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 №5

|  |  |  |
| --- | --- | --- |
| Выполнил: студент гр. ИКТб-3301 | \_\_\_\_\_\_\_\_\_\_\_\_\_\_ | А.В.Герасименко |
| Проверил: доцент | \_\_\_\_\_\_\_\_\_\_\_\_\_\_ | М.А. Земцов |

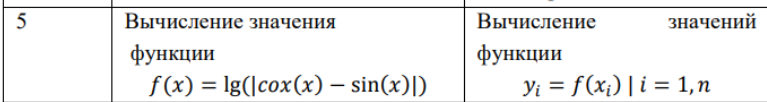
Киров 2023

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

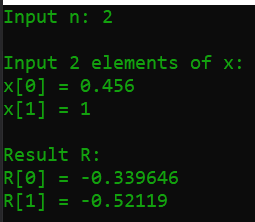
**Задание:**

В таблице 1 представлено задание.

Таблица 1

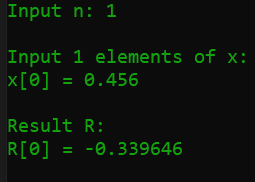


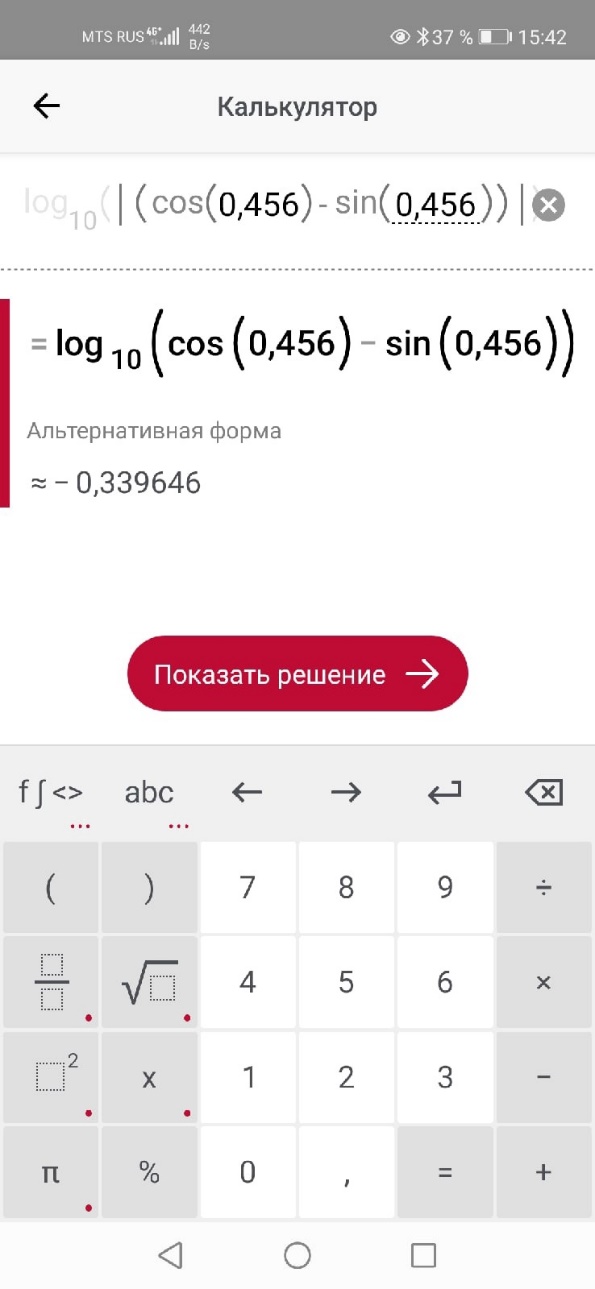
**Ход работы:**



Вводим изначально количество повторений, а далее считаются значения по формуле и записываются в массив, созданный в файле Source.cpp

Проверка:





