

**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:

студент групи ІМ-32  
Король Олександр Володимирович  
номер у списку групи: 15

Перевірила:

Молчанова А. А.

## 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. Результуючі дані вивести у форматі з сімома знаками після коми.

Варіант 15

$$S = \sum_{i=1}^n \frac{\sin(i) + 2}{i + \prod_{j=1}^i \sin(j)}$$

Текст програми яка використовує для обчислення 1 цикл з використанням методів динамічного програмування:

```
#include <stdio.h>

#include <math.h>

int main() {

    int n, i, count; double s, r;

    printf("Enter n: \n");

    scanf("%d", &n);

    s = 0; r = 1; count = 0;

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

        r *= sin(i);

        s += (sin(i) + 2) / (i + r);

        count += 10;

        // *=, sin(), +=, sin(), +, +, /, <=, ++, jump

    }

    count += 3;

    // 3 операції присвоєння

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

    printf("%d\n", count);

    return 0;

}
```

## Текст програми яка використовує для обчислення вкладені цикли:

```
#include <stdio.h>

#include <math.h>

int main() {

    int n, i, j, count; double s, r;

    printf("Enter n: \n");

    scanf("%d", &n);

    s = 0; count = 0;

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

        r = 1;

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

            r *= (sin(j)); // <=, ++, sin(), *=, jump - 5

            count += 5;

        }

        s += (sin(i) +2) / (i + r);

        // <=, ++, sin(), +, +, /, +=, =, =, jump - 10

        count += 10;

    }

    count += 2; // 2

    printf("%d \n",count);

    printf("%.7lf",s);

    return 0;

}
```

**Розрахунки кількості операцій для програми яка використовує для обчислення вкладені цикли:**

присвоєння  $s = 0$ : 1 операція

зовнішній цикл:

присвоєння  $r = 1$ : 1 операція

присвоєння  $j = 1$  - 1 операція

внутрішній цикл

$j \leq i$ ;  $j++$ , стрибок на наступну ітерацію - 3 операції

Обчислення  $r *= \sin(j)$ : 2 операції (присвоєння та множення, розрахунок синусу).

Обчислення  $s += (\sin(i) + 2) / (i + r)$ : 5 операцій (розрахунок  $\sin(i)$ , додавання  $+2$ , додавання  $i + r$ , ділення всього виразу, додавання з  $s$  та присвоєння  $s$  результату обчислень)

$i \leq n$ ;  $i++$ , стрибок на наступну ітерацію - 3 операції

**Розрахунки кількості операцій для програми яка використовує для обчислення 1 цикл з використанням методів динамічного програмування:**

присвоєння  $i = 1$ ,  $s = 0$ ;  $r = 1$ : 3 операція

цикл:

Множення  $r *= \sin(i)$ : 2 операції (присвоєння та множення, розрахунок синусу).

Обчислення  $s += (\sin(i) + 2) / (i + r)$ : 5 операцій (розрахунок  $\sin(i)$ , додавання  $+2$ , додавання  $i + r$ , ділення всього виразу, додавання з  $s$  та присвоєння  $s$  результату обчислень)

$i \leq n$ ;  $i++$ , стрибок на наступну ітерацію - 3 операції

## Тестування програми яка використовує для обчислення вкладені цикли:

```
D:\studying\university\labs\DSA-LABS\lab1.2\wayWithNestedCycles\cmake-build-debug\wayWithNestedCycles.exe
Enter n:
1
1.5430441
```

```
D:\studying\university\labs\DSA-LABS\lab1.2\wayWithNestedCycles\cmake-build-debug\wayWithNestedCycles.exe
Enter n:
2
2.5951752
Process finished with exit code 0
```

```
D:\studying\university\labs\DSA-LABS\lab1.2\wayWithNestedCycles\cmake-build-debug\wayWithNestedCycles.exe
Enter n:
3
3.2840862
Process finished with exit code 0
```

