

# Тестовое задание на вакансию «Разработчик / Младший разработчик C++»

Выполнила: Лапина Анастасия

ссылка на репозиторий GitHub: <https://github.com/agirk/For-TrueConf.git>

## Задание 1

В задании 1 требуется заполнить контейнеры vector и map данными из файла. Ограничения на размеры контейнеров и диапазон чисел учтены следующим образом:

- При количестве входных данных (размера контейнера) меньше минимального выводится сообщение об ошибке и выход из функции с возвращаемым значением -1
- При количестве входных данных (размера контейнера) больше максимального "лишние" данные не включаются в контейнер (отбрасываются)
- Если число не входит в допустимый диапазон, то оно не добавляется в контейнер и не учитывается при контроле размера контейнера

```
// Вывод данных для задания 1
void showData(vector<int>& a, map<string, int>& b) {
    for (auto vec_it = a.begin(); vec_it != a.end(); vec_it++)
        cout << *vec_it << " ";

    cout << endl;

    for (auto map_it = b.begin(); map_it != b.end(); map_it++)
        cout << map_it->first << " " << map_it->second << " ";

    cout << endl << endl;
}

// Задание 1
int solveTask1(string file_name) {
    const int MIN_COUNT = 5, MAX_COUNT = 1000, MIN_NUM = 0, MAX_NUM = 15;
    vector<int> my_vec; map<string, int> my_map;
    int nums_count, int_buff;
    string str_buff;

    // Чтение данных из файла
    ifstream in(file_name);
    if (in.is_open()) {
        in >> nums_count;
        if (nums_count < MIN_COUNT) {
            cout << "Incorrect data (change the input file)" << endl << endl;
            in.close();
            return -1;
        }
        for (int i = 0; i < nums_count; i++) {
            in >> int_buff;
            if (int_buff < MIN_NUM || int_buff > MAX_NUM) { // Числа, не соответствующие условию, не включаются в
вектор
                i--;
                nums_count--;
            }
        }
    }
}
```

```

        continue;
    }
    my_vec.push_back(int_buff);
    if (my_vec.size() == MAX_COUNT) { // Если в векторе уже максимум элементов, то остальные не берем
        in >> str_buff;
        if (str_buff != "\0") { // Если считывание завершилось не в конце строки, а раньше, нужно совершить переход
на строку ниже
            getline(in, str_buff);
            break;
        }
    }
}

in >> nums_count;
if (nums_count < MIN_COUNT) {
    cout << "Incorrect data (change the input file)" << endl << endl;
    in.close();
    return -1;
}
for (int i = 0; i < nums_count; i++) {
    in >> str_buff; in >> int_buff;
    if (int_buff < MIN_NUM || int_buff > MAX_NUM) { // Пары, не соответствующие условию, не включаются в
map
        i--;
        nums_count--;
        continue;
    }
    my_map.insert(make_pair(str_buff, int_buff));
    if (my_map.size() > MAX_COUNT) break; // Если в map уже максимум элементов, то остальные не берем
}

in.close();

// Вывод исходных данных
cout << "Input data:" << endl;
showData(my_vec, my_map);

// Поиск в map значений, присутствующих в vector
for (auto map_it = my_map.begin(); map_it != my_map.end(); ) {
    if (find(my_vec.begin(), my_vec.end(), map_it->second) == my_vec.end()) map_it = my_map.erase(map_it);
    else map_it++;
}

// Поиск в vector значений, присутствующих в измененном map
for (auto vec_it = my_vec.begin(); vec_it != my_vec.end(); ) {
    bool found_in_map = false;
    for (auto map_it = my_map.begin(); map_it != my_map.end(); map_it++) {
        if (map_it->second == *vec_it) {
            found_in_map = true;
            break;
        }
    }
    if (!found_in_map) vec_it = my_vec.erase(vec_it);
    else vec_it++;
}

// Вывод результата
cout << "Result data:" << endl;
showData(my_vec, my_map);

}

else cout << "Cannot open file." << endl;

return 0;
}

