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»

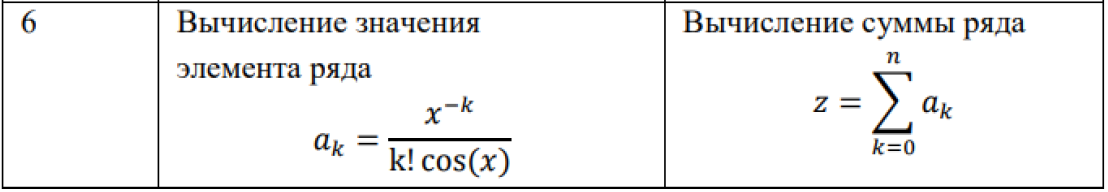
Вариант №6

Выполнила: студентка группы ИКБб– 31\_\_\_\_\_\_\_\_\_\_\_\_\_\_ М. А. Дубровин

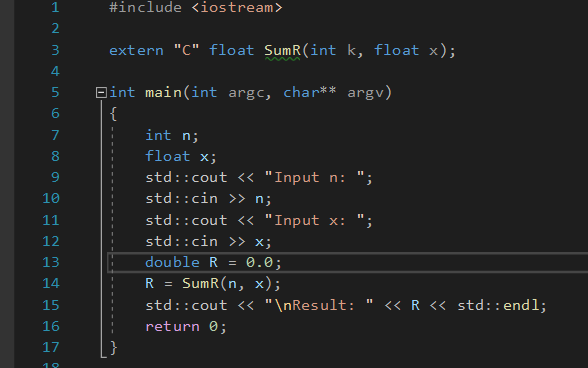
Проверил: доцент кафедры РЭС \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ М. А. Земцов

#### Киров 2022

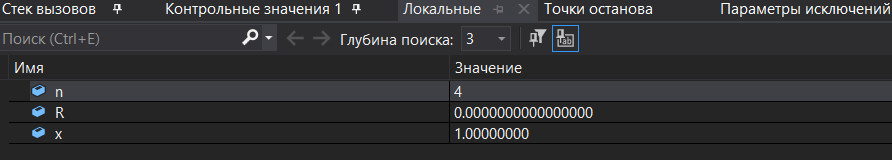
Задание:

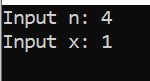


Работа программы:

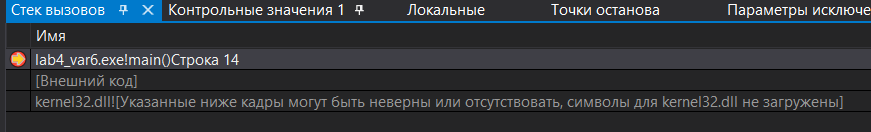


Главная программа

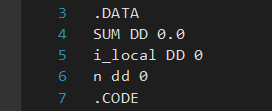




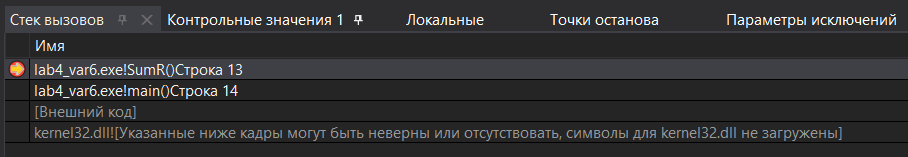
Ввели данные



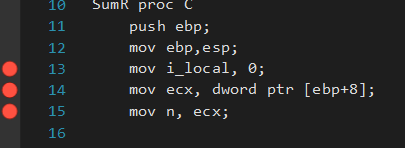
Стек вызовов перед вызовом функции ассемблера



