МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

**(ФГБОУ ВО «КубГУ»)**

**Факультет компьютерных технологий и прикладной математики**

**Кафедра информационных технологий**

**ОТЧЕТ ПО ИНДИВИДУАЛЬНОМУ ЗАДАНИЮ № 3**

**по дисциплине  
 «Параллельное и низкоуровневое программирование»**

Выполнил студент группы 25/2                                       А.А. Козин

Направление подготовки  02.03.03  Математическое обеспечение и администрирование информационных систем

Курс    2

Отчет принял доктор физико-математических наук, профессор                                                                                       А.И. Миков

Краснодар

2022 г.

**Задание 5. Арифметика с плавающей запятой.**Написать программу вычисления значения F(x) функции стандартного нормального распределения случайной величины (распределения Гаусса) на основе формулы разложения в степенной ряд, умноженный на квадратичную экспоненту.

**Решение**:  
**Нормальное распределение**, также называемое распределение Гаусса – распределение вероятностей, которое в одномерном случае задается функцией плотности вероятности, совпадающей с функцией Гаусса.

Есть несколько вариаций этой формулы, я возьму связанную с ней **функцию ошибок**.

Функция ошибок — неэлементарная функция, возникающая в теории вероятностей и статистике. Она определяется как

Для реализации программы возьму асимптотическое разложение этой функции, приведенное ниже:

Так как верхняя граница сумы — это бесконечность, я возьму какое-то большое число (которое позволяет компьютер), чтобы программа смогла вычислить F(x).

В программе используются 6 функций с ассемблерными для вычислений.

При запуске программы вы вводите t и end. Значение t нужно для подстановки в формулу, а end для конца суммы (1) т.к. если выбрать слишком большой диапазон суммы, нужно будет высчитывать большой факториал.

**Текст программы:**

#include <iostream>

#include <cmath>

using namespace std;

double get\_PI() {

double pi;

\_\_asm {

fldpi;

fstp pi;

}

return pi;

}

long get\_factorial(int x) {

long factor;

\_\_asm {

mov ebx, 1

mov eax, x

call fact

jmp quit

fact :

cmp eax, 1

jle exitfact

imul ebx, eax

dec eax

call fact

exitfact :

ret

quit :

mov factor, ebx

}

return factor;

}

long get\_pow(int x, int y) {

int m = 1;

\_\_asm

{

mov eax, x

mov ebx, m

LOOP1 :

add ebx, 1

mul x

cmp ebx, y

jb LOOP1

mov x, eax

}

return x;

}

double get\_add(int x, int y) {

\_\_asm {

mov eax, x

mov ebx, y

add eax, ebx

mov x, eax

}

return x;

}

double get\_mul(int x, int y) {

\_\_asm {

mov eax, x

mov ebx, y

mul ebx

mov x, eax

}

return x;

}

double get\_div(int x, int y) {

\_\_asm {

mov eax, x

mov ecx, y

sub edx, edx

div ecx

mov x, eax

}

return x;

}

int main() {

double e = 2.7183;

double t; cout << "Enter t = "; cin >> t;

int end; cout << "Enter end = "; cin >> end;

double numerator = 0, denominator = 0, sum = 0;

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

numerator = get\_mul(get\_pow(-1, n), get\_factorial(2 \* n));

denominator = get\_mul(get\_factorial(n), get\_pow(2 \* t, 2 \* n));

sum = get\_add(sum, get\_div(numerator, denominator));

}

double PI = get\_PI();

double first = get\_pow(e, get\_div(get\_pow(-t, 2), (t \* sqrt(PI))));

double x = get\_add(first, sum);

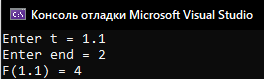
cout << "F(" << t << ") = " << x;

}

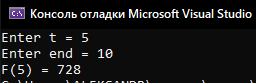
**Программа выдает в консоли результаты в следующем виде:**

Я не успел написать функцию ассемблерной вставки для корня, потому что сегодня уже дедлайн. Еще я не особо уверен в том, что я правильно использую ассемблерные вставки для вычислений т.к. очень большая погрешность выходит, и я не знаю, что с этим делать.

Test 1:



Test 2:



Test 3:

