

**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”

Виконав

Перевірила:

Молчанова А. А.

Студент групи ІМ-13

Кірович Олександр Костянтинович

номер у списку групи: 10

Київ - 2021

ЛАБОРАТОРНА РОБОТА №1.2.
АЛГОРИТМИ З ВКЛАДЕНИМИ ЦИКЛАМИ ТА
МЕТОД ДИНАМІЧНОГО ПРОГРАМУВАННЯ

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.

Варіант №10

$\begin{matrix} 1 \\ 0 \\ . \end{matrix}$	$P = \prod_{i=1}^n \frac{i \cdot (i + 1)}{\sum_{j=1}^i (\sin(j) + 1)}$
---	--

Код до завдання №1

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
int main()
{
    int n;
    int ct = 0;
    double result = 1.00;
    double i, j, sum;
    scanf ("%d", &n);
    for (i = 1; i <= n; i++){
        double sum = 0.00;
        for (j = 1; j <= i; j++){
            sum += sin (j) + 1;
            ct += 6;
        }
        result *= i*(i+1)/sum;
        ct += 8;
    }
    printf("%.7lf\n", result);
    printf("ct =: %d", ct);
    return 0;
}
```

Тестування

```
1
1.0860883
ct =: 14

...Program finished with exit code 0
Press ENTER to exit console.
```

```
2
1.7373852
ct =: 34

...Program finished with exit code 0
Press ENTER to exit console.
```

```
3
4.2618761
ct =: 60

...Program finished with exit code 0
Press ENTER to exit console.
```

```
10
3035196.8220279
ct =: 410

...Program finished with exit code 0
Press ENTER to exit console.
```

```
20
2489447429242830336.0000000
ct =: 1420

...Program finished with exit code 0
Press ENTER to exit console.
```

```
30
265579097969541451253971404455936.0000000
ct =: 3030

...Program finished with exit code 0
Press ENTER to exit console.
```

Код до завдання №2

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
int main()
{
    int n;
    int ct = 0;
    double result = 1.00;
```

```
double i, sum;

scanf ("%d", &n);
for (i = 1; i <= n; i++){
    sum += sin (i) + 1;
    result *= i*(i+1)/sum;
    ct += 10;
}
printf("%.7lf\n", result);
printf("ct =: %d", ct);
return 0;
}
```

Тестування

```
1
1.0860883
ct =: 10

...Program finished with exit code 0
Press ENTER to exit console.□
```

```
2
1.7373852
ct =: 20

...Program finished with exit code 0
Press ENTER to exit console.□
```

```
3
4.2618761
ct =: 30

...Program finished with exit code 0
Press ENTER to exit console.□
```

```
10
3035196.8220279
ct =: 100

...Program finished with exit code 0
Press ENTER to exit console.□
```

```

20
2489447429242830336.0000000
ct =: 200

...Program finished with exit code 0
Press ENTER to exit console.

```

```

30
265579097969541451253971404455936.0000000
ct =: 300

...Program finished with exit code 0
Press ENTER to exit console.

```

Перевірка на калькуляторі WolframAlpha

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

4.261876096114569998558304897056867669167

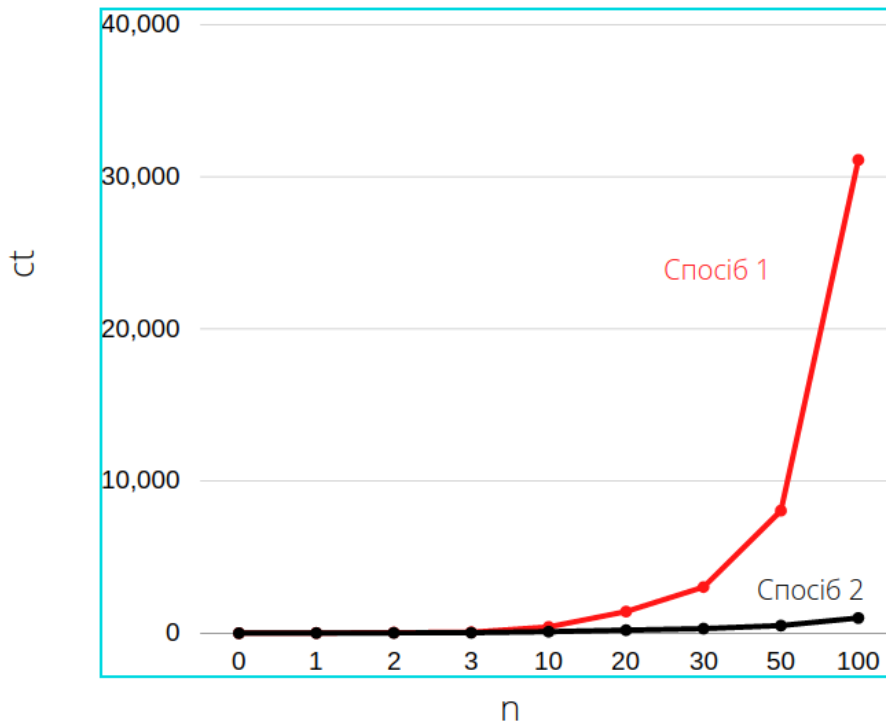
$$\prod_{i=1}^{10} i \cdot \frac{(i+1)}{\sum_{j=1}^i (\sin(j)+1)}$$

$3.035196822027931782741421929436476393148 \times 10^6$

Таблиця:

n		1	2	3	10	20	30	50	100
Кількість операцій	Спосіб 1	14	34	60	410	1420	3030	8050	31100
	Спосіб 2	10	20	30	100	200	300	500	1000

Графік (canva.com):



Висновок

В лабораторній роботі було розроблено 2 способи рахування значення заданої формули. З результатів тестування, таблиці й графіку видно, що перший спосіб є набагато складнішим за другий. Складність способу 2 зростає лінійно, а способу 1 - квадратично. Результат способу 2 був досягнут завдяки методу динамічного програмування, який зберігати результати попередніх обчислень, тоді як у способі 1 був застосований метод вкладених циклів.