МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ

БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ОБРАЗОВАНИЯ

ВЯТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Факультет автоматики и вычислительной техники

Кафедра радиоэлектронных средств

Отчет по дисциплине

«Цифровые устройства и микропроцессоры»

Лабораторная работа №4

«МОДУЛЬНОЕ ПРОГРАММИРОВАНИЕ»

Вариант №12

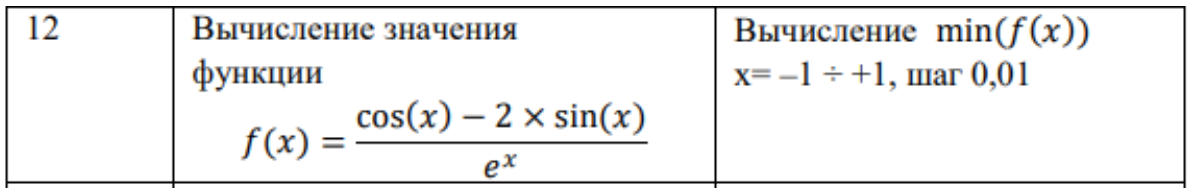
Выполнил: студент группы ИНБб – 3301\_\_\_\_\_\_\_\_\_\_\_\_\_\_В.С. Голубева

Проверил: к.т.н. доцент кафедры РЭС \_\_\_\_\_\_\_\_\_\_\_\_\_\_М. А. Земцов

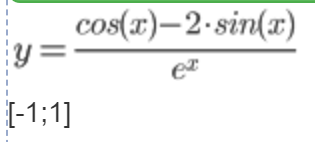
Киров 2023

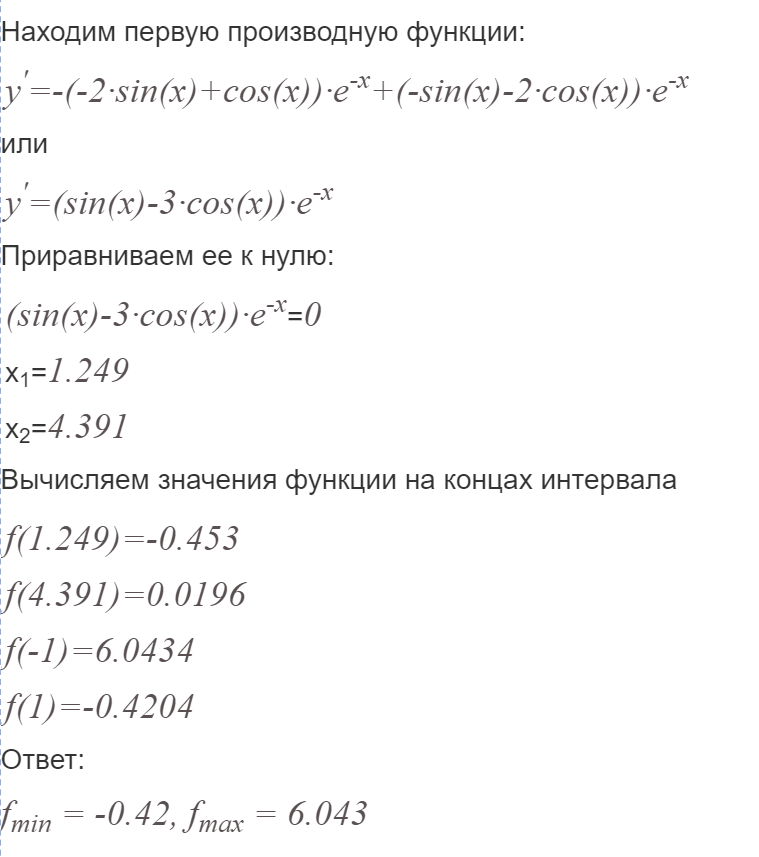
Задание:

