

**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
Костеніч Степан Станіславович
номер у списку групи: 17

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 №17

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

Текст програми №1

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

int main(void) {
    int n;

    printf("\nEnter n: \n");
    scanf("%d", &n);

    int operation_counter = 0;
    int math_call = 0;

    double sum = 0.0;
    double pow = 1.0;

    for (int i = 1; i <= n; i++) {
        double mul = 1.0;
        for (int j = 1; j <= i; j++) {
            mul *= j + cos(j);

            operation_counter += 5;
            math_call++;
        }
        pow *= 4.0;
        sum += mul / (pow - i);

        operation_counter += 9 + math_call;
    }

    operation_counter += 3;

    printf("The number of operations %d\n", operation_counter);
    printf("The result is %.7lf\n", sum);

    return 0;
}
```

Текст програми №2

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

int main(void) {
    int n;

    printf("\nEnter n: \n");
    scanf("%d", &n);

    int operation_counter = 0;
    int math_call = 0;

    double sum = 0.0;
    double mul = 1.0;
    double pow = 1.0;

    for (int i = 1; i <= n; i++) {
        mul *= i + cos(i);
        pow *= 4.0;
        sum += mul / (pow - i);

        operation_counter += 9;
        math_call++;
    }
    operation_counter += 4 + math_call;

    printf("The number of operations %d\n", operation_counter);
    printf("The result is %.7lf\n", sum);

    return 0;
}
```

Результати тестування програми №1

```
C:\Users\Stepan\CLionProjects\asd\lab_2\cmake-build-debug\lab_2.exe
```

```
Enter n:
```

```
2
```

```
The result is 0.6876922
```

```
Enter n:
```

```
3
```

```
The result is 0.7680797
```

```
Enter n:
```

```
7
```

```
The result is 1.3515391
```

```
Enter n:
```

```
10
```

```
The result is 5.6437567
```

```
Enter n:
```

```
20
```

```
The result is 2339522.7724877
```

```
Enter n:
```

```
30
```

```
The result is 207561992699393.3437500
```

Результати тестування програми №2

```
C:\Users\Stepan\CLionProjects\asd\lab_2\cmake-build-debug\main2.exe
```

```
Enter n:
```

```
2
```

```
The result is 0.6876922
```

```
Enter n:
```

```
3
```

```
The result is 0.7680797
```

```
Enter n:
```

```
7
```

```
The result is 1.3515391
```

```
Enter n:
```

```
10
```

```
The result is 5.6437567
```

```
Enter n:
```

```
20
```

```
The result is 2339522.7724877
```

```
Enter n:
```

```
30
```

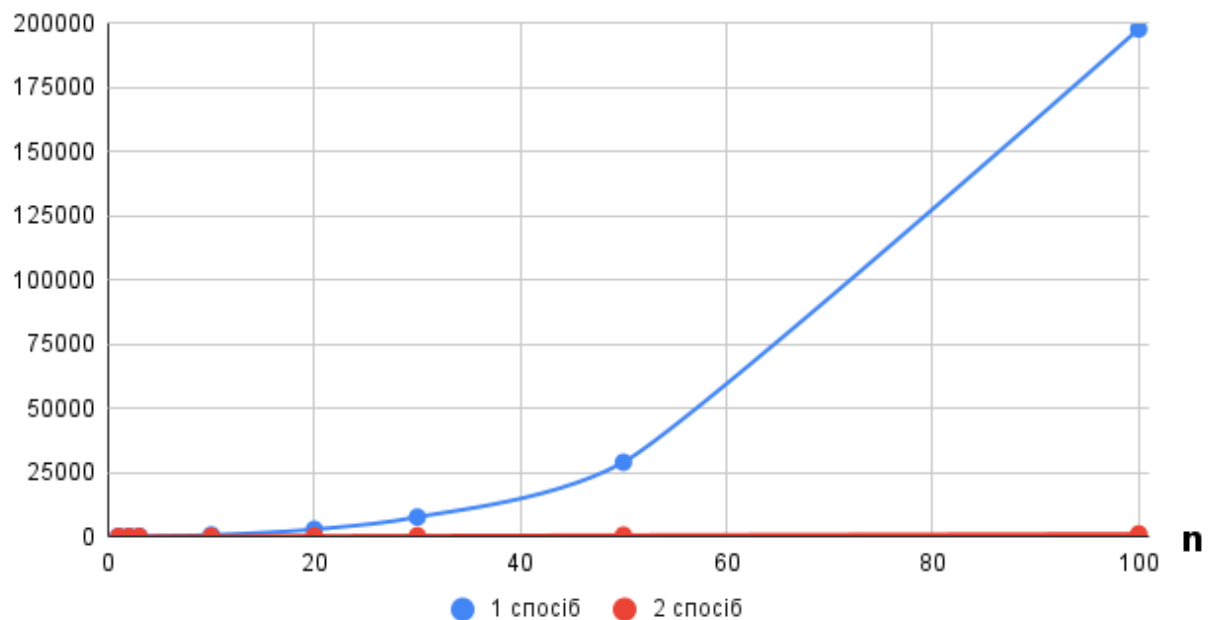
```
The result is 207561992699393.3437500
```

Таблиця з результатами запуску

n		1	2	3	10	20	30	50	100
Кількість операцій	1 спосіб	18	40	70	588	2773	7558	28928	197853
	2 спосіб	14	24	34	104	204	304	504	1004

Графік за таблицею

Кількість операцій



Результати перевірки

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

$$f(2)$$

$$= 0.687692150515$$

$$f(3)$$

$$= 0.768079688086$$

$$f(7)$$

$$= 1.35153913318$$

$$f(10)$$

$$= 5.64375669225$$

$$f(20)$$

$$= 2339522.77249$$

$$f(30)$$

$$= 2.075619927 \times 10^{14}$$

Висновок

Під час виконання лабораторної роботи № 2 я засвоїв теоретичний матеріал та набув практичних навичок використання різних циклічних керуючих конструкцій, вкладених циклів, методу динамічного програмування та обчислення кількості операцій алгоритмів.

Результати, отримані в ході виконання обох програм, продемонстрували суттєву різницю в кількості операцій. Написання першої програми було якоюсь мірою легшим завдяки циклам, але кількість операцій була великою. Друга програма з використанням динамічного програмування була значно ефективнішою і такою ж мірою простою, порівняно з першою. При тестуванні стало видно меншу ефективність першого методу для великих значень n через збільшення часу виконання. Другий метод навпаки показав кращу ефективність у кількості операцій.

На прикладі обох програм явно стало видно переваги динамічного програмування та мемоїзації для виконання довгих та складних обчислень. Зручність й ефективність мемоїзації я також побачив під час імплементації піднесення до степеня без використання окремого циклу для цього. Це звільнило мене від використання додаткових ресурсів та збільшило продуктивність програми.

Для піднесення до степеня я створював окрему змінну `row` за межами циклу, спочатку взявши за тип `int`. Уже при $n = 16$ програма перестала коректно працювати, тому я перевіряв також `long int` та `long long int`, але вже вдруге переконався у потрібності використання `double` при великих обчисленнях.

Отже, виконання лабораторної роботи № 2 було корисним, дозволило закріпити теоретичні знання та набути практичних навичок в області програмування мовою C.