**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: Перевірила:

студент групи IM-43 Молчанова А. А.  
Олексійчук Станіслав Юрійович  
номер у списку групи: 23

Київ 2024

***Постановка задачі***

1. Задане натуральне число n. Вирахувати значення заданої формули за варіантом.
2. Для вирішення задачі написати дві програми:
3. перша програма повинна використовувати для обчислення формули вкладені цикли;
4. друга програма повинна виконати обчислення формули за допомогою одного циклу з використанням методу динамічного програмування.
5. Виконати розрахунок кількості операцій для кожного з алгоритмів за методикою, викладеною на лекції, додавши до неї підрахунок кількості викликів стандартних функцій.
6. Програма має правильно вирішувати поставлену задачу при будь-якому заданому n, для якого результат обчислення може бути коректно представлений типом double.
7. Результуючі дані вивести у форматі з сімома знаками після крапки.

***Варіант 23:***

***Тексти програм***

1. **1st task (not dynamic)**

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

int main()

{

double S = 0.0;

double P = 1.0;

int n;

printf("Input n: ");

scanf("%d",&n);

if (n>=0){

int i = 1;

int j = 1;

while (i<=n){

while (j<=i) {

P \*= (j+1)\*sin(j);

j++;

}

S+=P/(i\*(i+1));

i++;

}

printf("The sum is %.7f\n", S);

}

else {

printf("Error...\n");

}

return 0;

}

1. **2nd task (dynamic)**

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

int main()

{

double S = 0.0;

double P = 1.0;

int n;

printf("Input n: ");

scanf("%d",&n);

if (n>=0){

int i = 1;

while (i<=n) {

P \*= (i+1)\*sin(i);

S+=P/(i\*(i+1));

i++;

}

printf("The sum is %.7f\n", S);

}

else {

printf("Error...");

}

return 0;

}

***Розрахунки кількості операцій***

Для зручності та правильності обчислень операцій було написано програму для двох варіантів розв’язку виразу:

1. **1st task\_operations (not dynamic)**

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

int main()

{

double S = 0.0;

double P = 1.0;

int n;

int ct = 0;

int sin\_ct = 0;

int operations = 0;

printf("Input n: ");

scanf("%d",&n);

if (n>=0){

int i = 1;

while (i<=n){

int j = 1;

while (j<=i) {

P \*= (j+1)\*sin(j);

j++;

ct += 6;

sin\_ct++;

}

S+=P/(i\*(i+1));

i++;

ct += 8;

}

operations = ct + sin\_ct + 3;

printf("Operations = %d",operations);

}

else {

printf("Error...\n");

}

return 0;

}

1. **2nd task\_operations (dynamic)**

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

int main()

{

double S = 0.0;

double P = 1.0;

int n;

int ct = 0;

int sin\_ct = 0;

int operations = 0;

printf("Input n: ");

scanf("%d",&n);

if (n>=0){

int i = 1;

while (i<=n) {

P \*= (i+1)\*sin(i);

S+=P/(i\*(i+1));

i++;

ct+=10;

sin\_ct+=1;

}

operations = ct + sin\_ct + 3;

printf("Operations = %d",operations);

}

else {

printf("Error...");

}

return 0;

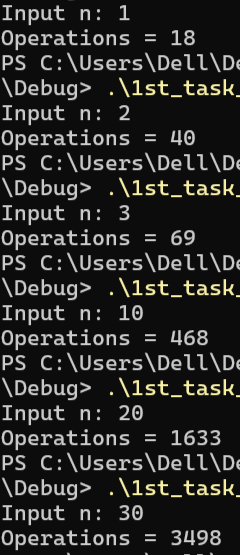
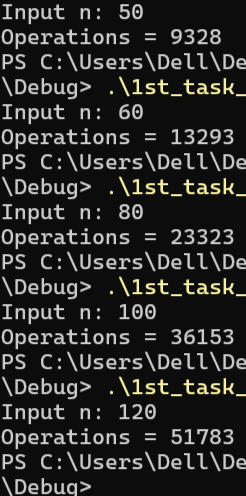
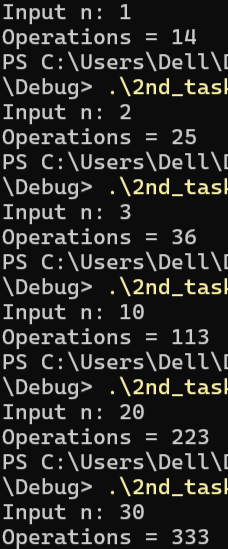
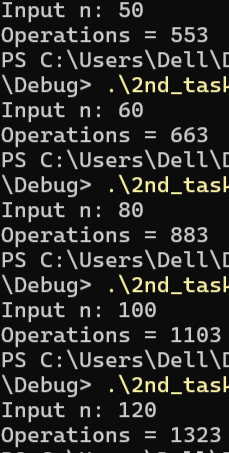
}

