Министерство образования и науки Российской Федерации  
Федеральное‌ ‌государственное‌ ‌бюджетное‌ ‌образовательное‌ ‌учреждение‌

высшего‌ ‌образования‌

**«Пермский национальный исследовательский  
политехнический университет»**

Кафедра «Информационные технологии и автоматизированные системы»

**О Т Ч Ё Т**

**по лабораторной работе №1**

Дисциплина: основы алгоритмизации и программирования

Тема: Знакомство с Си++. Выполнение программы простой структуры

Выполнил работу

студент группы МИР-21-2б

Поздняков Е. С.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Проверила

Доцент кафедры ИТАС

к.т.н. Полякова О. А.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Пермь 2022

**Цель работы**

Знакомство со средой программирования, создание, отладка и выполнение простой программы, содержащей ввод/вывод информации и простейшие вычисления.

**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.
2. Вычислить значения выражений. Объяснить полученные результаты.

**Анализ задачи**

Для удобства выполнения задания разобьем его на 3 программы, в программе 1 выполним вычисления с типом float, в программе 11 – double, а в программе 12 соответственно вычислим значения выражений (из пункта 2)

Для возведения числа в степень используем функцию pow из библиотеки cmatch

Порядок действий для вычисления выражения таков:

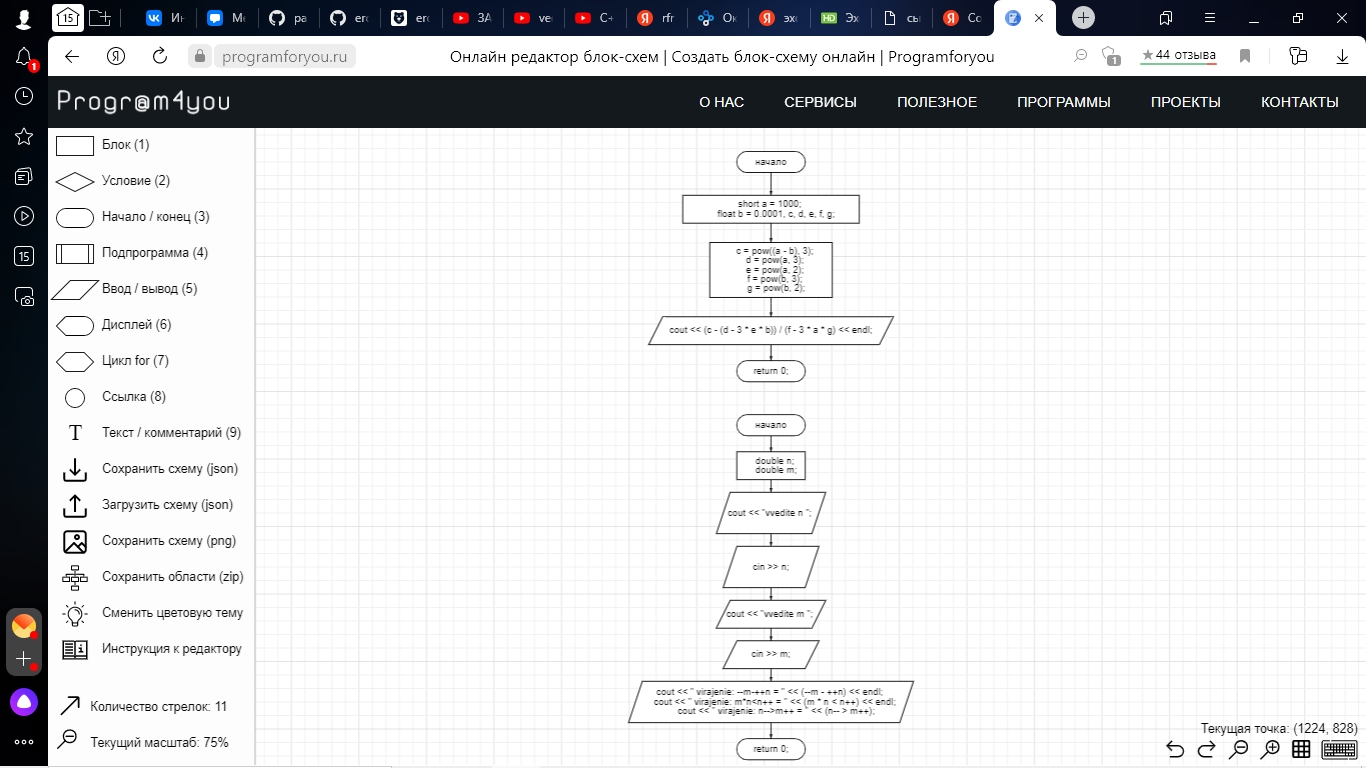
1 – Возвести каждое число в необходимую степень и присвоить значение выражения какой-либо переменной

2 – Вывести значение выражения, попутно скомпилировав его окончательное значение в строке вывода (cout << (c - (d - 3 \* e \* b)) / (f - 3 \* a \* g) << endl;)

Порядок действий для вычисления выражений (2 подзадание) таков:

1. Вывести пользователю сообщения с просьбой ввести числа
2. Получить числа с помощью функции cin
3. Вывести значения выражений, скомпилировав их конечные значения в строке вывода (cout << " virajenie: --m-++n = " << (--m - ++n) << endl; cout << " virajenie: m\*n<n++ = " << (m \* n < n++) << endl; cout << " virajenie: n-->m++ = " << (n-- > m++);)