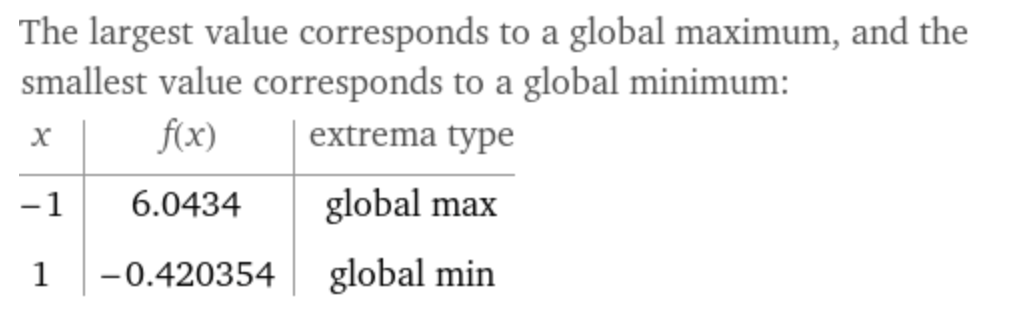




Расчет с помощью калькулятора:

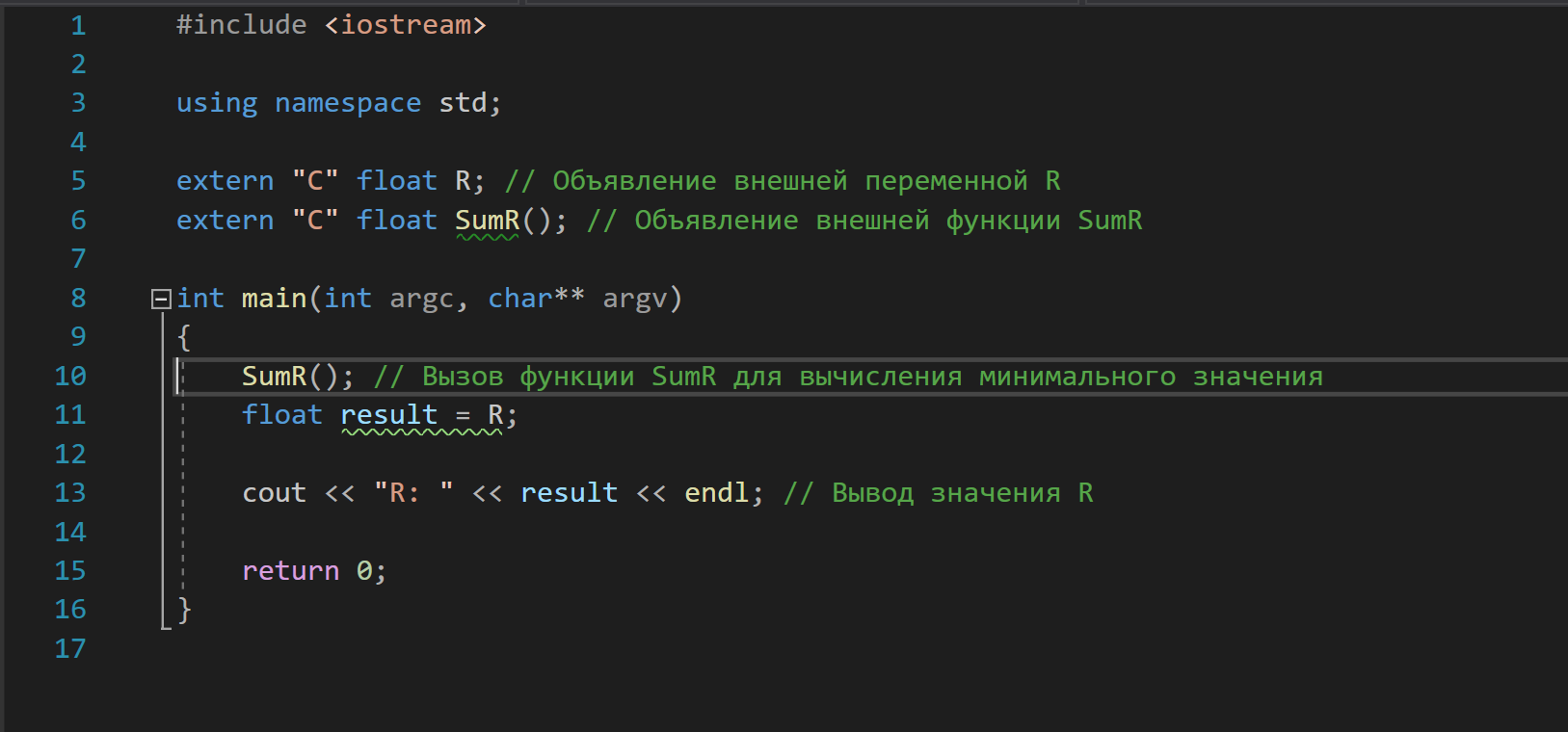


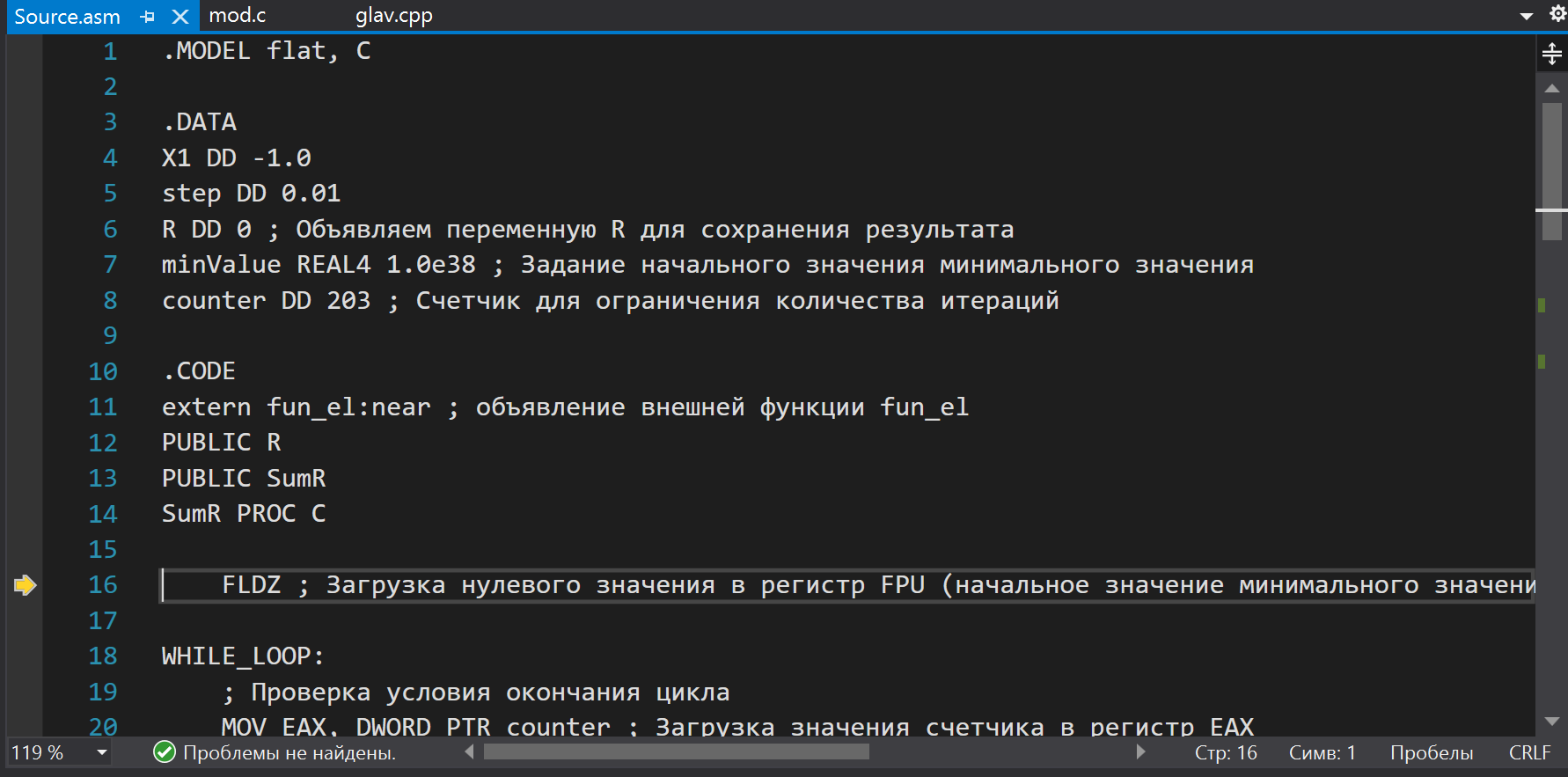




