

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение  
профессионального образования

НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ  
УНИВЕРСИТЕТ им. Р.Е.АЛЕКСЕЕВА



Институт радиоэлектроники и информационных технологий  
Кафедра информатики и систем управления

## “Методы сортировки”

### Отчет по лабораторной работе №5

по дисциплине

Технология программирования

7 вариант

РУКОВОДИТЕЛЬ:

\_\_\_\_\_  
(подпись)

Капранов С.Н.  
(фамилия, и.,о.)

СТУДЕНТ:

\_\_\_\_\_  
(подпись)

Карсаков Д.Д.  
(фамилия, и.,о.)

18-ИСТ-2

(шифр группы)

Работа защищена «\_\_\_» \_\_\_\_\_

С оценкой \_\_\_\_\_

Нижний Новгород 2020

## Текст задания:

В музее регистрируется в течение дня время прихода и ухода каждого посетителя. Таким образом за день получены N пар значений, где первое значение в паре показывает время прихода посетителя и второе значения - время его ухода. Найти промежуток времени, в течение которого в музее одновременно находилось максимальное число посетителей.

## Текст программы:

```
#include "stdafx.h"
#include <fstream>
#include <iostream>
#include <string>
#include <vector>

//Функция считывания с файла времен прибытия и ухода посетителей
void read_file(std::vector<int> & mas1, std::vector<int> & mas2)
{
    std::ifstream fin;
    fin.open("ishodnik.txt");
    std::string str;
    while (!fin.eof())
    {
        char ch;
        ch = fin.get();
        if (ch != ' ' && ch != '\n' && ch != ':' && !fin.eof())
            str += ch;
        if (ch == ' ')
        {
            mas1.push_back(atoi(str.c_str()));
            str.clear();
        }
        if (ch == '\n' || fin.eof())
        {
            mas2.push_back(atoi(str.c_str()));
            str.clear();
        }
    }
}

//Функция быстрой сортировки по возрастанию
void sort(std::vector<int> & mas, int begin, int end)
{
    int left = begin, right = end;
    int piv = mas[(left + right) / 2];
    while (left <= right)
    {
        while (mas[left] < piv)
            left++;
        while (mas[right] > piv)
            right--;
        if (left <= right)
        {
            int buf = mas[left];
            mas[left] = mas[right];
            mas[right] = buf;
            left++;
            right--;
        }
    }
    if (begin < right)
        sort(mas, begin, right);
    if (end > left)
        sort(mas, left, end);
}
```

```

int main()
{
    //mas1 - вектор для хранения времен прибытия посетителей;
    //mas2 - вектор для хранения времен ухода посетителей
    std::vector<int> mas1, mas2;
    read_file(mas1, mas2);
    mas1.shrink_to_fit();
    mas2.shrink_to_fit();
    //Сортировка векторов
    sort(mas1, 0, mas1.size() - 1);
    sort(mas2, 0, mas2.size() - 1);
    //-----
    //time1 - время начала промежутка
    //time2 - время конца промежутка
    //maxVisitors - максимальное кол-во посетителей
    //bufMaxVisitors - дополнительная переменная
    int time1 = mas1[0], time2 = mas2[0], i = 1, j = 1, maxVisitors = 0,
    bufMaxVisitors = 1;
    //Поиск промежутка времени
    while (i < mas1.size())
    {
        if (mas1[i] < mas2[j])
        {
            bufMaxVisitors++;
            if (bufMaxVisitors > maxVisitors)
            {
                time1 = mas1[i];
                maxVisitors = bufMaxVisitors;
            }
            i++;
            continue;
        }
        if (mas1[i] == mas2[j])
        {
            time2 = mas2[j + 1];
            i++;
            j++;
            continue;
        }
        if (mas1[i] > mas2[j])
        {
            if (bufMaxVisitors == maxVisitors)
                time2 = mas2[j];
            bufMaxVisitors--;
            j++;
            continue;
        }
    }
    //-----
    //Вывод результата на экран
    std::string str;
    char bufStr[5];
    _itoa_s(time1, bufStr, 10);
    str = bufStr;
    auto it = str.end();
    it -= 2;
    str.insert(it, ':');
    std::cout << str << ' ';
    _itoa_s(time2, bufStr, 10);
    str = bufStr;
    it = str.end();
    it -= 2;
    str.insert(it, ':');
    std::cout << str << std::endl;
    system("pause");
    //-----
    return 0;
}

```

**Исходные данные:**

9:10 14:30  
13:10 15:50  
8:10 17:20  
13:30 13:40  
12:10 12:30  
9:15 10:10  
12:10 12:30  
13:10 15:50  
9:15 10:10  
20:40 21:30

**Результат:**

13:30 13:40