Лабараторна робота № 1.2

На тему: «АЛГОРИТМИ З ВКЛАДЕНИМИ ЦИКЛАМИ ТА МЕТОД

ДИНАМІЧНОГО ПРОГРАМУВАННЯ»

З дисципліни «Структури даних та алгоритми»

Виконав: Землянський Едуард

Група: КВ-22

**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 обчи-

слення формули вкладені цикли;

2) друга програма повинна виконати обчислення форму-

ли за допомогою одного циклу з використанням методу ди-

намічного програмування.

3. Виконати розрахунок кількості операцій для кожного з

алгоритмів за методикою, викладеною на лекції, додавши до неї

окремий підрахунок кількості викликів стандартних функцій, по-

значивши цю кількість KF, а також включити KF до загальної суми

кількості операцій.

4. Програма має правильно вирішувати поставлену задачу при

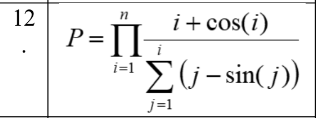
будь-якому заданому n, для якого результат обчислення може бути

коректно представлений типом double.

5. Результуючі дані вивести у форматі з сімома знаками після

крапки.

**Варіант №12:**



**Текст програми варіант 1** *(вкладені цикли)*

#include <stdio.h>

#include <math.h>

int main(){

    int n;

    float P = 1, j\_sum = 0;

    printf("Enter value of 'n': ");

    scanf("%d", &n);

    if (n >= 1){

        for (int i = 1; i <= n; i++)

        {

            j\_sum = 0;

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

            {

                j\_sum += (j - sin(j));

            }

            P \*= (i + cos(i)) / j\_sum;

        }

        printf("P = %0.7f", P);

    }

    else{

        printf("Wrong input data");

    }

    return 0;

}

**Текст програми варіант 2** *(За принципом динамічного прог-ня)*

#include <stdio.h>

#include <math.h>

int main(){

    int n;

    float P = 1, j\_sum = 0;

    printf("Enter value of 'n': ");

    scanf("%d", &n);

    if (n >= 1){

        for (int i = 1; i <= n; i++)

        {

            j\_sum += (i - sin(i));

            P \*= (i + cos(i)) / j\_sum;

        }

        printf("P = %0.7f", P);

    }

    else{

        printf("Wrong input data");

    }

    return 0;

}

**3.1. Розрахунок к-сті операцій для алгоритму 1**

Kil = n

Kjl = (n­2+n) / 2

Ka = (n­2+n) + 3n

Kas = (n­2+n) / 2 + 2n

Kc = 1

KF = (n­2+n) / 2 + n

Сума: 8.5n2 + 3.5n + 1

Кількість операцій має вид квадратиної залежності від n: O(n2)

