ДЕРЖАВНИЙ ТОРГОВЕЛЬНО-ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ

Кафедра інженерії програмного забезпечення та кібербезпеки

ТЕМА: Практичне завдання №4

([Основи програмування)](https://assignments.onenote.com/classes/f54d4356-2a75-4886-a001-7eb5c9117092/assignments/c11171b0-b386-4b3d-a7fd-8e1e9a08fde5?returnPath=%2Fclasses%2Fall%2Flist)

Бурих Тимур Ярославович

Група № 5

Курс 1

Факультет: інженерії програмного забезпечення

Варiант 6

Задача 1

Вот код на C#, который находит сумму чисел 1+3+5+7+...+(2n-1):

int n = 10; // здесь мы выбираем значение n

int sum = 0;

for (int i = 1; i <= 2\*n-1; i += 2)

{

sum += i;

}

Console.WriteLine(sum); // выводим сумму в консоль

В этом коде мы сначала выбираем значение n, равное 10, но вы можете заменить его на любое другое число. Затем мы инициализируем переменную sum нулевым значением, которое будет увеличиваться с каждой итерацией цикла. В цикле for мы перебираем все нечётные числа от 1 до 2n-1 (включительно) и добавляем их к sum. Наконец, мы выводим сумму в консоль с помощью метода Console.WriteLine.

Задача 2

using System;

namespace ConsoleApp

{

class Program

{

static void Main(string[] args)

{

int n = 1;

double eps = 0.000001;

double sum = 0;

double term = 0;

do

{

term = Math.Pow(-1, n) \* (1 - Math.Pow(n + 1, 2) / Math.Pow(n + 2, 2));

sum += term;

n++;

} while (Math.Abs(term) > eps);

Console.WriteLine($"Сумма членов ряда с точностью до {eps} равна {sum}");

sum = 0;

n = 1;

while (n <= 10)

{

term = Math.Pow(-1, n) \* (1 - Math.Pow(n + 1, 2) / Math.Pow(n + 2, 2));

sum += term;

n++;

}

Console.WriteLine($"Сумма первых 10 членов ряда равна {sum}");

Console.ReadKey();

}

}

}

В этом коде мы сначала инициализируем необходимые переменные: n - номер члена ряда, eps - желаемая точность, sum - сумма членов ряда, term - значение текущего члена ряда.

Затем мы запускаем цикл do-while, в котором вычисляем каждый член ряда с помощью заданной формулы и добавляем его к сумме. Цикл продолжается до тех пор, пока модуль текущего члена ряда не станет меньше, чем желаемая точность eps.

После этого мы выводим в консоль значение суммы членов ряда с заданной точностью.

Затем мы запускаем второй цикл while, в котором вычисляем первые 10 членов ряда и добавляем их к сумме. После этого мы выводим в консоль значение суммы первых 10 членов ряда.

Оба значения выводятся в консоль с помощью метода Console.WriteLine. Код ожидает нажатия любой клавиши, прежде чем завершиться, с помощью метода Console.ReadKey().

Задача 3

using System;

class Program

{

static void Main(string[] args)

{

double sum = 0;

double currentTerm = 1;

int negativeCount = 0;

double maxNumber = currentTerm;

for (int n = 1; Math.Abs(currentTerm) >= 0.000001; n++)

{

sum += currentTerm;

currentTerm = Math.Pow(-1, n) \* (1 - Math.Pow((n + 1), 2) / Math.Pow((n + 2), 2));

if (currentTerm < 0)

{

negativeCount++;

}

if (currentTerm > maxNumber)

{

maxNumber = currentTerm;

}

}

Console.WriteLine($"Сумма ряда: {sum}");

Console.WriteLine($"Количество отрицательных чисел: {negativeCount}");

Console.WriteLine($"Максимальное число: {maxNumber}");

}

}

В этом коде мы используем цикл for, который начинается с n = 1 и продолжается до тех пор, пока текущий член ряда (вычисленный по формуле) не станет меньше 0.000001 (по модулю). Внутри цикла мы добавляем текущий член к сумме ряда, вычисляем следующий член по формуле и проверяем, является ли текущий член отрицательным или большим, чем максимальное число. Если да, то мы обновляем соответствующие переменные.

Наконец, мы выводим значение суммы ряда, количество отрицательных чисел и максимальное число.

Задача 4

Один з можливих способів розв’язання цієї задачі - перебрати всі можливі комбінації натуральних чисел 𝑥, 𝑦 та 𝑘, які належать інтервалу від 1 до 30, та перевірити, чи задовольняють вони рівняння 𝑥2+𝑦2=𝑘2. Якщо так, то вивести їх на екран. Для цього можна використати дві вкладені логічні петлі for, які будуть перебирати всі можливі комбінації чисел 𝑥, 𝑦 та 𝑘.

Отже, код на C# може виглядати таким чином:

using System;

class Program

{

static void Main(string[] args)

{

for (int x = 1; x <= 30; x++)

{

for (int y = 1; y <= 30; y++)

{

for (int k = 1; k <= 30; k++)

{

if (x \* x + y \* y == k \* k)

{

Console.WriteLine($"{x}, {y}, {k}");

}

}

}

}

}

}

Цей код перебирає всі можливі комбінації чисел 𝑥, 𝑦 та 𝑘 від 1 до 30, перевіряє, чи задовольняють вони рівняння 𝑥2+𝑦2=𝑘2, і якщо так, то виводить їх на екран у вигляді пар чисел 𝑥 та 𝑦, та числа 𝑘. Цей код може працювати тривалий час, оскільки перебирає дуже багато комбінацій, тому можна додатково використати оптимізації, які дозволять прискорити роботу програми, наприклад, обмежити перебір деякими умовами.