## Тестування програми яка використовує для обчислення 1 цикл з використанням методів динамічного програмування:

```
D:\studying\university\labs\DSA-LABS\lab1.2\wayWithDynamicProgramming\cmake-build-debug\wayWithDynamicProgramming.exe
Enter n:
1
1.5430441
Process finished with exit code 0
```

```
D:\studying\university\labs\DSA-LABS\lab1.2\wayWithDynamicProgramming\cmake-build-debug\wayWithDynamicProgramming.exe
Enter n:
2
2.5951752
Process finished with exit code 0
```

```
D:\studying\university\labs\DSA-LABS\lab1.2\wayWithDynamicProgramming\cmake-build-debug\wayWithDynamicProgramming.exe
Enter n:
3
3.2840862
Process finished with exit code 0
```

## Результати розрахунків за калькулятором:

Sum

$$\sum_{i=1}^1 \frac{\sin(i) + 2}{\prod_{j=1}^i \sin(j) + i} = \frac{2 + \sin(1)}{1 + \frac{1}{4} i (1 - e^i) e^{-i} (-1; e^i)_2}$$

((

Decimal approximation

1.5430441251857794943698361272581131671361528168571302294846285201

Sum

$$\sum_{i=1}^2 \frac{\sin(i) + 2}{\prod_{j=1}^i \sin(j) + i} = \frac{2 + \sin(1)}{1 + \frac{1}{4} i (1 - e^i) e^{-i} (-1; e^i)_2} + \frac{2 + \sin(2)}{2 - \frac{1}{8} e^{-3i} (-1; e^i)_3 (e^i; e^i)_2}$$

Decimal approximation

2.5951751709040092512210140340796671493097225080012993124822867415

Sum

$$\sum_{i=1}^3 \frac{\sin(i) + 2}{\prod_{j=1}^i \sin(j) + i} = \frac{2 + \sin(1)}{1 + \frac{1}{4} i (1 - e^i) e^{-i} (-1; e^i)_2} + \frac{2 + \sin(2)}{2 - \frac{1}{8} e^{-3i} (-1; e^i)_3 (e^i; e^i)_2} + \frac{2 + \sin(3)}{3 - \frac{1}{16} i e^{-6i} (-1; e^i)_4 (e^i; e^i)_3}$$

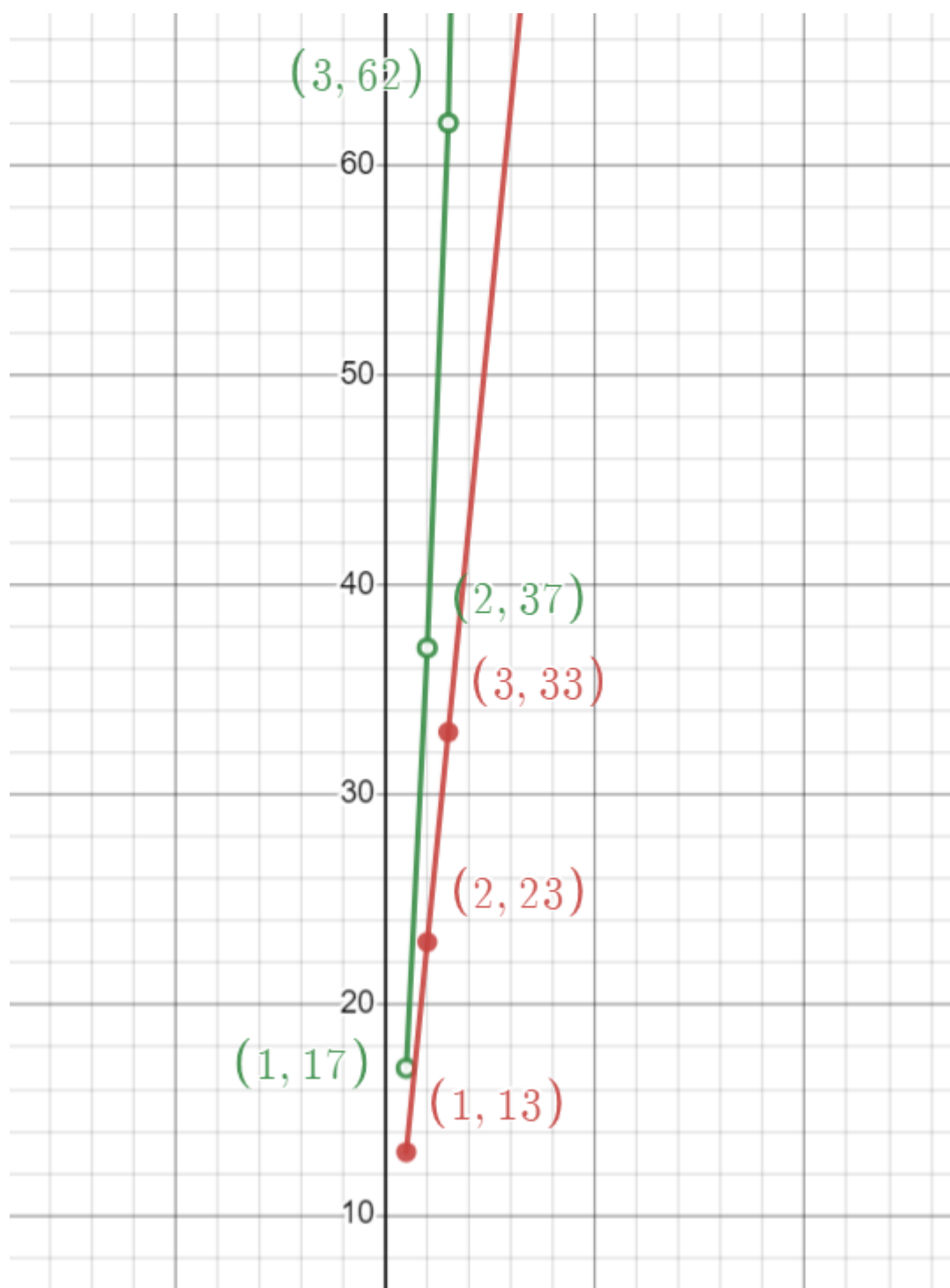
(d)

