

## Решения принимаются до 14 декабря

**Задача 1** (2 балла). Отрецензируйте следующий код на C++:

```
#include <iostream>
#include <vector>
#include <list>
#include <cstdlib>
using namespace std;
template <typename T>
class Array {
private:
    T * data;
    int size;
public:
    Array(int n, T * a): size(n), data(0) {
        data = new T[n];
        for (int i = 0; i < n; i++)
            data[i] = a[i];
    }
    Array(vector<T> v) {
        size = v.size();
        for (int i = 0; i < size; ++i)
            data[i] = v[i];
    }
    Array(list<T> l) {
        size = l.size();
        int i = 0;
        for (list<T>::iterator it = l.begin(); it != l.end(); it++)
            data[i++] = *it;
    }
    Array(const Array& other) {
        delete data;
        size = other.size();
        for (int i = 0; i < size; i++)
            data[i] = other[i];
    }
    T& operator [] (int i) {
        if (i > size)
            exit(0);
        return data[i];
    }
    Array& operator += (Array& b) const {
        Array temp(*this);
        for (int i = 0; i < size(); i++)
            temp[i] += b[i];
        return temp;
    }
    int length() {
        return size();
    }
    virtual ~Array() {
        delete data;
    }
};
```

```
template <typename T>
ostream& operator << (ostream& out, Array<T> a) {
    for (int i = 0; i < a.length(); ++i)
        out << i << " ";
    return out;
}
```

Код следует загрузить в систему Code Review и написать к нему комментарии. Что бы вы посоветовали автору этого кода? :)

**Задача 2** (2 балла). Напишите шаблонный класс и/или вспомогательные шаблонные функции, позволяющие «развернуть» формулу для вычисления скалярного произведения двух числовых последовательностей заранее известной длины на этапе компиляции без использования инструкций цикла. Параметром шаблона должна быть длина последовательности. Например, если **a** и **b** — два массива фиксированной длины **N**, то вызов **scalar\_product<N>(a,b)** должен фактически превратиться в **a[0]\*b[0] + a[1]\*b[1] + ... + a[N-1]\*b[N-1]** на стадии компиляции. Сравните эффективность такой «развертки цикла» по сравнению с обычным вычислением скалярного произведения.

**Задача 3** (6 баллов). Определите набор шаблонных классов и две специальные переменные **\_1** и **\_2**, чтобы можно было «на лету» создавать несложные унарные или бинарные функторы, записывая выражения от **\_1** и **\_2**. Лямбда-функции, появившиеся в новом стандарте C++11, а также библиотеку **boost** использовать не разрешается. Программа должна позволять делать примерно следующее:

```
int a[] = {3, 14, 15, 92, 6};

// сортируем по возрастанию
std::sort(a, a+5, _1 > _2);

// выводим на экран через пробел
std::for_each(a, a+5, std::cout << _1 << " ");

// присваиваем 100 всем элементам массива
std::for_each(a, a+5, _1 = 100);

// выводим результаты, большие на 5
std::transform(a, a+5, std::ostream_iterator<int>(std::cout, " "), _1 + 5);
```