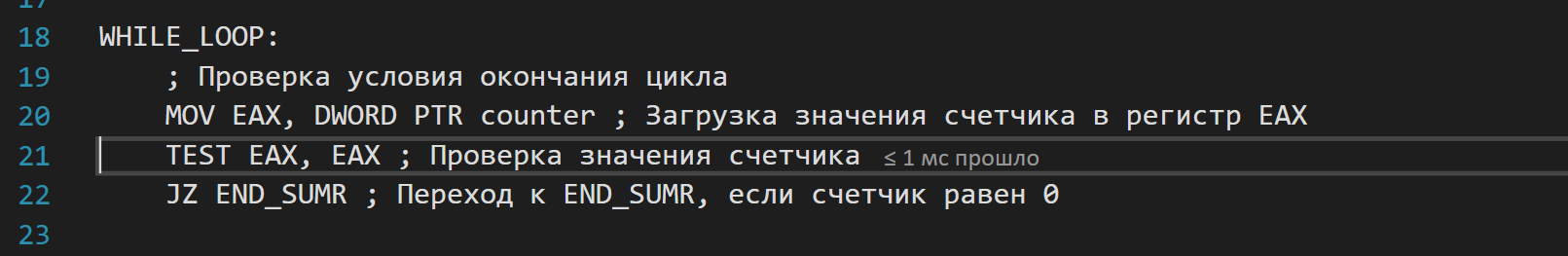
**Работа программы:**

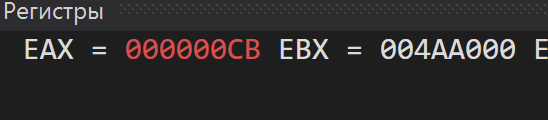
Из glav.cpp вызывается функция SumR, которая находится в Source.asm





Заход в цикл, загрузка количества итераций в EAX, сравнение с 0.При нуле переход к END\_SUMR



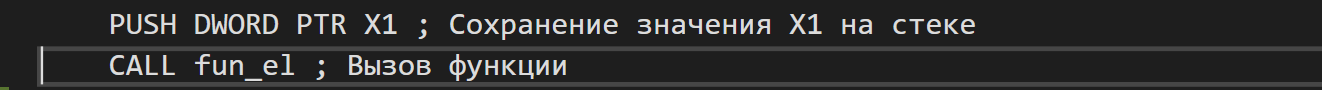


Загрузка значения X в ST(0)