Decimal approximation

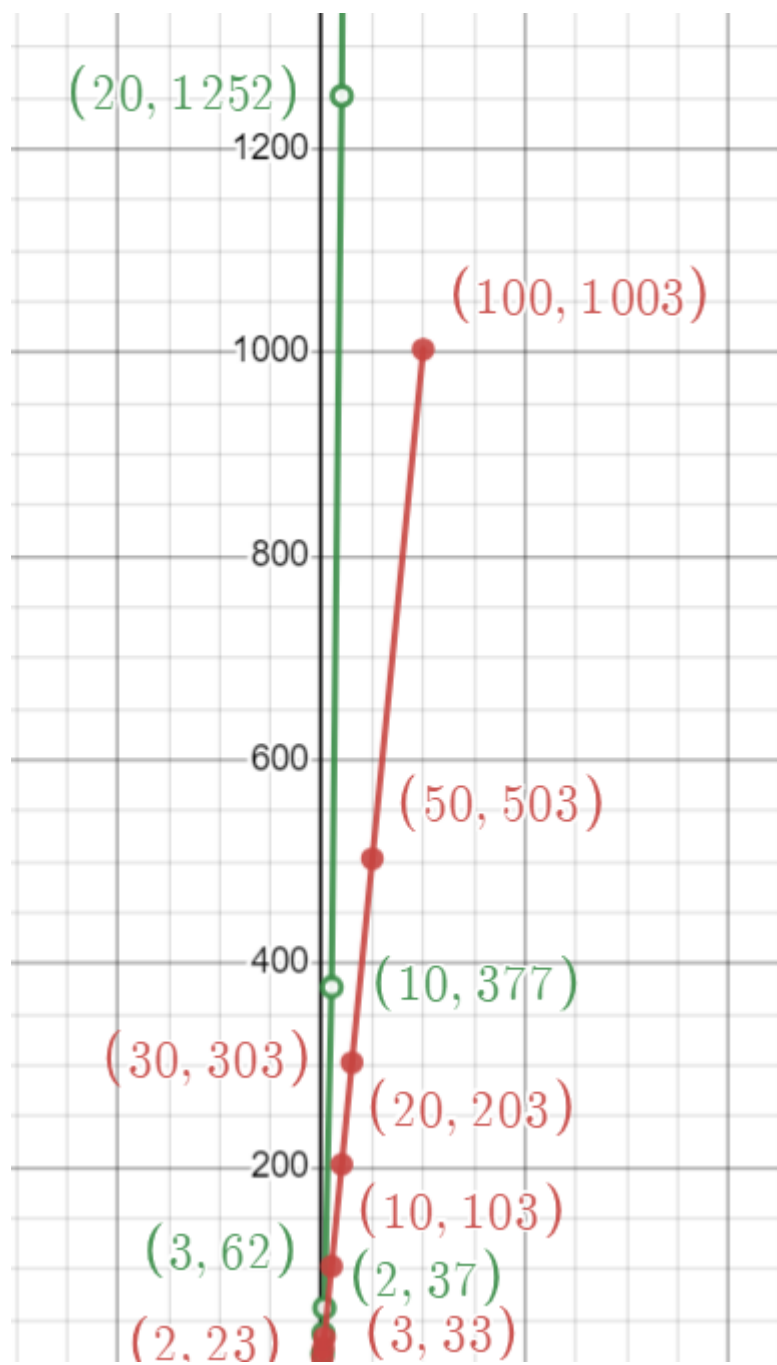
3.2840861858810252678821158045788479715234787293090782439619527231

