

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.

Студент гр. 9383

Гладких А.А.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2020

Цель работы.

Применить на практике знания по работе с математическим сопроцессором. Написать программу, обрабатывающую вещественные числа.

Текст задания.

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.

ВАРИАНТ 2.

Name cosh - hyperbolic function:

Usage double cosh(double x);

Prototype in math.h

Description cosh computes the hyperbolic cosine of the input value.

$\cosh(x) = (\exp(x) + \exp(-x)) / 2$

cosh is more accurately calculated by the polynomial $(1 + x^2/2)$

when x is tiny ($|x| < 2^{-13}$).

Ход работы.

В ходе работы была разработана программа, которая вычисляет значение гиперболического косинуса.

В функции `main()` происходит считывание числа x , для которого вызывается функция `cosh()`, непосредственно считающая значение гиперболического косинуса в точке x .

Сначала функция проверяет, попадает ли заданный x в диапазон от 0 до 2 в -13 . Если да, то происходит переход к метке `little_x` и считает значение функции по формуле $(1 + x^2/2)$. Если же x не попадает в этот диапазон, то есть, он больше 2 в -13 , то значение гиперболического косинуса вычисляется по

формуле $(\exp(x) + \exp(-x)) / 2$. Сначала функция считает значение экспоненты в точке x , затем в точке $-x$. Полученные значения она складывает и делит на два.

В конце функция печатает результат функции `cosh()` на экран.

Исходный код программы представлен в приложении А.

Примеры работы программы.

Таблица 1 — Примеры работы программы.

№	Входные данные	Выходные данные
1	10	11013.232920103328
2	0.00011444091796875	1.00000000065483619
3	0	1

Выводы.

Были применены на практике знания по работе с математическим сопроцессором и была написана программа, которая обрабатывает вещественные числа, вычисляя значение гиперболического косинуса в заданной точке.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: source.cpp

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

double cosh(double x) {

    double res;
    int deg = -13;

    _asm {
        fld x; //st(0) = x
        fabs; //st(0) = |x|

        fild deg; //st(0) = deg st(1) = |x|
        fld1; //st(0) = 1 st(1) = deg st(2) = |x|
        fscale; // st(0) = st(0) * 2 ** (st(1)) st(1) = |x|

        fcomip st, st(1); // st(0) = |x|
        fstp st(0); // st is empty

        j1 little_x;

        fld x; //st(0) = x
        fldl2e; //st(0) = log2e st(1) = x
        fmul; //st(0) = x * log2e
        fld st; //st(0) = x * log2e st(1) = x * log2e
        frndint; //st(0) = [x * log2e] st(1) = x * log2e
        fsub st(1), st; //st(0) = [x * log2e] st(1) = x * log2e -
[x * log2e]
        fxch st(1); //st(0) = x * log2e - [x * log2e] st(1) = [x *
log2e]
        f2xm1; //st(0) = 2 ** (x * log2e - [x * log2e]) - 1 st(1) =
[x * log2e]
```

```

        fld1; //st(0) = 1 st(1) = 2 ** (x * log2e - [x * log2e]) -
1 st(2) = [x * log2e]
        fadd; //st(0) = 2 ** (x * log2e - [x * log2e]) st(1) = [x *
log2e]

        fscale; //st(0) = e ** x st(1) = [x * log2e]
        fstp st(1); //st(0) = e ** x


        fld x; //same, but with -x
        fchs;
        fldl2e;
        fmul;
        fld st;
        frndint;
        fsub st(1), st;
        fxch st(1);
        f2xm1;
        fld1;
        fadd;
        fscale;
        fstp st(1);


        fadd; //e ** x + e ** (-x)
        fld1;
        fld1;
        fadd;
        fdiv; //(e ** x + e ** (-x)) / 2


        jmp fin;

little_x:

        fld1;
        fld x;
        fld x;
        fmul;
        fld1;
        fld1;
        fadd;
        fdiv;

```

```

        fadd;
        jmp fin;

    fin:
        fstp res;
    }

    return res;
}

int main()
{
    setlocale(LC_ALL, "Rus");
    double x = 0.0;

    std::cout.precision(17);
    std::cout << "Введите число : ";
    std::cin >> x;
    std::cout << "Ответ : " << cosh(x) << "\n";

    //      std::cout << "Тест для очень маленького числа :  $x = 2^{(-13)} - 2^{(-17)}$ \n";
    //      x = pow(2, -13) - pow(2, -17);
    //      std::cout << x << '\n';
    //std::cout << "Ответ : " << cosh(x) << "\n";
}

```