// При ответе на вопросы 3, 4, 5 и 6 из прошлого ребуса вы использовали собственные **функторы** (т.е. классы с перегруженным оператором вызова функции), а теперь при ответе на вопросы 1 и 2 необходимо выполнить аналогичные задания, но используя лямбда функции.

Напоминаю, что нужно было для старых вопросов:

```
// Точка в трёхмерном пространстве
struct Point3D{ double x, y, z; };
```

```
// Генератор рандомных наборов точек размера num
std::vector<Point3D> generate_points(size_t num)
```

Вопрос 1

```
std::vector<Point3D> points = generate_points(20);
// Рассортировать точки в порядке возрастания до центра системы координат, после чего распечатать их в поток вывода.
```

Вопрос 2

```
std::vector<Point3D> points = generate_points(20);
// Посчитать количество точек, все координаты которых отрицательные, после чего заменить их на точки, у которых все координаты нулевые.
```

Вопрос 3

```
std::vector<int> vec1 = { 1, 3, 2, 4, 1, 3, 2 };
std::vector<char> vec2 = { 'a', '3', 'd', '5', 'd'};
std::vector<double> vec3 = { 2.71, 3.13, 3.13, 0.0 };
size_t comparisons_num = 0;
std::sort(vec1.begin(), vec1.end());
std::sort(vec2.begin(), vec2.end());
std::sort(vec3.begin(), vec3.end());
std::cout << "Comparisons num: " << comparisons_num;
// Как используя лямбда функции посчитать количество сравнений, которое понадобилось сделать во всех трёх сортировках?
```

// Используется в вопросах 4, 5 и 6:

```
struct MyClass {
    MyClass(int, int) { std::cout << "Constructed!\n"; }
    ~MyClass() { std::cout << "Destroyed!\n"; }
};
```

Вопрос 4

```
// Хочу выделить память под объект MyClass (используя древний «Си указатель»)
MyClass* ptr = new MyClass;// Почему не компилируется? Как исправить, не меняя MyClass?
delete ptr; // Не забываем подчистить память.
```

Вопрос 5

```
// Как код из 4-го вопроса оформить в современном стиле C++?(не используя new и delete)
// Примечание: используйте не ptr «Си указатель», а u_ptr «умный указатель» unique_ptr
```

Вопрос 6

```
void foo(std::unique_ptr<MyClass> u_ptr) {
    std::cout << "Hello from foo!\n";
}
```

```
foo(u_ptr); // почему не компилируется?
// Предложите три способа решения проблемы:
  1) Изменить способ передачи умн. указ. в аргумент функции (почему это сработает?)
  2) Изменить типа аргумента в интерфейсе функции (почему это сработает?)
  3) Использовать другой вид умного указателя (какой? почему это сработает?)
```