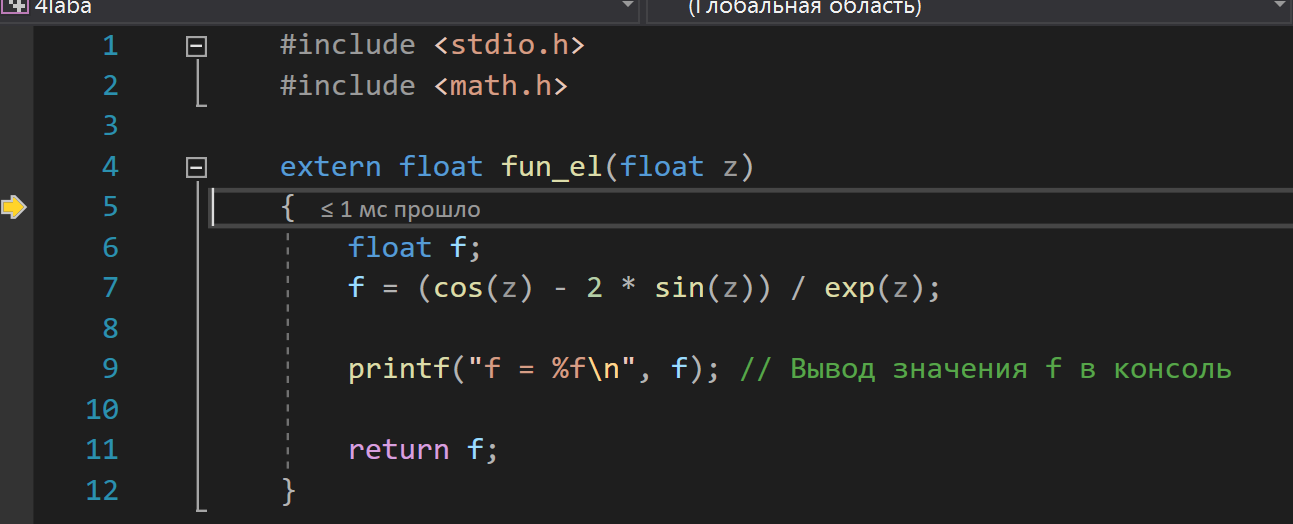




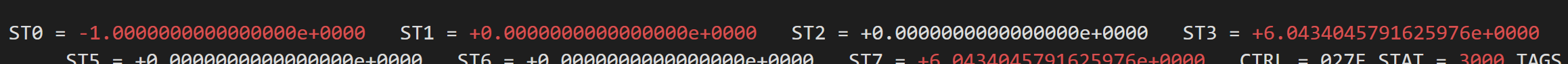
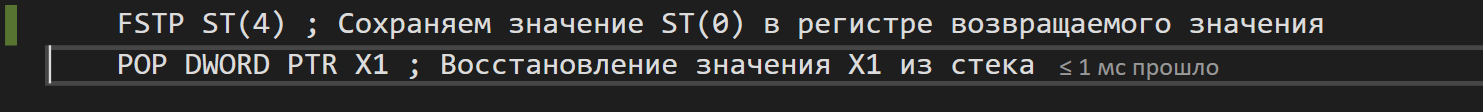
Вызов функции fun\_el, переход в mod.с



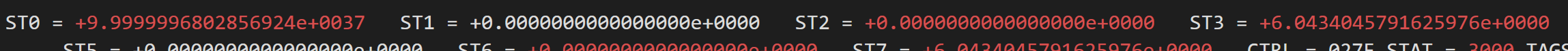
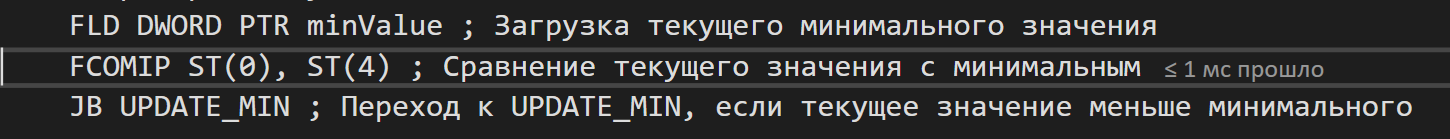
Расчёт значения функции

