МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ

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 14

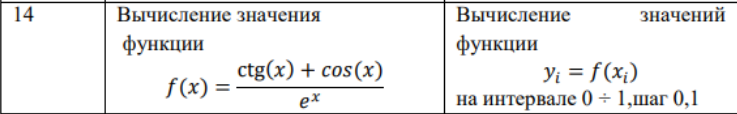
Выполнил: студент группы ИКТб-3301 \_\_\_\_\_\_\_\_\_\_\_\_\_ / А.С. Шабардин /

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

Киров 2023

**Цель работы:** знакомство с технологией применения языка ассемблера при разработке программного обеспечения на языках высокого уровня.

**Индивидуальное задание:**



**Верификация программы:**

Расчёт на ассемблере и проверка на C:

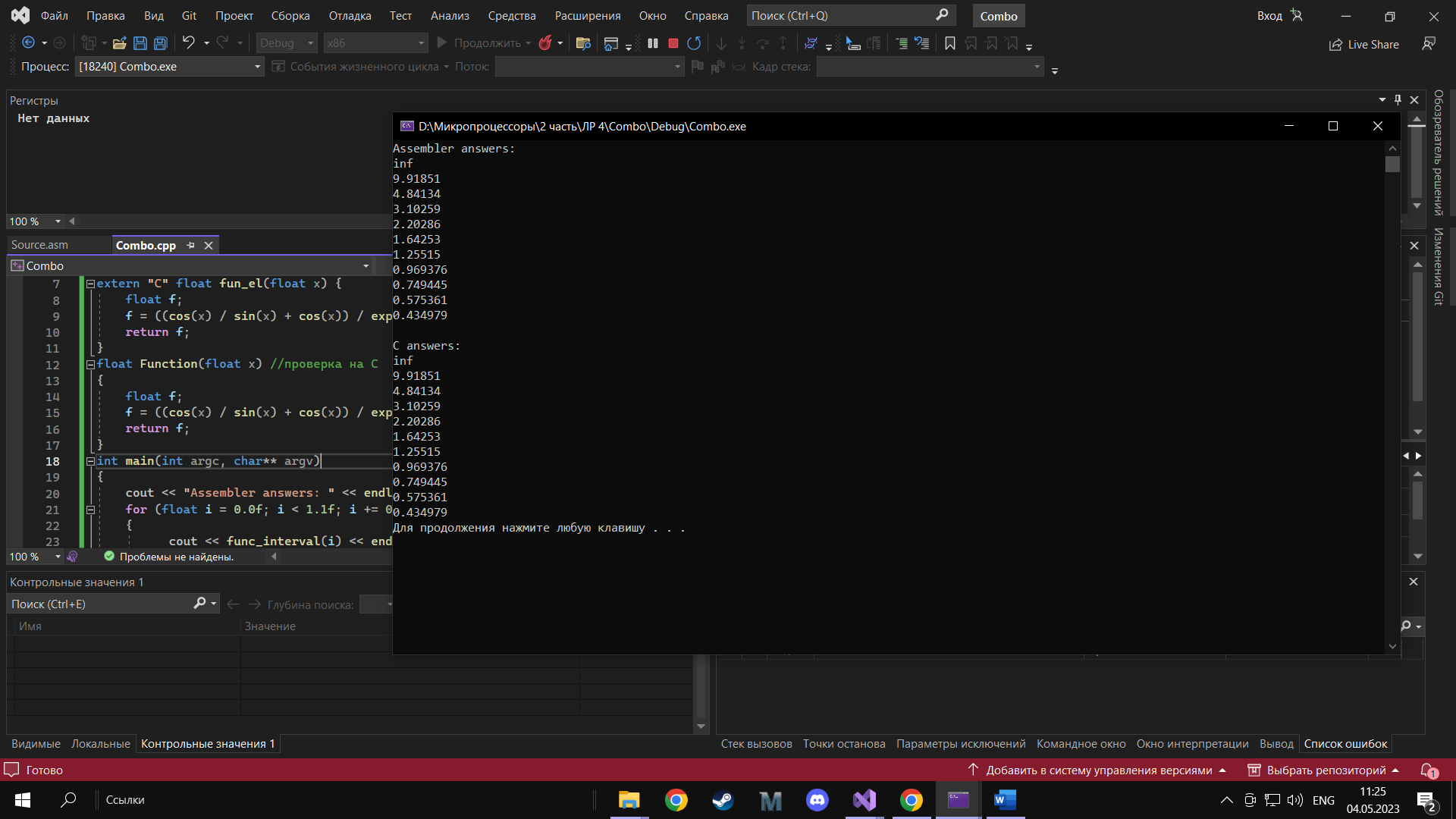


Рисунок 1 – Расчеты на ассемблере и на C

На рисунке 2 представлена проверка в математическом пакете Mathcad.

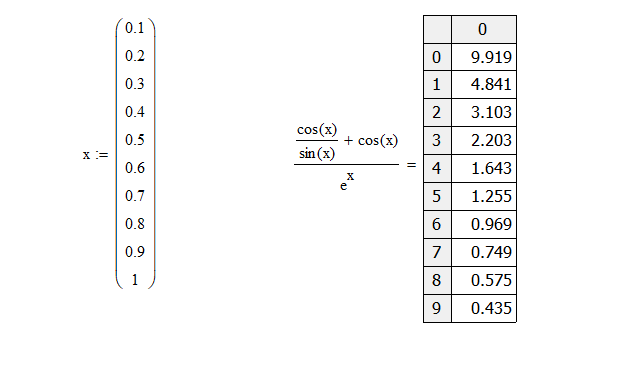


Рисунок 2 – Проверка в Mathcad

Т.к. котангенс от 0 не определён (ctg(0) = cos(0)/sin(0) = 1/0), то Mathcad его посчитать не может.

**Текст программы:**

1) Combo.cpp – Главный файл

#include <iostream>

#include <math.h>

using namespace std;

extern "C" float func\_interval(float x);// функция на ассемблере

extern "C" float fun\_el(float x) {

float f;

f = ((cos(x) / sin(x) + cos(x)) / exp(x));

return f;

}

float Function(float x) //проверка на C

{

float f;

f = ((cos(x) / sin(x) + cos(x)) / exp(x));

return f;

}

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

{

cout << "Assembler answers: " << endl;

for (float i = 0.0f; i < 1.1f; i += 0.1f)

{

cout << func\_interval(i) << endl;

}

cout << endl << "C answers: " << endl;

