Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

**«Пермский национальный исследовательский политехнический университет»**

Электротехнический факультет

Кафедра **«Информационные технологии и автоматизированные системы»**

направление подготовки: 09.03.04 - «Программная инженерия»

Лабораторная работа №1

По теме **«Знакомство с С++. Выполнение программы простой структуры». Вариант № 21**

Выполнял:

студент группы РИС-24-1б

Морозова Н.С.

Проверял:

Доцент кафедры ИТАС

Полякова О.А.

**Постановка задачи:**

1. Вычислить значение выражения при различных вещественных типах данных (float и double). Вычисления следует выполнять с использованием промежуточных переменных. Сравнить и объяснить полученные результаты.

Выражение , при а=100, b=0.001.

1. Вычислить значения выражений. Объяснить полученные результаты.
2. n++-m
3. m-- >n
4. n-- >m

**Решение задачи №1.**

Код

#include <iostream>

#include <clocale>

#include <cmath>

using namespace std;

int main()

{

float a = 100, b = 0.001;

float c, d, e, f, g, h;

setlocale(0, "rus");

c = pow(a - b, 4);

d = 4 \* a \* pow(b, 3);

e = 4 \* pow(a, 3) \* b;

f = 6 \* pow(a, 2) \* pow(b, 2);

g = c - (pow(a, 4) - e + f);

h = pow(b, 4) - d;

float res = g / h;

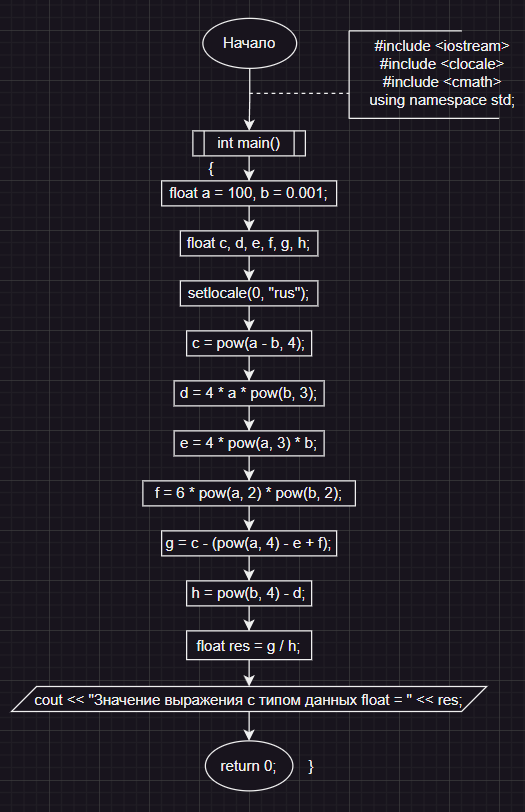
cout << "Значение выражения с типом данных float = " << res;

return 0;

}

(То же самое с типом данных double)

Блок-схема (то же самое с типом данных double)



Результаты работы программы для данных типа float:



Результаты работы программы для данных типа double:



Объяснение: тип данных float занимает меньше памяти и предоставляет менее точную десятичную точность (2 байта - 16 бит) по сравнению с типом данных double, он обычно занимает 4 байта (32 бита) и обеспечивает приблизительно 6-7 десятичных знаков точности, поэтому результат становится более точным.

**Решение задачи №2.**

Код

#include <iostream>

#include <clocale>

using namespace std;

int main()

{

int n, m;

setlocale(0, "rus");

cout << "Введите значения m и n: ";

cin >> m >> n;

cout << "n++ - m = " << n++ - m << endl;

bool a = m-- > n;

cout << "m-- > n = " << a << endl;

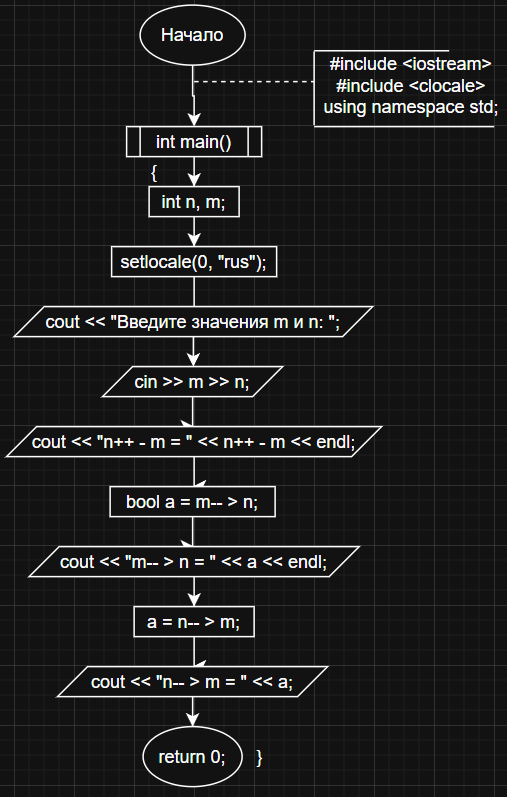
a = n-- > m;

cout << "n-- > m = " << a;

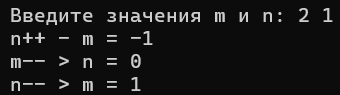
return 0;

}

Блок-схема



Результат работы программы:



Объяснение: в первом действии по приоритету операции сначала произойдёт вычитание (1 - 2), а только потом операция инкремента. А во втором действии так же по приоритету операций сначала выполнится логическое сравнение, а потом уже операция декремента (2 > 2). Третье действие будет сравнивать 2 и 1, а потом выполнится операция декремента.