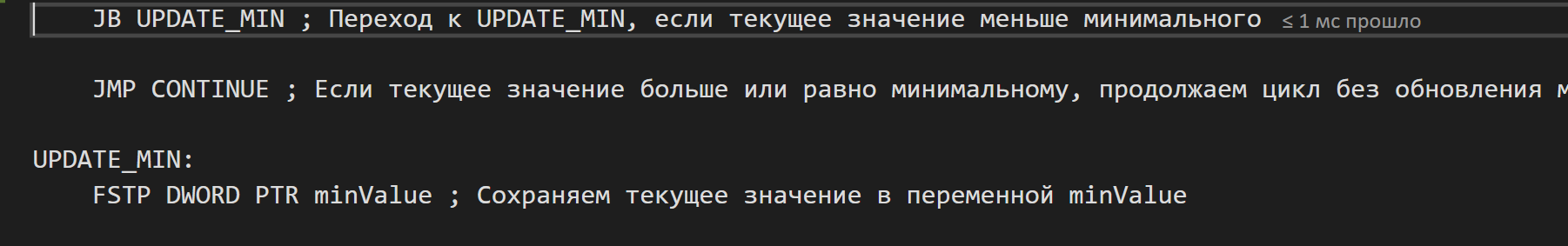
Переход к Source.asm, сохранения значения в ST(4), возвращение Х в ST(0)



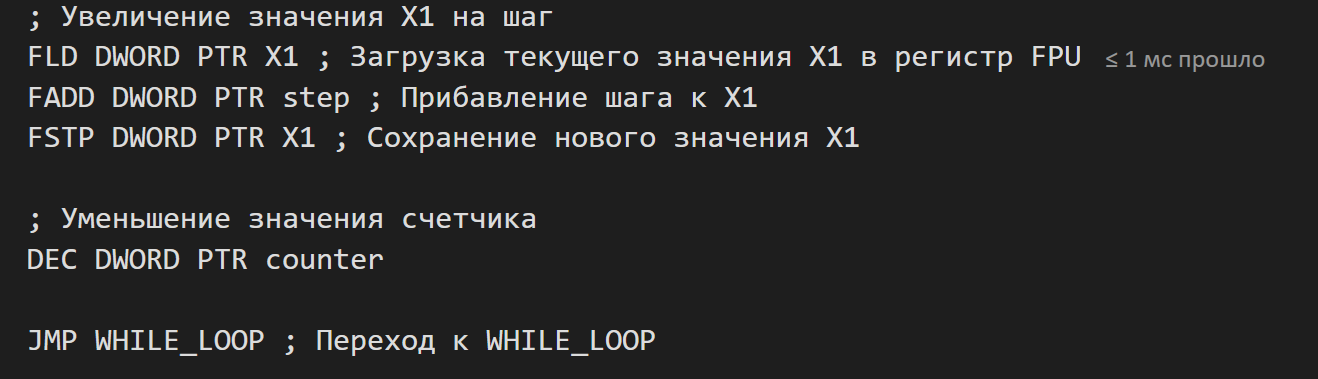
Сравнение текущего значения с минимальным



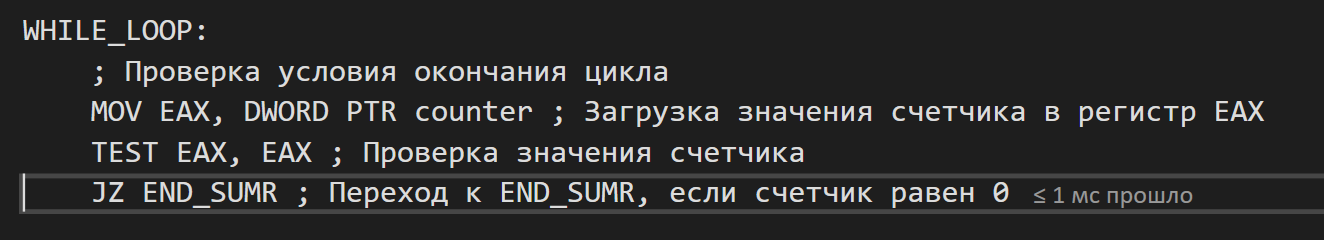
Если минимальное значение больше, чем рассчитанное, то переходим к UPDATE\_MIN

