ВШЭ АиСД 2021. Повторение C++, часть 1.

12 янв 2022, 18:04:09
старт: 3 сен 2021, 11:30:00
финиш: 12 сен 2021, 23:59:59
длительность: 9д. 12ч.
начало: 3 сен 2021, 11:30:00
конец: 12 сен 2021, 23:59:59

F. Вычисление синуса с помощью ряда Маклорена (0.2)

Ограничение времени	1 секунда
Ограничение памяти	64.0 Мб
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

Известно разложение тригонометрической функции $sin(x)$ для x, лежащего в интервале $[-\frac{\pi}{2}, \frac{\pi}{2}]$, в ряд Маклорена:
$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \frac{x^9}{9!} - ...$

Требуется разработать программу, которая вычисляет значение функции $\sin x$ для любого значения X, т.е. не только из интервала $[-\frac{\pi}{2}, \frac{\pi}{2}]$.
Вычисления ограничиваются точностью e (от 10^{-1} до 10^{-6}).
Напомним, что функция синуса является периодической, т.е.:
$\sin x = \sin(x + 2\pi n)$

Кроме того, справедливы формулы приведения:
$\sin(\pi + x) = -sin(x)$ и $\sin(\pi - x) = sin(x)$.

В расчетах примем $\pi = 3.14159265$.

Формат ввода

В первой строке – значение аргумента X. Целая часть аргумента содержит до семи цифр.
Во второй строке – значение точности e (от 10^{-1} до 10^{-6}).

Формат вывода

В первой строке значение суммы (вывести только k знаков после запятой, где k - число знаков после запятой в точности e.
Во второй строке вывести номер последнего добавленного слагаемого.

Пример 1

Ввод	Вывод
10 0.1	-0.6 1

Пример 2

Ввод	Вывод
0 0.000001	0 1

Набрать здесь | Отправить файл

```cpp
/*Задача 6. Вычисление синуса с помощью ряда Маклорена*/

#include <iostream>
#include <math.h>
#include <cmath>
#include <iomanip>

using namespace std;

// Сначала делим и получаем целое число - количество раз, которое необходимо вычесть из x.
// Затем вычитаем это число раз из x и проверяем меньше или равно данное число, чем pi/2.
// Если да, то все оставляем как есть, иначе отнимаем еще pi и увеличиваем число вычитаний.
// Если число вычитаний четное, то знак плюс, иначе минус.
double FindInterval(double x)
{
    const double Pi = 3.14159265;

    int countSubstraction = x / Pi;

    x -= Pi * countSubstraction;

    if (x > Pi / 2)
    {
        x -= Pi;
        countSubstraction++;
    }

    if (x < -Pi / 2)
    {
        x += Pi;
        countSubstraction++;
    }

    return x * pow(-1, countSubstraction);
}

// Считаем в конце цикле ряд по общей формуле, до того момента, пока не увидим
```

Отправить

Предыдущая