

**МИНОБРНАУКИ РОССИИ**  
**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**  
**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**  
**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**  
**Кафедра МОЭВМ**

**ОТЧЕТ**  
**по лабораторной работе №8**  
**по дисциплине «Организация ЭВМ и систем»**  
**ТЕМА: ОБРАБОТКА ВЕЩЕСТВЕННЫХ ЧИСЕЛ. ПРОГРАММИРОВАНИЕ**  
**МАТЕМАТИЧЕСКОГО СОПРОЦЕССОРА.**

Студентка гр. 9383

\_\_\_\_\_

Сергиенкова А.А.

Преподаватель

\_\_\_\_\_

Ефремов М.А.

Санкт-Петербург

2020

### **Цель работы.**

Освоить основные принципы обработки вещественного числа.  
РЕАЛИЗОВАТЬ МАТЕМАТИЧЕСКИЙ СОПРОЦЕССОР.

### **Текст задания.**

#### **Вариант 6.**

Разработать подпрограмму на языке Ассемблера, обеспечивающую вычисление заданной математической функции с использованием математического сопроцессора. Подпрограмма должна вызываться из головной программы, разработанной на языке С. При этом должны быть обеспечены заданный способ вызова и обмен параметрами. Альтернативный вариант реализации: разработать на языке Ассемблера фрагмент программы, обеспечивающий вычисление заданной математической функции с использованием математического сопроцессора, который включается по принципу in- line в программу, разработанную на языке С.

$$\text{Acos}(x) = \text{atan}(\text{sqrt}(1 - x*x)/x)$$

### **Ход работы.**

Для реализации ввода начальных данных, а также пользовательского интерфейса, в файле `main.cpp` пользователь вводит число с плавающей точкой, которое затем проверяется на вхождение в отрезок  $[-1; 1]$ . Реализована обработка ошибок. Затем в консоль выводится значение арккосинуса с точностью до 20 знаков после запятой из стандартной библиотеки *cmath*, а после реализованная функция, которая содержит ассемблерную вставку по принципу in-line. Функция принимает и возвращает число с плавающей точкой. После вывода стандартного значения арккосинуса, печатается значение арккосинуса, тоже с точностью до 20 знаков после запятой, реализованным ассемблерной вставкой.

### Тест.

```
Enter x from [-1; 1]
0.777777777
cmath asin = 0.67967382014568078485
assembly asin = 0.67967382014568078485
```

Рисунок 1 - Тест программы

Исходный код программы представлен в приложении А.

### Выводы.

БЫЛИ ОСВОЕНЫ ОСНОВНЫЕ ПРИНЦИПЫ РАБОТЫ С ВЕЩЕСТВЕННЫМ ЧИСЛОМ.  
ТАКЖЕ БЫЛ РЕАЛИЗОВАН МАТЕМАТИЧЕСКИЙ СОПРОЦЕССОР.

## ПРИЛОЖЕНИЕ А

### ИСХОДНЫЙ КОД ПРОГРАММЫ

#### main.cpp

```
// acos(x) = atan(sqrt(1-x*x)/x)

#include <iostream>
#include <cmath>
#include <iomanip>

using namespace std;

double Acos_v2(double *xP) {
    double x = *xP;
    double x1 = 0;

    __asm {
        fld x;                // stack = x(0)
        fld x;                // stack = x(0) x(1)
        fmul;                 // stack = x * x(0)
        fld1;                 // stack = x * x(0), 1(1)
        fxch st(1);           // меняем местами : stack =
1(0), x* x(1)
        fsub;                 // вычитаем из 1 нулевой элемент
        // стэка и вставляем в нулевой элемент, stack = 1 - x*x(0)
        fsqrt;                // stack = sqrt(1 - x*x)(0)
        fld x;                // stack = x(0), sqrt(1 -
x*x)(1)
        fdiv;                 // делим первый элемент стэка на
        // нулевой. stack = sqrt(1 - x*x) / x
        fld1;                 // stack = 1(0), sqrt(1 -
x*x)(1)
        fpatan;               // вычисляем арктангенс, от
        // значения при делении первого элемента стэка на нулевой. stack = atg(sqrt(1 -
x*x) / x)
        fstp x1;              // вытащили из стэка значение
    }

    return x1;
}

int main() {
    double x;

    cout << "Enter x from [-1; 1]" << endl;
    cin >> x;
    while (x < -1 || x > 1) {
        cout << "Error" << endl;
        cin >> x;
        cout << endl;
    }
}
```

```
    cout << "cmath asin = " << setprecision(20) << acos(x) << endl;
    cout << "assembly asin = " << setprecision(20) << Acos_v2(&x) <<
endl;

    return 0;
}
```