Пример. Нахождение дробной и целой частей числа

```
#include <iostream>
#include <cmath>
using namespace std;
int main()
{
	double x, y, n;
	x = -41.56789;
	// x - дробная часть числа
	y = modf(x, &n);
	printf("modf(%f)=fractional %f and integer : %f",x, y, n);

	return 0;
}
```

Задания. Шаблоны

1. Для работы с одномерными арифметическими массивами разработать шаблоны
    a. ввода массива с клавиатуры
    b. заполнения случайными числами
    c. вывода массива
    d. суммы элементов массива
    e. произведения элементов массива
    f. среднего арифметического элементов массива
    g. нахождение минимального (максимального) элемента и его индекса
    h. инвертирования массива

2. Для одной из задач из 1 написать функции перегрузки.

Задания. Лямбда-функции

3. Напишите лямбда функции:
    a. выводит на экран сообщение.
    b. возвращает истину, если аргумент четное, положительное число.
    c. возвращает истину, если аргумент целое число в диапазоне от 65 до 122.
    d. возвращает символ по его коду
    e. возвращает истину, если аргумент имеет нулевую дробную часть.
    f. возвращает сумму квадратов двух чисел

Задания. vector

4. Создать случайное число n.
    a. Заполнить vector из n элементов случайными числами.
    b. Вставить в начало вектора 7 элементов, равных 10.
    c. Удалить второй и предпоследний элементы.
    d. Удалить 5 последних элементов.
5. Создать два одинаковых вектора. Сравнить их (v1==v2). Очистить полностью первый и снова сравнить.

http://www.cplusplus.com/reference/algorithm/ - справочная информация

Задания. algorithm

6. Создайте вектор, состоящий из 1 и 2. Все 2 замените на 5.
7. Создайте вектор, состоящий из чисел 1, -2, 3, -4, 5, -6,…-10. Замените все отрицательные элементы на их модуль (**replace_if**).
8. Замените все четные элементы на их квадраты.
9. Замените все нечетные элементы на 0.
10. Удалите все нулевые элементы (**remove_if**).
11. Создайте вектор, в котором есть повторяющиеся элементы. Оставьте в векторе только уникальные элементы (**unique**).
12. Создать вектор с отрицательными и положительными элементами. Скопировать в новый вектор только отрицательные элементы (**copy_if**).
13. Создать вектор из целых чисел. Выяснить все ли элементы в нем четные (**all_of**). (http://www.cplusplus.com/reference/algorithm/all_of/)
14. Создать вектор из целых чисел. Выяснить есть ли в нем хотя бы один четный элемент. (**any_of**). (http://www.cplusplus.com/reference/algorithm/any_of/)
15. С помощью функции **for_each** (http://www.cplusplus.com/reference/algorithm/) вывести на экран элементы вектора.
16. С помощью функции **none_of** удостоверьтесь, что в векторе нет ни одного четного элемента.
17. Определите количество элементов вектора, кратных 5 (**count_of**).
18. Вектор, заполненный случайными числами, отсортируйте по возрастанию (**sort**). (http://www.cplusplus.com/reference/algorithm/sort/
19. Найдите минимум и максимум и их индексы в векторе (**minmax_element**). http://www.cplusplus.com/reference/algorithm/minmax_element/
20. Переупорядочите элементы вектора таким образом, чтобы сначала были все отрицательные, а потом все положительные элементы (**partition**). http://www.cplusplus.com/reference/algorithm/partition/