2.2\cmake-build-debug\untitled1.exe"  
Enter n:2  
    the number of operations: 24  
0.6104383
```

```
"C:\Users\kozac\Desktop\unik\acd\laba 2\2.2.2\cmake-build-debug\untitled1.exe"  
Enter n:3  
    the number of operations: 34  
0.0032290
```

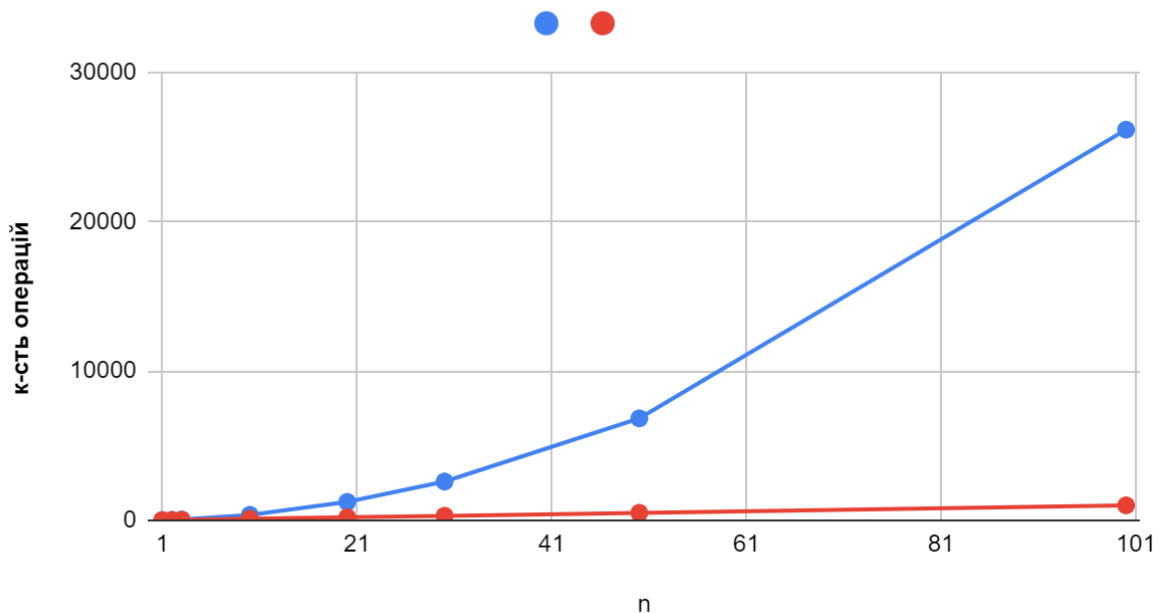
```
"C:\Users\kozac\Desktop\unik\acd\laba 2\2.2.2\cmake-build-debug\untitled1.exe"  
Enter n:7  
    the number of operations: 74  
-0.4317164
```

### Таблиця з результатами запуску

n		1	2	3	10	20	30	50	100
Кількість операцій	1 спосіб	17	36	60	368	1233	2598	6828	26153
	2 спосіб	14	24	34	104	204	304	504	1004

### Графіки :

Графік залежність кількості операцій від n



### Результати :

$$\prod_{i=1}^1 \frac{\cos(i) + 1}{\sum_{j=1}^i \sin(j)} = \frac{\cos(1) + 1}{\sin(1)} = \frac{0.5403 + 1}{0.8415} = \frac{1.5403}{0.8415} \approx 1.83$$



$$\begin{aligned}
\prod_{i=1}^2 \frac{\cos(i) + 1}{\sum_{j=1}^i \sin(j)} &= \frac{\cos(1) + 1}{\sin(1)} \times \frac{\cos(2) + 1}{\sin(1) + \sin(2)} \\
&= \frac{0.5403 + 1}{0.8415} \times \frac{-0.4161 + 1}{0.8415 + 0.9093} \\
&= \frac{1.5403}{0.8415} \times \frac{0.5839}{1.7508} \\
&= 1.8305 \times 0.3336 \approx 0.6107
\end{aligned}$$

$$\begin{aligned}
\prod_{i=1}^3 \frac{\cos(i) + 1}{\sum_{j=1}^i \sin(j)} &= \frac{\cos(1) + 1}{\sum_{j=1}^1 \sin(j)} \times \frac{\cos(2) + 1}{\sum_{j=1}^2 \sin(j)} \times \frac{\cos(3) + 1}{\sum_{j=1}^3 \sin(j)} \\
&= \frac{1.5403}{0.8415} \times \frac{0.5839}{1.7508} \times \frac{0.01001}{1.8919} \\
&= 1.8305 \times 0.3336 \times 0.00529 \approx 0.00323
\end{aligned}$$

### **Висновок :**

Під час виконання лабораторної роботи я засвоїла теоретичний матеріал та набула практичних навичок використання різних циклічних керуючих конструкцій, вкладених циклів, методу динамічного програмування та обчислення кількості операцій алгоритмів. Виконавши цю лабораторну роботу двома способами я побачила, що динамічне програмування дозволяє виконати ту саму задачу за меншу кількість операцій ніж вкладені цикли.