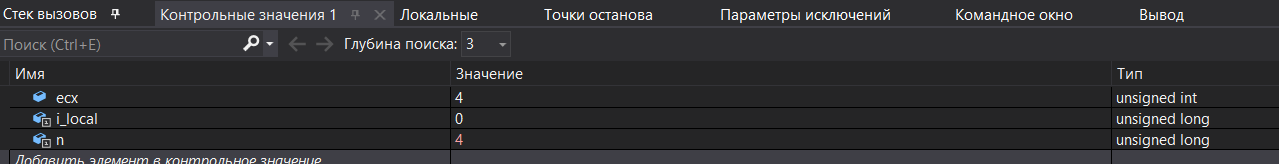
Инициализация данных в ассемблере

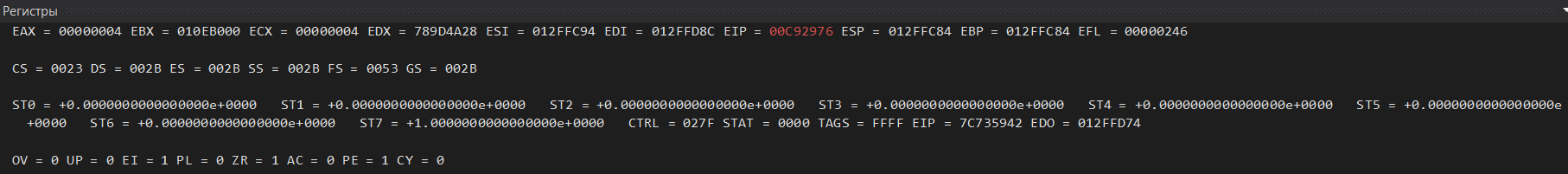


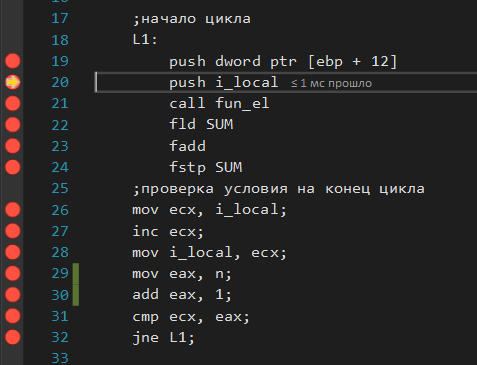
Стек вызовов после вызова функции ассемблера



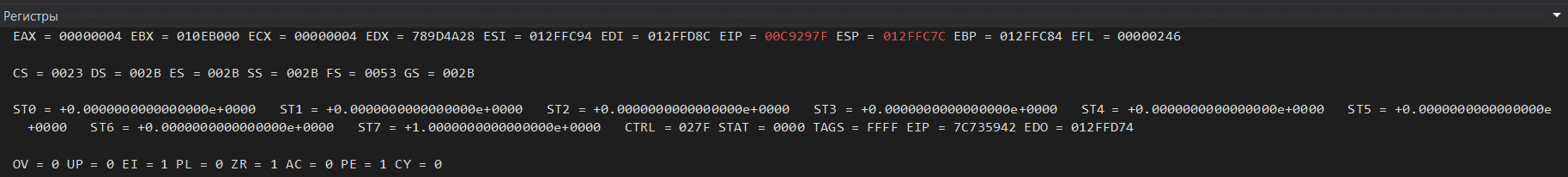
Помещаем в i\_local 0, а в nчисло итераций из полученного параметра



