Министерство образования Республики Беларусь

Учреждение образования Белорусский государственный университет информатики и радиоэлектроники

Кафедра ЭВМ

Отчёт по лабораторной работе №6

“Интерфейс с языками высокого уровня. Работа с математическим

сопроцессором”

Проверил: Выполнил:

ассистент каф. ЭВМ студент гр.030501

Туровец Николай Олегович Вайтехович П.В.

Минск 2022

**Цель работы:** Ознакомиться с вариантами внедрения ассемблерной процедуры в программу, написанную на языке программирования C\C++, изучить архитектуру математического сопроцессора и команды работы с ним.

**Алгоритм**

* Соглашение об объединении программных модулей
* Написание функции на языке ассемблера
* Использование команд математического сопроцессора

Связь ассемблерных модулей с языками высокого уровня требует следующих соглашений, которые сильно зависят от применяемых компиляторов и операционной системы. При входе в ассемблерную процедуру в стеке будут сохранены регистры SI и DI и размещены локальные переменные х и у. Доступ к этим данным организуется с помощью адресации по базе с использованием регистра ВР. При этом нет необходимости вычислять смещения вручную, поскольку ассемблер автоматически генерирует макроподстановки. Встроенный ассемблер – вставка ассемблерного кода непосредственно в код программы на языке высокого уровня. Использование встроенного ассемблера позволяет создавать программы более быстро, используя небольшие фрагменты кода без выполнения выше изложенных требований по сборке проекта. Встроенный ассемблер – вставка ассемблерного кода непосредственно в код программы на языке высокого уровня. Использование встроенного ассемблера позволяет создавать программы более быстро, используя небольшие фрагменты кода без выполнения выше изложенных требований по сборке проекта. В командах встроенного ассемблера можно свободно использовать переменные из языка высокого уровня, так как они автоматически преобразуются в соответствующие выражения.

**Листинг программы**

#include "stdio.h"

#include "windows.h"

#define SIZE 10

float array[SIZE];

void inputArray();

void outputArray();

void asmAlgorithm();

int main() {

inputArray();

printf("Input array: \n");

outputArray();

asmAlgorithm();

printf("\nResult array: \n");

outputArray();

return 0;

}

void inputArray() {

int res;

printf("Input 10 elements: \n");

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

do {

res = scanf\_s("%f", &array[i]);

while (getchar() != '\n');

if (res != 1) printf("Invalid input\n");

} while (res != 1);

}

}

void outputArray() {

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

printf("%.3f ", array[i]);

}

}

void asmAlgorithm() {

\_\_asm {

finit

xor ecx, ecx

mov ecx, SIZE / 2

lea ebx, array

calculate\_loop:

fld[ebx]

fsin

fstp[ebx]

add ebx, 4

fld[ebx]

fcos

fstp[ebx]

add ebx, 4

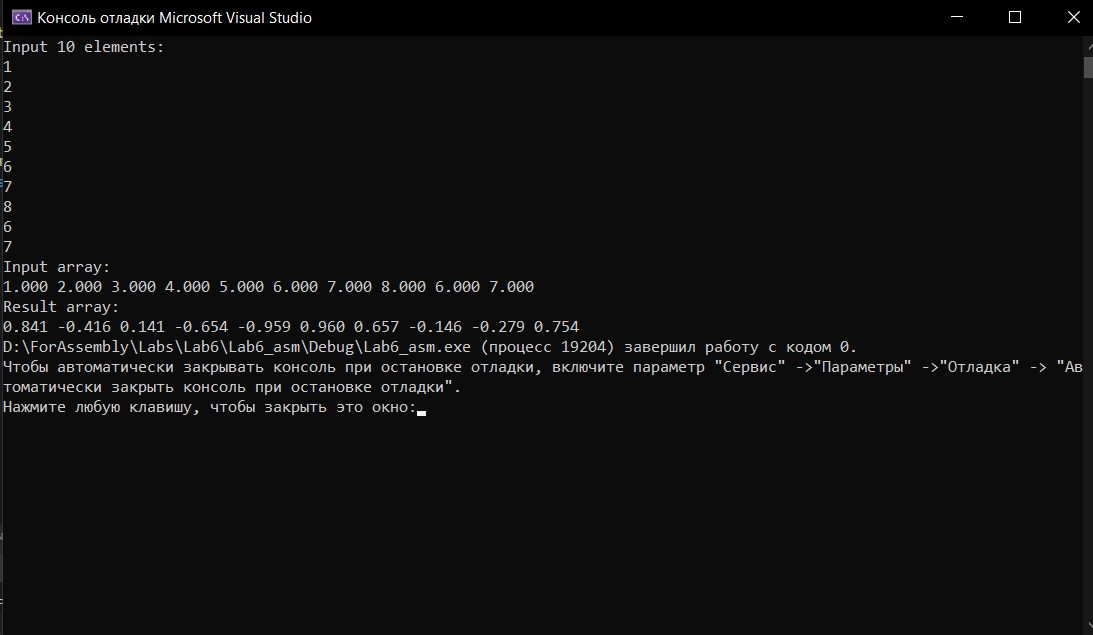
loop calculate\_loop

fwait

}

}

**ТЕСТ**

****

*Рисунок 1 –* результат выполнения программы

**Вывод:** В данной лабораторной работе разработана программа вычисления синуса и косинуса чисел.