

**МИНОБРНАУКИ РОССИИ**  
**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**  
**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**  
**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**  
**Кафедра математического обеспечения и применения ЭВМ**

**ОТЧЕТ**  
**по лабораторной работе №8**  
**по дисциплине «Организация ЭВМ и систем»**  
**Тема: Обработка вещественных чисел. Программирование**  
**математического сопроцессора.**

Студент гр. 9383

\_\_\_\_\_

Арутюнян С.Н.

Преподаватель

\_\_\_\_\_

Ефремов М.А.

Санкт-Петербург

2020

# Содержание

1. Цель работы.....	3
2. Задание. Вариант 1.....	3
3. Текст программы lab8.cpp.....	4
Выводы.....	9

## 1. Цель работы

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 inline в программу, разработанную на языке С.

## 2. Задание. Вариант 1

Name poly - generates a polynomial from arguments

Usage double poly(double x, int n, double c []);

Prototype in math.h

Description poly generates a polynomial in x, of degree n, with coefficients c[0], c[1], ..., c[n].

For example, if n=4 the generated polynomial is

$$c[4].x^4 + c[3].x^3 + c[2].x^2 + c[1].x + c[0]$$

The polynomial is calculated using Horner's method:

$$\text{polynom} = (..((x.c[n] + c[n-1]).x + c[n-2]).).x + c[0]$$

Return value poly returns the value of the polynomial as evaluated for the given x.

### 3. Текст программы main.cpp

```
#include <iostream>
#include <math.h>

float poly(float x, int n, float* c) {
    float result = 0;

    __asm {
        ; умножаем на n - 1
        mov ecx, n
        sub ecx, 1
        mult_loop:
            dec ecx
            add c, 4
            cmp ecx, 0
            jne mult_loop

        mov ecx, n
        sub ecx, 1
        mov edx, c
        fld [edx]; st(0) = c[n-1]

        add_loop:
            dec ecx

            fld x; st(0) = x, st(1) = result
            fmulp st(1), st

            sub c, 4
            mov edx, c
            fld [edx]
            faddp st(1), st

            fst result
    }
```

```

        cmp ecx, 0
        jne add_loop
    }

    return result;
}

int main()
{
    float x;
    int n;

    std::cout << "Введите x: ";
    std::cin >> x;
    std::cout << "Введите n: ";
    std::cin >> n;

    float c = new float[n];
    for (int i = 0; i < n; ++i) {
        std::cin >> c[i];
    }

    std::cout << "Ответ: " << poly(x, n, c) << std::endl;
}

```

## **Выводы**

В процессе выполнения лабораторной работы была изучена работа с вещественными числами и математическим сопроцессором.