Блок-схема



**Решения**

Программа 1:

#include <iostream>

#include <cmath>

using namespace std;

int main()

{

short a = 1000;

float b = 0.0001, c, d, e, f, g;

c = pow((a - b), 3);

d = pow(a, 3);

e = pow(a, 2);

f = pow(b, 3);

g = pow(b, 2);

cout << (c - (d - 3 \* e \* b)) / (f - 3 \* a \* g) << endl;

return 0;

}

Программа 11:

#include <iostream>

#include <cmath>

using namespace std;

int main()

{

short a = 1000;

double b = 0.0001;

double c, d, e, f, g;

c = pow((a - b), 3);

d = pow(a, 3);

e = pow(a, 2);

f = pow(b, 3);

g = pow(b, 2);

cout << (c - (d - 3 \* e \* b)) / (f - 3 \* a \* g) << endl;

return 0;

}

Программа 12:

#include <iostream>

using namespace std;

int main()

{

double n;

double m;

cout << "vvedite n ";

cin >> n;

cout << "vvedite m ";

cin >> m;

cout << " virajenie: --m-++n = " << (--m - ++n) << endl;

cout << " virajenie: m\*n<n++ = " << (m \* n < n++) << endl;

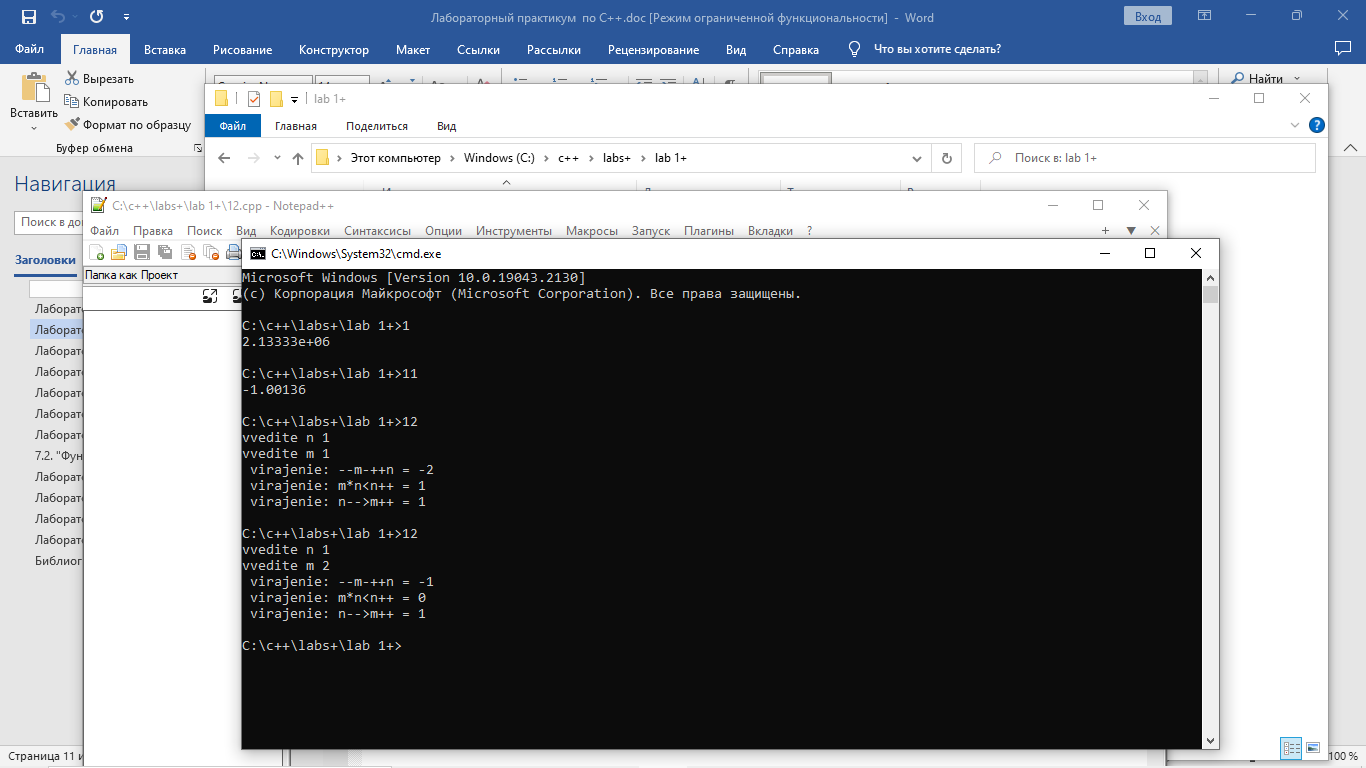
cout << " virajenie: n-->m++ = " << (n-- > m++);

cout << "\n";

return 0;

}

Скриншот и объяснение результатов



В случаях с вычислением выражения из подзадания 1 мы получаем разные результаты т.к. тип double имеет более высокую точность , что сказывается на результате вычислений

При выполнении подзадания 2 следует учитывать что префиксные операции выполняются над переменными до их использования, а постфиксные после, соответственно заданные выражения правильней интерпретировать как : 0-2=-2; 0\*2<3; 3>0 или 1-2=-1; 1\*2<3; 3>2

Тут стоит отметить, что если в выражении стоит знак больше или меньше, или подобный им, то такое выражение возвращает 0 или 1, где 0 это лож, а 1 это истина