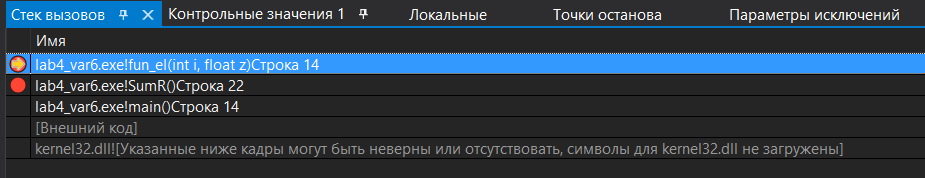




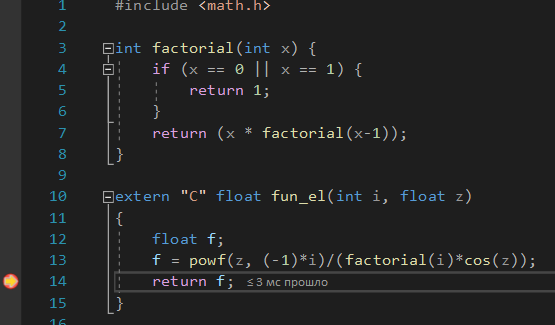
Начинаем цикл

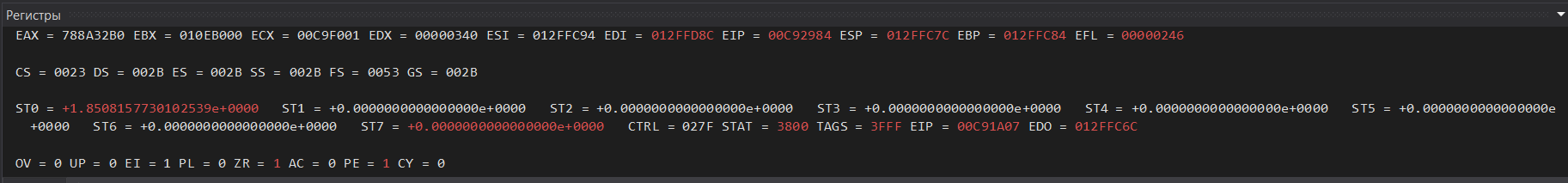


Передали в функцию fun\_elтекущее значение счётчика и аргумент x



Стек вызовов, после вызова функции fun\_el

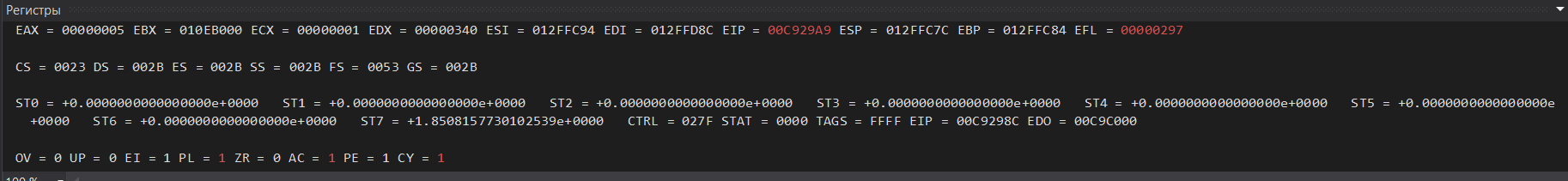


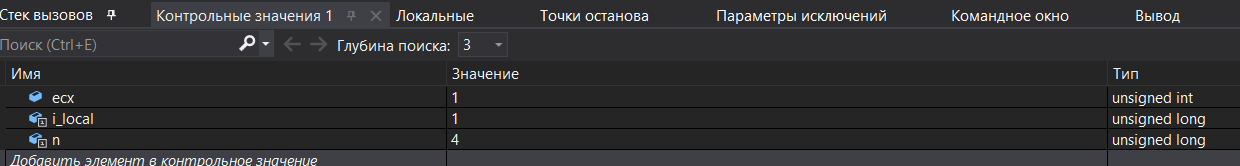


После выполнение функции fun\_elв st(0) поместилось число.

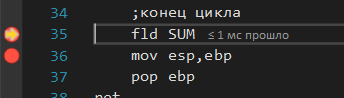
Складываем его с SUM

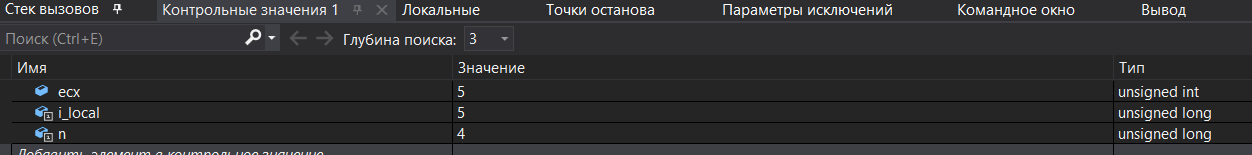
И так как счётчик итераций меньше заданного числа итераций цикла, то продолжаем цикл

