

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ПЕРМСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

ЭЛЕКТРОТЕХНИЧЕСКИЙ ФАКУЛЬТЕТ

КАФЕДРА «ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И
АВТОМАТИЗИРОВАННЫХ СИСТЕМ»

**ОТЧЁТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ №1**

Дисциплина: «Основы алгоритмизации и программирования»

Выполнил:
Студент группы ИВТ-21-26
Безух Владимир Сергеевич

Проверил:
Доцент кафедры ИТАС
Полякова Ольга Андреевна

Пермь,
Ноябрь 2021

Вариант №2

Постановка задач

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.

Задание №1	Задание №2
$a=1000, b=0.0001, \frac{(a-b)^2 - (a^2 - 2ab)}{b^2};$	<ol style="list-style-type: none">1. <code>++n * ++m;</code>2. <code>m++ < n;</code>3. <code>n++ > m;</code>

Анализ задач

1. Вся суть задачи заключается в сравнении результатов вычислений при использовании вещественных типов данных разной точности (float — одинарной, double — двойной).
2. Нужно объяснить, как выполняются выражения.

Описание переменных

1.

const float A_F = 1000, B_F = 0.0001; — параметры функции типа float.
function(A_F, B_F);

const double A_D = 1000, B_D = 0.0001; — параметры функции типа double.
function(A_D, B_D);

T t_1, t_2, t_3, t_4, result; — переменные для хранения результатов вычислений.

t_1 = (a - b) * (a - b);
t_2 = a * a - 2 * a * b;
t_3 = t_1 - t_2;
t_4 = b * b;
result = t_3 / t_4;

2.

int n = 1, m = 1; — произвольные значения для анализа.

Анализ результатов

1.

В обоих случаях установлен модификатор вывода `setprecision(12)`.

Результаты работы программы для данных типа float

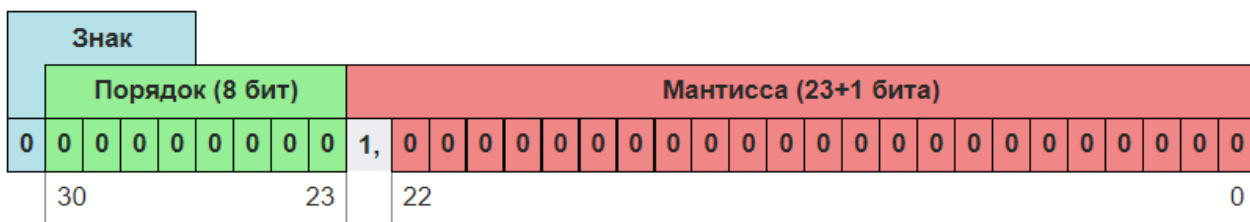
$(a-b)^2$	(a^2-2ab)	$(a-b)^2-(a^2-2ab)$	b^2	$\frac{(a-b)^2-(a^2-2ab)}{b^2}$
999999.75	999999.8125	-0.0625	9.99999905 105e-09	-6250000.5

Результаты работы программы для данных типа double

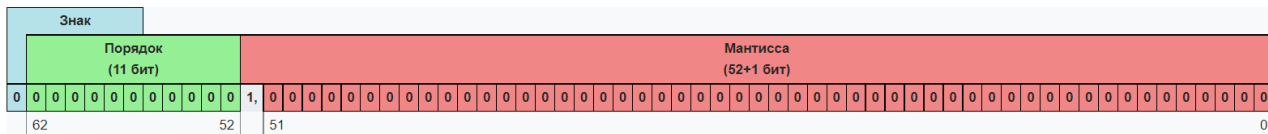
$(a-b)^2$	(a^2-2ab)	$(a-b)^2-(a^2-2ab)$	b^2	$\frac{(a-b)^2-(a^2-2ab)}{b^2}$
999999.8	999999.8	1.00117176771e-08	1e-08	1.00117176771

Разница в результатах вычислений объясняется различной точностью типов данных.

Машинное слово для float (4 байта):



Машинное слово для double (8 байт):



Можно заметить, как начиная уже с первых вычислений, появляется ошибка округления. Результаты вычислений $(a-b)^2$ и (a^2-2ab) представляют собой числа с большим количеством знаков после запятой. Из-за недостатка разрядов, значения `float` округляются менее удачно, нежели в случае с `double`. В конечном итоге, лавинообразно накапливаются ошибки округления, итоговые результаты заметно отличаются.

Если нет какой-то причины использовать непосредственно `float`, то по умолчанию имеет смысл применять именно `double`. Стоит помнить, что гарантированная точность для `float` составляет примерно 6-9 знаков после запятой (обычно 7) в зависимости от особенностей среды.

```
template <typename T>
void function(const T &a, const T &b)
{
    T t_1, t_2, t_3, t_4, result;

    t_1 = (a - b) * (a - b); cout << t_1 << "\n";
    t_2 = a * a - 2 * a * b; cout << t_2 << "\n";
    t_3 = t_1 - t_2; cout << t_3 << "\n";
    t_4 = b * b; cout << t_4 << "\n";
    result = t_3 / t_4; cout << result << "\n\n";
}
```

2.

Для удобного анализа выражений возьмём n и m равными 1.

Результат выражения определён, т.к. между двумя точками следования оба скалярных объекта изменяются не более одного раза. Рассмотрим исполнение кода детально ($n = m = 1$).

а) $++n * ++m$

Унарный оператор префиксный инкремент увеличивает значение переменной n на единицу ($n = 2$), затем возвращает значение переменной n в левую часть бинарного оператора умножения.

Унарный оператор префиксный инкремент увеличивает значение переменной m на единицу ($m = 2$), затем возвращает значение переменной m в правую часть бинарного оператора умножения.

Бинарный оператор умножения принимает левый и правый операнды ($2 * 2$) и возвращает результат вычислений — литерал типа `int` равный 4.

б) $m++ < n$

Унарный оператор постфиксный инкремент возвращает значение переменной m в левую часть бинарного оператора меньше, затем увеличивает значение переменной m на единицу ($m = 2$).

Переменная n возвращает своё значение в правую часть бинарного оператора меньше.

Бинарный оператор меньше принимает левый и правый операнды ($1 < 1$) и возвращает результат логического выражения — `false` или 0.

в) $n++ > m$

Унарный оператор постфиксный инкремент возвращает значение переменной n в левую часть бинарного оператора больше, затем увеличивает значение переменной n на единицу ($n = 2$).

Переменная m возвращает своё значение в правую часть бинарного оператора больше.

Бинарный оператор больше принимает левый и правый операнды ($1 > 1$) и возвращает результат логического выражения — `false` или 0.

```
void firstTask()
{
    int n = 1, m = 1;

    cout << (++n * ++m) << "\n";
    cout << (m++ < n) << "\n";
    cout << (n++ > m) << "\n\n";
}
```