**Код програмы:**

Source.cpp

#include <iostream>

#include "Source.h"

extern "C" float\* SumR(int, float\*);

int main() {

int n = 0;//число элементов массива

float\* x = nullptr;//массив иксов

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

std::cin >> n;

x = new float[n];

std::cout << "\nInput " << n << " elements of x:\n";

for (int i = 0; i < n; i++) {

std::cout << "x[" << i << "] = ";

std::cin >> x[i];//получаем иксы

}

float\* R = nullptr;

R = SumR(n, x);//получаем из ассемблера реузальтирующий массив

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

for (int i = 0; i < n; i++) {

std::cout << "R[" << i << "] = " << R[i] << std::endl;//вывод на экран

}

return 0;

}

**Header.h**

#include <math.h>

extern "C" float fun\_el(float z)

{

float f = 0;

f = log10(fabs(cos(z) - sin(z)));//lg(abs(cos(x) - sin(x)))

return f;

}

**S1.asm**

.586

.MODEL flat,C

.DATA

SUM dd 0.0;переменная в которую мы поместим результат функции на C

i\_local DD 0;счётчик

n dd 0;число итераций

.CODE

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

public SumR

SumR proc C

push ebp;

mov ebp,esp;

mov i\_local,1;помещаем в счётчик 1

mov ecx, dword ptr [ebp+8];помещаем в ecx число итераций цикла n

mov n, ecx;поместили в локальную переменную "n" число итераций из ecx

mov esi, [ebp + 12];поместили адрес массива в esi

;начало цикла

L1:

mov ebx, i\_local;поместили в ebx текущий счётчик

sub ebx, 1;вычли 1, чтобы использовать в обращении к элементам массива

push dword ptr [esi + ebx\*4];передаём текущий элемент массива в функцию fun\_el

call fun\_el;вызываем функцию fun\_el

fld SUM;помещаем в st(0) SUM

fadd;складываем st(0) и st(1)

inc i\_local;увеличиваем счётчик на 1

fstp SUM;помещаем значение из st(0) в SUM

mov ecx, SUM;помещаем SUM в ecx

mov [esi + ebx\*4], ecx;помещаем в текущий элемент массива значение из ecx

mov SUM, 0;обнуляем SUM

mov eax, n;помещаем n в eax

inc eax;увеличиваем n на 1

mov ebx, i\_local;помещаем в ebx, текущее значение счётчика

cmp ebx, eax;сравниваем текущее значение счётчика и n

jne L1;если они не равны переходим к L1

;конец цикла

mov eax, esi;помещаем адрес массива в eax для возврата в C++

mov esp,ebp;

pop ebp;

ret

SumR endp

End

**Вывод:**

В ходе данной лабораторной работы изучена технология применения языка ассемблера при разработке программного обеспечения на языках высокого уровня.

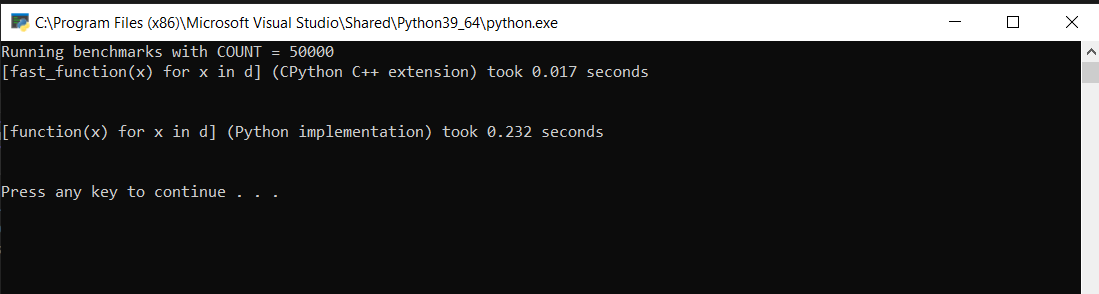
****

Рисунок 1 - Сравнение по времени выполнения программ, написанных на Python и Python с модулем ускорения на C++