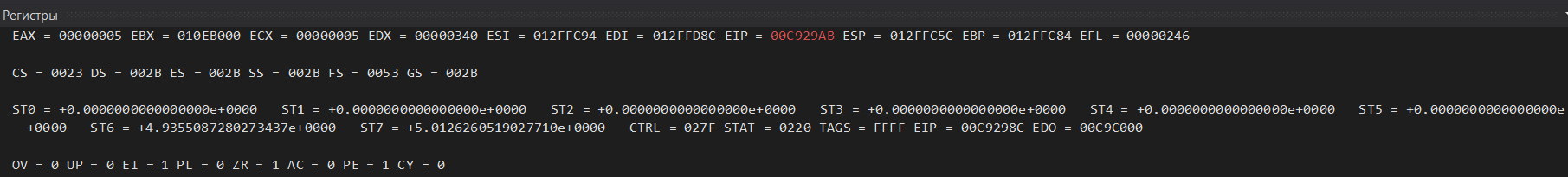


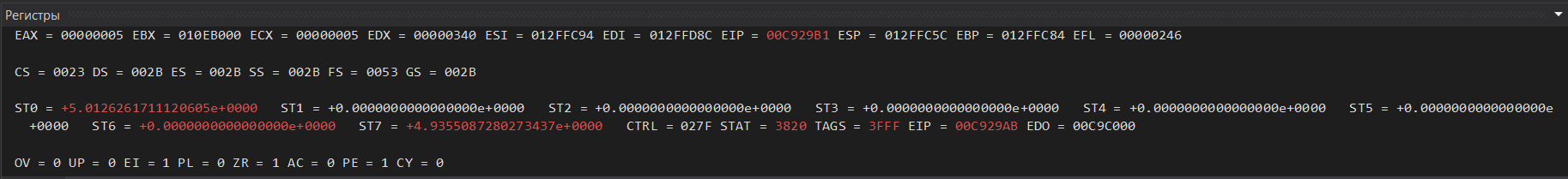


Помещаем результат из SUMв st(0) для возврата в C++

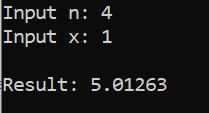








Результат работы совпадает с результатом работы программы на Python



Программныйкод:

Lab4\_var6.cpp

#include<iostream>

extern"C"floatSumR(intk, floatx);

intmain()

{

int n;

float x;

std::cout<<"Input n: ";

std::cin>> n;

std::cout<<"Input x: ";

std::cin>> x;

double R = 0.0;

R = SumR(n, x);

std::cout<<"\nResult: "<< R << std::endl;

return 0;

}

Fun\_as.asm

.586

.MODELflat,C

.DATA

SUM DD 0.0

i\_local DD 0

n dd 0

.CODE

extern fun\_el:near ; объявлениевнешнейфункцииfun\_el

public SumR

SumR proc C

push ebp;

mov ebp,esp;

mov i\_local, 0;

mov ecx, dwordptr [ebp+8];

mov n, ecx;

;начало цикла

L1:

push dwordptr [ebp + 12]

push i\_local

call fun\_el

fld SUM

fadd

fstp SUM

;проверка условия на конец цикла

movecx, i\_local;

incecx;

mov i\_local, ecx;

mov eax, n;

add eax, 1;

cmpecx, eax;

jne L1;

;конеццикла

fld SUM

mov esp,ebp

pop ebp

ret

SumRendp

End

Fun\_c.cpp

#include<math.h>

intfactorial(intx) {

if (x == 0 || x == 1) {

return 1;

}

return (x \* factorial(x-1));

}

extern"C"floatfun\_el(inti, floatz)

{

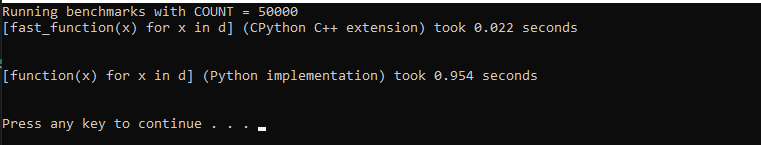
float f;

f = powf(z, (-1)\*i)/(factorial(i)\*cos(z));

return f;

}

Проверка на быстроту



Как видим, код на питоне с модулем на языке Си работает в несколько раз быстрее чем чистый код на питоне, так как модуль выполняет роль ускорителя в вычислениях, задействуя лишь необходимые для конкретной операции ресурсы и функции.