Виберемо деякі значення n, за яких обчислимо кількість операцій для кожного способу. Занесемо ці дані в таблицю:

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **n** | | **1** | **2** | **3** | **10** | **20** | **30** | **50** | **60** | **80** | **100** | **120** |
| **Кількість**  **операцій** | **1 спосіб** | 18 | 40 | 69 | 468 | 1633 | 3498 | 9328 | 13293 | 23323 | 36153 | 51783 |
| **2 спосіб** | 14 | 25 | 36 | 113 | 223 | 333 | 553 | 663 | 883 | 1103 | 1323 |

Відобразімо ці дані у вигляді графіку:

Ось перевірка кількості операцій з допомогою програми:

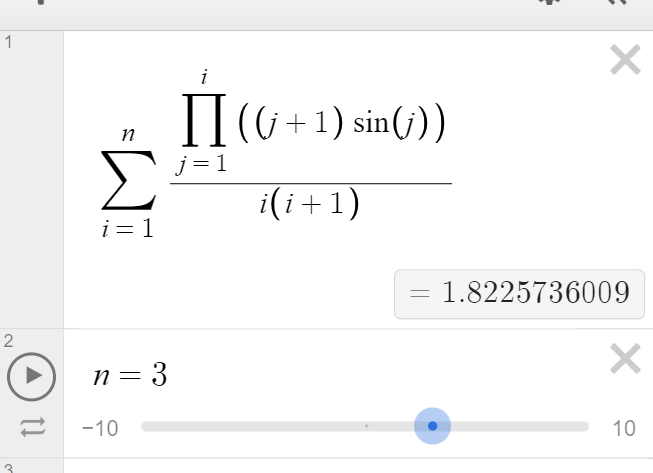
1)   2)  