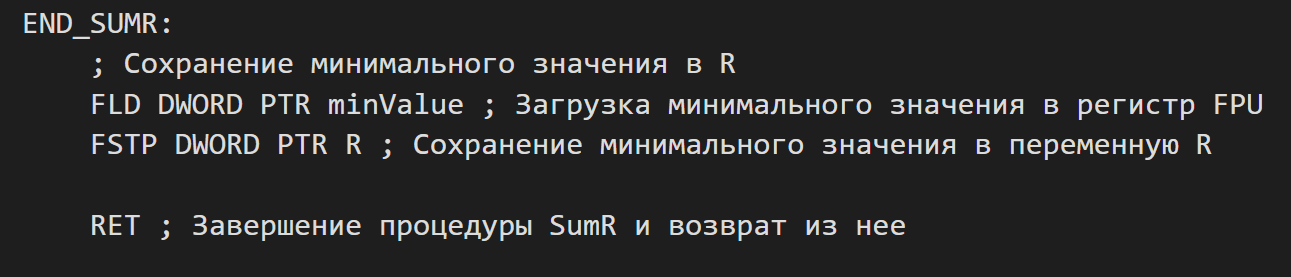


Прибавляем шаг к значению Х, сохраняем данное значение и уменьшаем счетчик, переходим заново к WHILE\_LOOP рассчитывать новое значение.

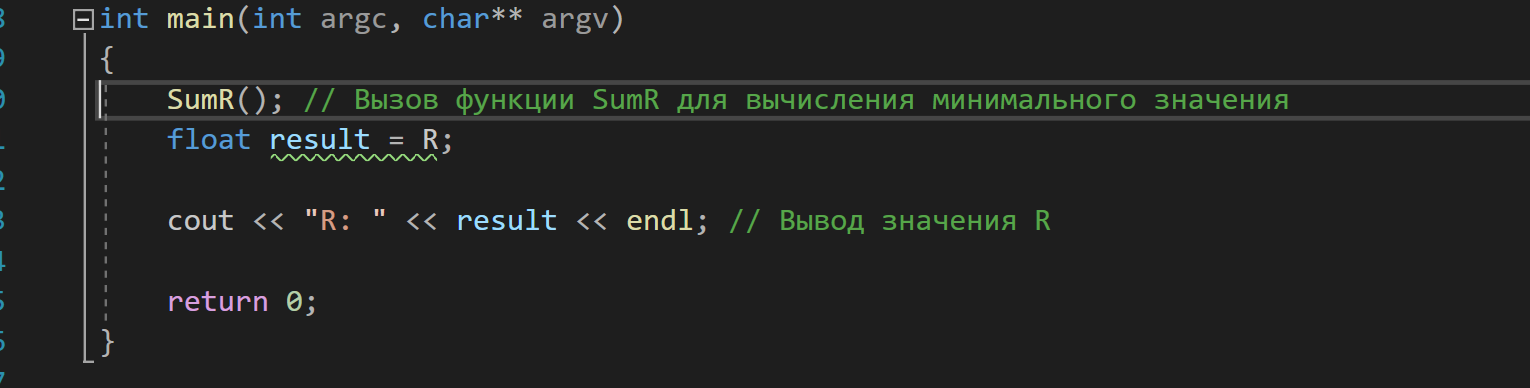


Если счетчик равен 0, в начале нового расчета, то сохраняем минимальное значение в r, и переходим в glav.cpp





Вывод минимального значения





Результаты работы программы сходятся с результатами вычисления.