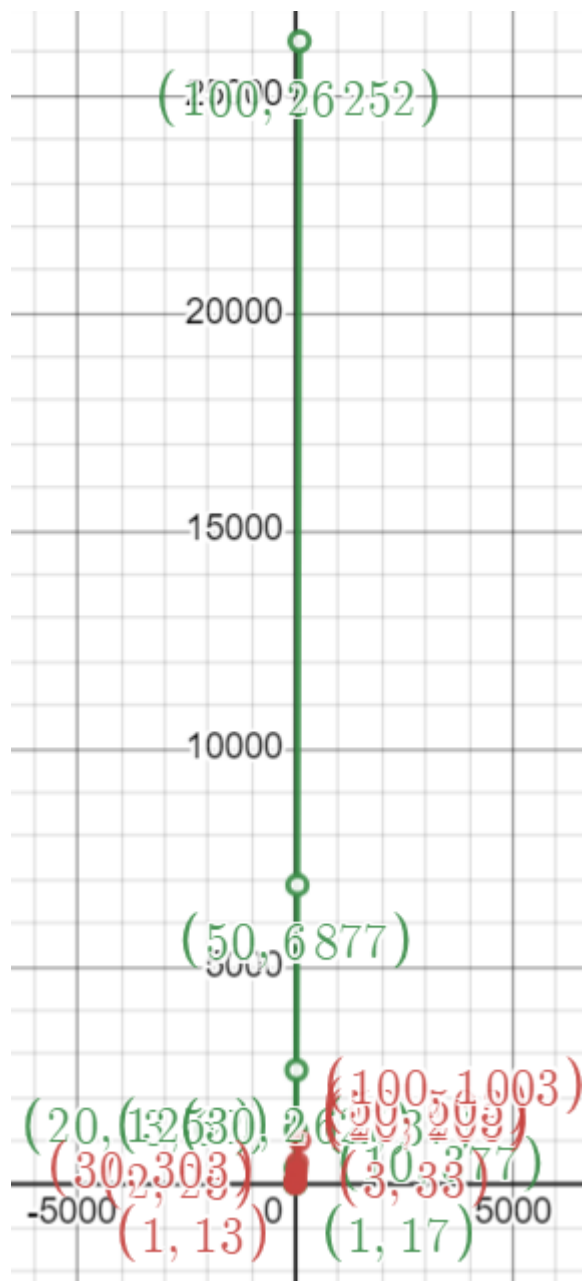
$n$		1	2	3	10	20	30	50	100	
Кількість операцій	1 спосіб	17	37	62	377	1252	2627	6877	26252	
	2 спосіб	13	23	33	103	203	303	503	1003	

графіки функцій:









червона - програма яка використовує для обчислення 1 цикл з використанням методів динамічного програмування

зелена- програма яка використовує для обчислення вкладені цикли