***Обчислення виразу***

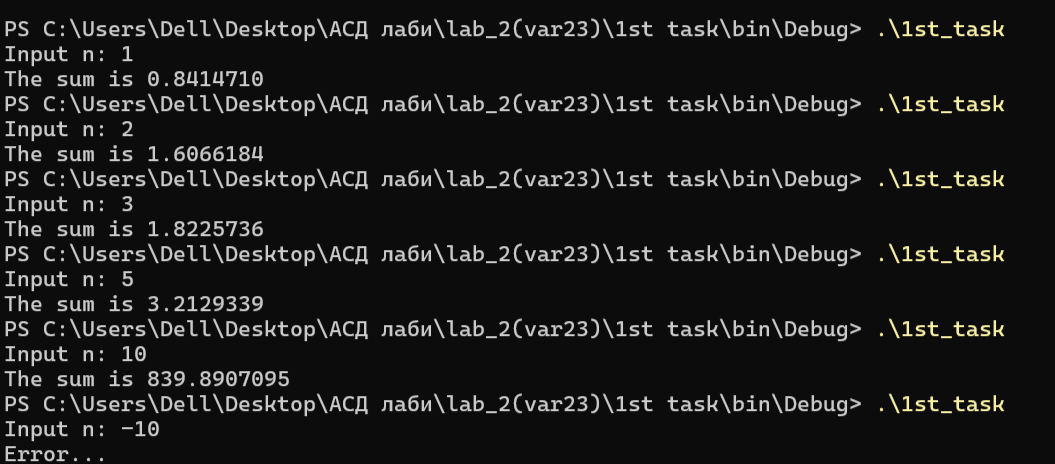
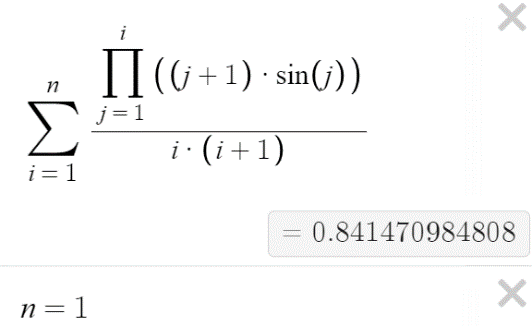
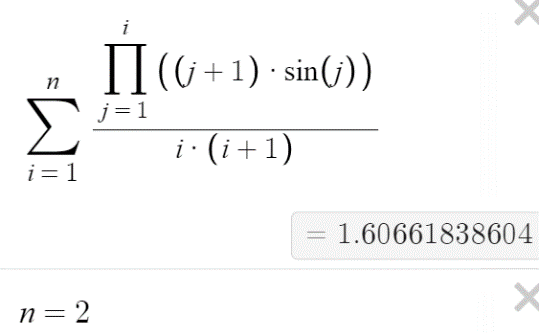
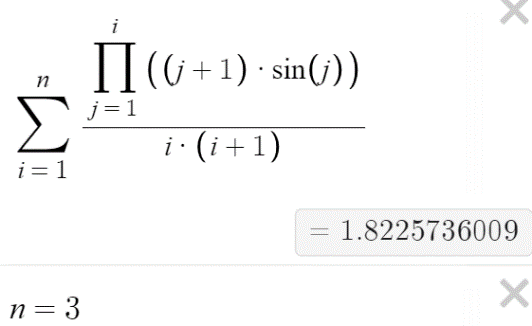
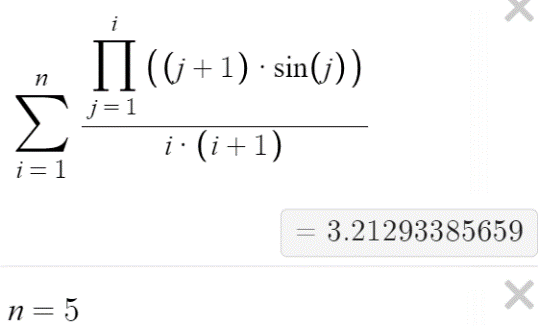
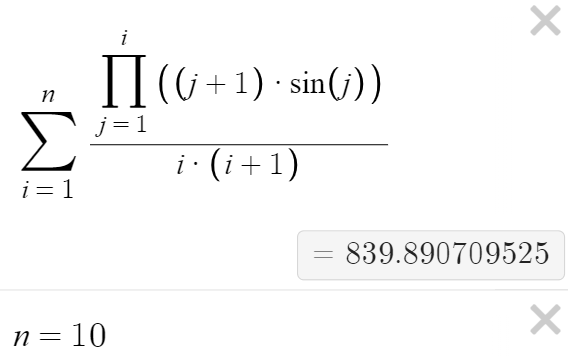
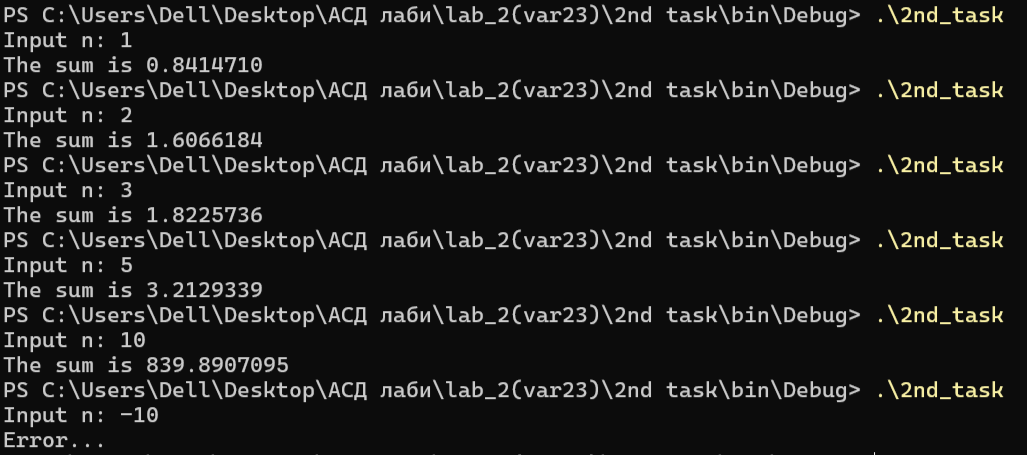
**1) Скорочення виразу без використання калькулятора**

Нехай n=3, тоді

1. **Рішення виразу з використанням графічного калькулятора** [**Desmos**](https://www.desmos.com/calculator?lang=uk)**:**

******

***Результати тестування програми***

1. 
2. 

Оскільки n=-10 не входить у проміжок значень натуральних чисел, то немає сенсу його перевіряти, оскільки на калькуляторі покаже 0, а в програмі від’ємні числа виведуть помилку «Error…»

***Висновок:*** я навчився використовувати цикли та вкладені цикли для розв’язування суми та добутку певних операцій у формулі, використовуючи середовище CodeBlocks; зрозумів, як використовувати динамічне програмування для оптимізації програми та відповідно зменшення кількості операцій програми, внаслідок чого зрозумів його перевагу; навчився рахувати кількість операцій у програмі, включаючи jump; зрозумів, що є операцією, а що – ні (що потрібно враховувати під час підрахунку кількості операцій, а що – ні); навчився виводити значення (представляти output) у вигляді числа з точністю до певної кількості знаків після коми (у цій роботі – до 7).