Министерство образования и науки Российской Федерации

Федеральное государственное автономное образовательное учреждение

высшего образования

**«Пермский национальный исследовательский**

**политехнический университет»**

**(ПНИПУ)**

Электротехнический факультет

Кафедра «Информационные технологии и автоматизированные системы»

Направление подготовки 15.03.06 Мехатроника и робототехника

**ЛАБОРАТОРНАЯ РАБОТА №1**

Знакомство с Си++. Выполнение программы простой структуры

Выполнил студент гр. ПРТ-20-1б

Усманов Азат Камилович

(фамилия, имя, отчество)

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

(подпись)

**Пермь 2021**

Содержание

[1. Постановка задачи 3](#_Toc82022972)

[2. Программа решения задания 1 3](#_Toc82022973)

[3. Результаты работы программы для данных типа *float* 4](#_Toc82022974)

[4. Результаты работы программы для данных типа *double* 4](#_Toc82022975)

[5. Объяснение результатов 4](#_Toc82022976)

[6. Программа решения задания 2 4](#_Toc82022977)

[7. Результаты работы программы 4](#_Toc82022978)

[8. Объяснение результатов 5](#_Toc82022979)

# Постановка задачи

Задача1. Реализовать программу в которой выполняется следующее

# Написать программу для решения выражения ,

где а=1000 а b=0.0001 используя два типа данных (double и float)

затем данные сравнить

1. Вычислить значение выражений а) n++\*m б) n++<m с) m-- >m

и объяснить полученные результаты

# Программа решения задания 1

***Double***

#include <iostream>

#include <iomanip>

using namespace std;

int main()

{

double a = 1000.0, b = 0.0001, c = 0.0, k = 0.0, m = 0.0, n = 0.0, p = 0.0, h = 0.0, i=0.0, j=0.0;

k = pow(a - b, 3);

m = pow(a, 3);

n = pow(b, 3);

p = 3 \* a \* pow(b, 2);

h = 3 \* pow(a, 2) \* b;

i = k - m;

j = n - p - h;

c = (k - m) / (n - p - h);

cout <<setprecision(15)<<k<< endl;

cout << setprecision(15) << m << endl;

cout << setprecision(25) << n << endl;

cout << setprecision(25) << p << endl;

cout << setprecision(15) << h << endl;

cout << setprecision(15) << j << endl;

cout << setprecision(15) << i << endl;

cout << setprecision(15) << c << endl;

cout << fixed;

return 0;

***Float***

#include <iostream>

#include <iomanip>

using namespace std;

int main()

{

float a = 1000.0f, b = 0.0001f, c = 0.0f, k = 0.0f, m = 0.0f, n = 0.0f, p = 0.0f, h = 0.0f, i = 0.0f, j = 0.0f;

k = pow(a - b, 3);

m = pow(a, 3);

n = pow(b, 3);

p = 3 \* a \* pow(b, 2);

h = 3 \* pow(a, 2) \* b;

i = k - m;

j = n - p - h;

c = (k - m) / (n - p - h);

cout << setprecision(30) << k << endl;

cout << setprecision(30) << m << endl;

cout << setprecision(25) << n << endl;

cout << setprecision(25) << p << endl;

cout << setprecision(15) << h << endl;

cout << setprecision(15) << j << endl;

cout << setprecision(15) << i << endl;

cout << setprecision(15) << c << endl;

cout << fixed;

return 0;

# Результаты работы программы для данных типа *float*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
| 999999700.00003 | 1000000000 | 1.0000000000000001818e-12 | 3.000000000000000076e-05 | 300 | 0.99999979986422 |

Таблица 4.1 – Результаты работы программы для данных типа float

# результаты программы для данных типа *double*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
| 999999616 | 1000000000 | 9.9999988758398e-13 | 2.99999992421363e-05 | 300 | 1.27999985218048 |

Таблица 5.1 – Результаты работы программы для данных типа double

# Объяснение результатов

Разница результатов вычисления появилась ввиду различной точности типов данных. В данном случае тип double – вещественный двойной точности (8 байт), когда float – вещественный одинарной точности (4 байт).

при использовании типа float, значение уменьшаемого и вычитаемого ( из-за малого значения числа потеряло точность малых разрядов. В итоге это привело к результату, отличному то того что получено от того, что был получен при типе double.Этапы вычисления можно увидеть в таблице 4.1 и таблице 5.1.

# Программа решения задания 2

using namespace std;

void task1()

{

int n, m, k, t;

cout << "n = "; cin >> k;

cout << "m = "; cin >> t;

n = k;

m = t;

cout << "n++\*m = " << n++ \* m << " n = " << n << " m = " << m << endl;

n = k;

m = t;

cout << "n++<m = " << (n++ < m ? "true" : "false") << " n = " << n << " m = " << m << endl;

n = k;

m = t;

cout << "m-->m = " << (m-- > m ? "true" : "false") << " n = " << n << " m = " << m << endl;

}

int main()

{

task1();

return 0;

}

# Результаты работы программы

Сначала программа попросила ввести два числа я ввёл n=5 m=6

n++\*m = 25 n = 6 m = 6

n++<m = true n = 6 m = 6

m-->m = false n = 5 m = 5

# Объяснение результатов

Операции ++ и -- имеют префиксную форму написания и постфиксную. В префиксной форме сначала изменяется операнд, а затем его значение становится результирующим значением выражения, а в постфиксной форме значением выражения является исходное значение операнда, после чего он изменяется. В первом примере над n cделали действие (m\*n) а уже потом прибавили к ней единицу. Во втором примере была проверка неравенства м и n, а потом уже была прибавлена к n единица. В третьем примере всегда будет false поскольку идет сначала проверка неравенства, а потом вычитание единицы, то m при проверке неравенства будут равны, следовательно, неравенство всегда будет неверным.