for (float i = 0.0f; i < 1.1f; i += 0.1f)

{

cout << Function(i) << endl;

}

system("pause");

return 1;

}

2) Source.asm – файл ассемблера

.586

.MODEL flat,C

.DATA

.CODE

extern fun\_el:near ; объявление внешней функции

public func\_interval

func\_interval proc C

push ebp

mov ebp,esp

push dword ptr [ebp+8]

call fun\_el

mov esp,ebp

pop ebp

ret; выход из подпрограммы

func\_interval endp

End

**Вывод:** результат расчёта в C, результат расчёта в C с использованием подпрограммы на ассемблере и расчёт с помощью математического пакета Mathcad сошлись.

**Исследование работы программы на Python с использованием dll и без ее использования.**

Для проверки скорости выполнения программы была написана динамическая библиотека с функцией HardCount(), которая считает сумму . Время выполнения программы с использованием dll и без представлено на рисунках 3-4 (первое число – результат вычисления, второе число – время выполнения программы в секундах).



Рисунок 3 – Время выполнения программы с использованием dll

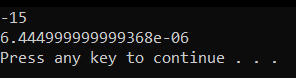


Рисунок 4 – Время выполнения программы без использования dll

Вывод: Как показывают результаты: использование dll не ускоряет работу программы, а , наоборот, замедляет ее. Время выполнения программы с использованием dll в 24 раза превышает время выполнения программы без использования dll. Замеры времени проводились с помощью библиотеки timeit. Для усреднения значений было проведено 100 замеров и взято среднее значение.