Программный код:

**glav.cpp**

#include <iostream>

using namespace std;

extern "C" float R; // Объявление внешней переменной R

extern "C" float SumR(); // Объявление внешней функции SumR

int main(int argc, char\*\* argv)

{

SumR(); // Вызов функции SumR для вычисления минимального значения

float result = R;

cout << "R: " << result << endl; // Вывод значения R

return 0;

}

**mod.c**

#include <stdio.h>

#include <math.h>

extern float fun\_el(float z)

{

float f;

f = (cos(z) - 2 \* sin(z)) / exp(z);

printf("f = %f\n", f); // Вывод значения f в консоль

return f;

}

**Source.asm**

.MODEL flat, C

.DATA

X1 DD -1.0

step DD 0.01

R DD 0 ; Объявляем переменную R для сохранения результата

minValue REAL4 1.0e38 ; Задание начального значения минимального значения

counter DD 203 ; Счетчик для ограничения количества итераций

.CODE

extern fun\_el:near ; объявление внешней функции fun\_el

PUBLIC R

PUBLIC SumR

SumR PROC C

FLDZ ; Загрузка нулевого значения в регистр FPU (начальное значение минимального значения)

WHILE\_LOOP:

; Проверка условия окончания цикла

MOV EAX, DWORD PTR counter ; Загрузка значения счетчика в регистр EAX

TEST EAX, EAX ; Проверка значения счетчика

JZ END\_SUMR ; Переход к END\_SUMR, если счетчик равен 0

FLD DWORD PTR X1 ; Загрузка текущего значения X1 в регистр FPU

; Остальные действия цикла

PUSH DWORD PTR X1 ; Сохранение значения X1 на стеке

CALL fun\_el ; Вызов функции

FSTP ST(4) ; Сохраняем значение ST(0) в регистре возвращаемого значения

POP DWORD PTR X1 ; Восстановление значения X1 из стека

; Получение возвращаемого значения из функции fun\_el

FSTP ST(0) ; Сохраняем значение ST(0) в регистре возвращаемого значения

; Проверка текущего значения и обновление минимального значения

FLD DWORD PTR minValue ; Загрузка текущего минимального значения

FCOMIP ST(0), ST(4) ; Сравнение текущего значения с минимальным

JB UPDATE\_MIN ; Переход к UPDATE\_MIN, если текущее значение меньше минимального

JMP CONTINUE ; Если текущее значение больше или равно минимальному, продолжаем цикл без обновления минимального значения

UPDATE\_MIN:

FSTP DWORD PTR minValue ; Сохраняем текущее значение в переменной minValue

CONTINUE:

; Увеличение значения X1 на шаг

FLD DWORD PTR X1 ; Загрузка текущего значения X1 в регистр FPU

FADD DWORD PTR step ; Прибавление шага к X1

FSTP DWORD PTR X1 ; Сохранение нового значения X1

; Уменьшение значения счетчика

DEC DWORD PTR counter

JMP WHILE\_LOOP ; Переход к WHILE\_LOOP

END\_SUMR:

; Сохранение минимального значения в R

FLD DWORD PTR minValue ; Загрузка минимального значения в регистр FPU

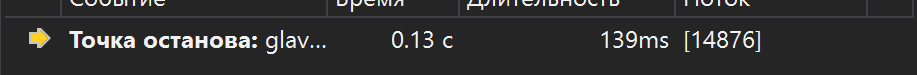
FSTP DWORD PTR R ; Сохранение минимального значения в переменную R

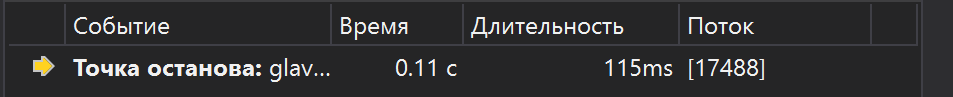
RET ; Завершение процедуры SumR и возврат из нее

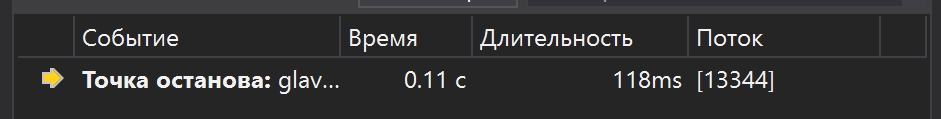
SumR ENDP ; Завершение определения

End

Время исполнения программы, написанной с помощью модульного программирования : 124 мс







Время исполнения программы на с++ : 259 мс

