

**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-22
Тимофеев Даниїл Костянтинович
номер у списку групи: 20

Перевірила:

Молчанова А. А.

Київ 2022

Постановка задачі : 1. Задане натуральне число n . Вирахувати значення заданої формули за варіантом.

$$P = \prod_{i=1}^n \frac{\sum_{j=1}^i (2j+1)}{2i \cdot \ln(i+3)}$$

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.

Текст програм

Перша програма :

```
#include <stdio.h>
#include <math.h>
int main() {
    int n, counter;
    counter = 0;
    printf ("Please, write the value for N : \n"); //Programme with nested
loops
    scanf("%i", &n);
    double result = 1;
    for (int i = 1; i<=n; i++) {
        double x = 2*i*log(i+3);
        double y = 0;
        for (int j = 1; j<=i; j++){
            y += 2*j+1;
            counter += 6;
        }
        result *= y/x;
        counter += 11;
    }
    printf("The result of formula = %.7f \n", result);
    printf("The amount of operations is : %d\n", counter );
}
```

Друга програма :

```
#include <stdio.h>
#include <math.h>
int main() {
    int n, counter;
    counter = 0;
    printf ("Please, write the value for N : \n"); //Dynamic programming
    scanf("%i", &n);
    double result = 1;
    double y = 0;
    for (int i = 1; i<=n; i++) {
        double x = 2*i*log(i+3);
        y += 2*i+1;
        result *= y/x;
        counter += 13;
    }
    printf("The result of formula = %.7f \n", result);
    printf("The amount of operations is : %d\n", counter );
}
```

Скріншоти тестування програм

Перша програма :

```
Please, write the value for N :
2
The result of formula = 1.3445952
The amount of operations is : 40

Please, write the value for N :
3
The result of formula = 1.8760822
The amount of operations is : 69

Please, write the value for N :
4
The result of formula = 2.8923467
The amount of operations is : 104
```

Друга програма :

Please, write the value for N :

2

The result of formula = 1.3445952

The amount of operations is : 26

Process finished with exit code 0

Please, write the value for N :

3

The result of formula = 1.8760822

The amount of operations is : 39

Process finished with exit code 0

Please, write the value for N :

4

The result of formula = 2.8923467

The amount of operations is : 52

Process finished with exit code 0

Перевірка на калькуляторі

Для $N = 2$:

ШАГИ РЕШЕНИЯ

Упростите выражение

$$\frac{2+1}{2\ln(1+3)} \times \frac{2+1+4+1}{2 \times 2\ln(2+3)}$$

↓
Упростить

$$\frac{3}{2\ln(2)\ln(5)}$$

$\approx 1,3446$

Показать шаги по решению →

Для $N = 3$:

ШАГИ РЕШЕНИЯ

Упростите выражение

$$\frac{2+1}{2\ln(1+3)} \times \frac{2+1+4+1}{2 \times 2\ln(2+3)} \times \frac{2+1+4+1+6+1}{2 \times 3\ln(3+3)}$$



↓
Упростить

$$\frac{15}{4\ln(2)\ln(5)\ln(6)}$$

$\approx 1,87608$

Показать шаги по решению →

Для $N = 4$:

ШАГИ РЕШЕНИЯ

Упростите выражение

$$\frac{2+1}{2\ln(1+3)} \times \frac{2+1+4+1}{2 \times 2\ln(2+3)} \times \frac{2+1+4+1+6+1}{2 \times 3\ln(3+3)} \times \frac{2+1+4+1+6+1+8+1}{2 \times 4\ln(4+3)}$$



↓
Упростить

$$\frac{45}{4\ln(2)\ln(5)\ln(6)\ln(7)}$$

$\approx 2,89235$

Показать шаги по решению →

Висновок :

На основі лабораторної роботи можу стверджувати, що метод динамічного програмування є більш ефективним у порівнянні із програмою з вкладеними циклами, бо цей метод виконує меншу кількість операцій (див. графік та таблицю). Це означає, що програма не так сильно навантажує процесор та швидше працює.

Таблиця

<i>n</i>		1	2	3	10	20	30	50	100
Кількість операцій	Перша програма	17	40	69	440	1480	3120	8200	31400
	Друга програма	13	26	39	130	260	390	650	1300

Графік