```

```
int main() {
    cout << "Task 1: " << endl << endl;
    solveTask1("input.txt");

    return 0;
}
```

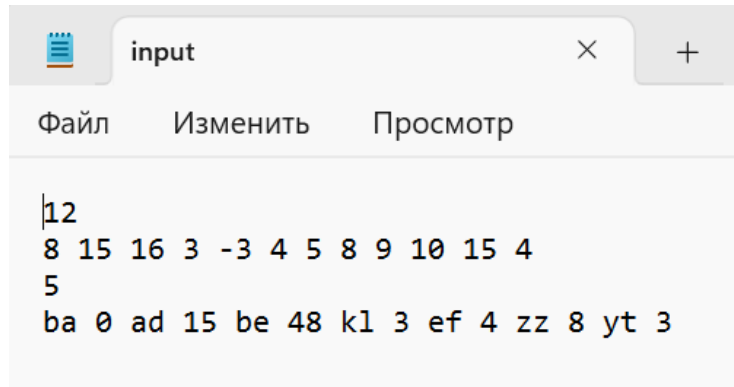


Рисунок 1. Файл input.txt со входными данными

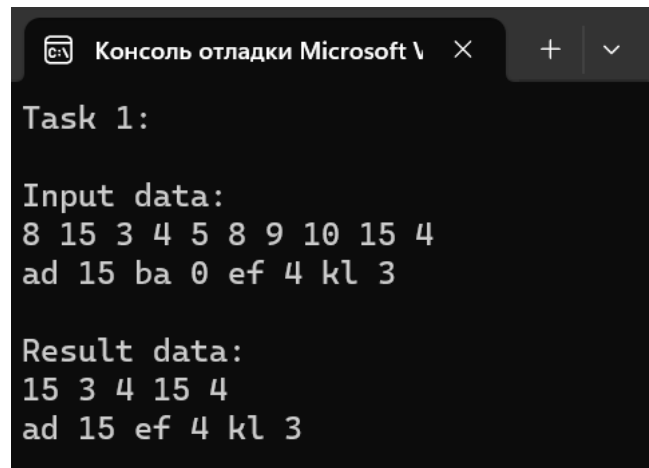


Рисунок 2. Консоль — результат выполнения программы (задание 1)

## Задание 2

```
// Задание 2
void remove_spaces(char* s) {
    char* current_position = s;
    char current_symbol;
    do {
        current_symbol = *s++;
        if (current_symbol != ' ') {
            *current_position++ = current_symbol;
        }
    } while (current_symbol != '\0');
}

int main() {
    cout << "Task 2: " << endl << endl;
    char s[] = "Cats are wonderful pets that many people love. They come in different colors and sizes. Cats are known for being playful and sometimes a bit lazy.";
    remove_spaces(s);
    cout << s << endl;
}
```

```
return 0;  
}
```

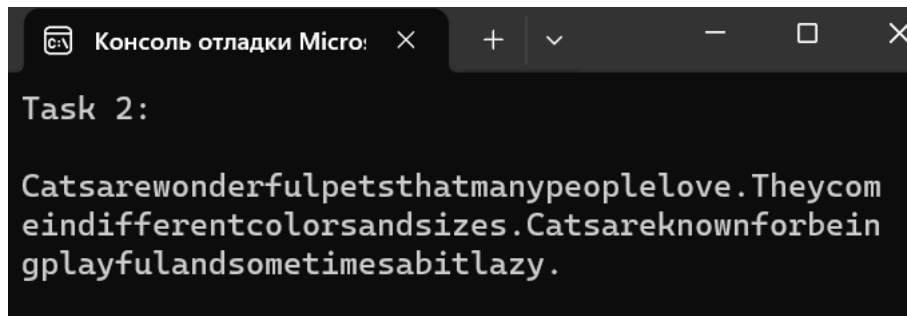


Рисунок 3. Консоль — результат выполнения программы (задание 2)