**3.2. Розрахунок к-сті операцій для алгоритму 2**

Kil = n

Ka = 5n

Kas = 2n

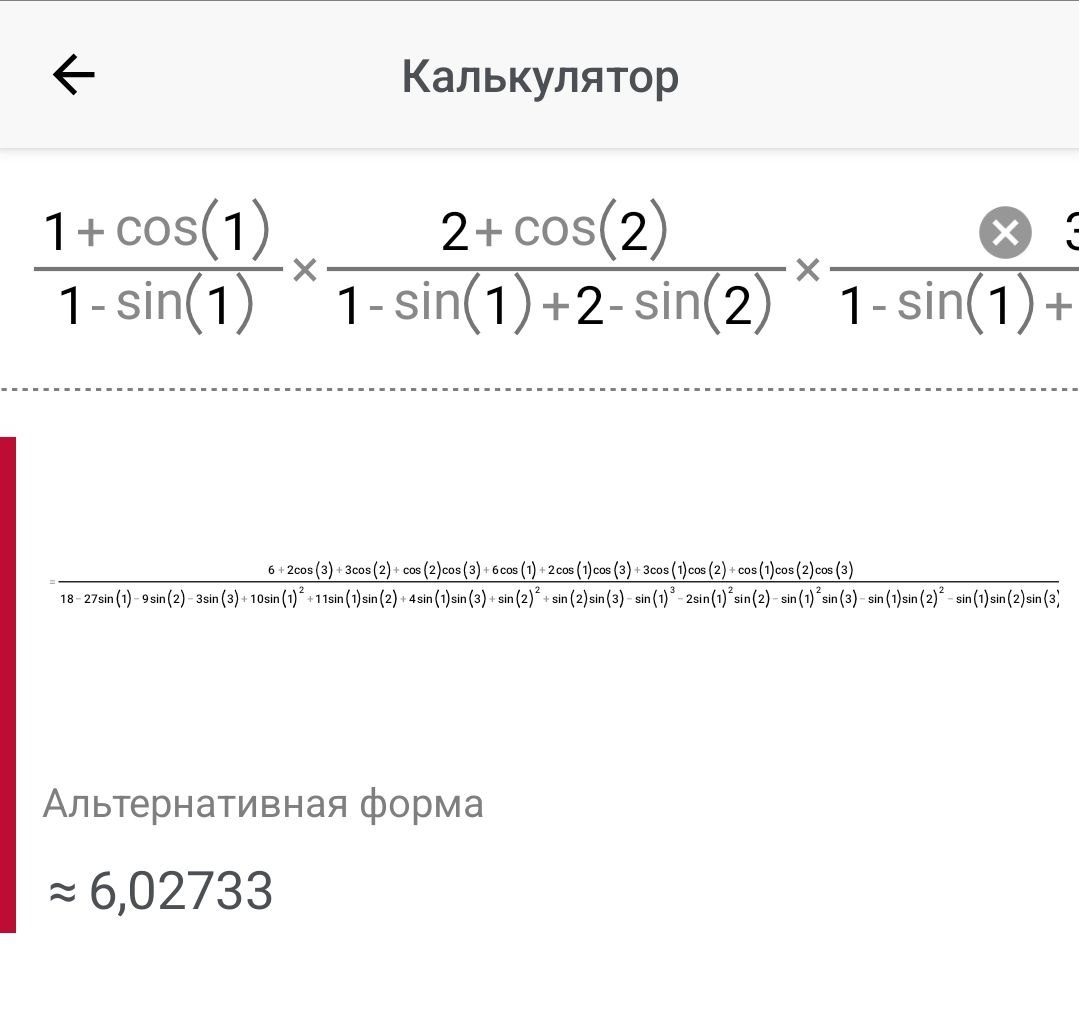
Kc = 1

KF = 2n

Сума: 10n + 1

Кількість операцій має вид лінійної залежності від n: O(n)

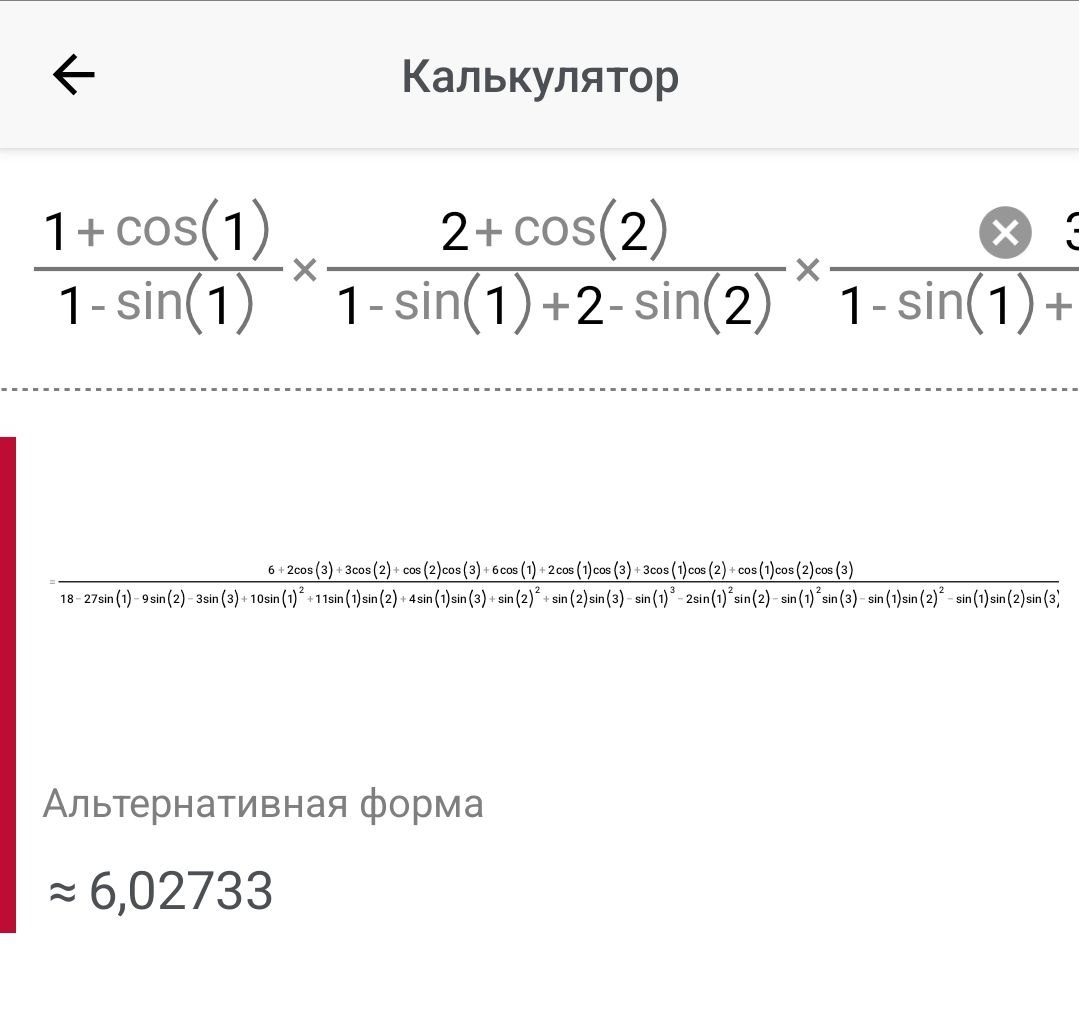
**4.1. Тестування алгоритму 1**



Значення з калькулятора: 6.02733

Значення, отримане розрахунком алгоритму: P = 6.0273256

**4.2. Тестування алгоритму 2**



Значення з калькулятора: 6.02733

Значення, отримане розрахунком алгоритму: P = 6.0273256

**5.1. Тестування алгоритму 1**

n = 1: 

n = 2: 

n = 3: 

**5.2. Тестування алгоритму 2**

n = 1: 

n = 2: 

n = 3: 