Министерство образования Российской Федерации  
НИЖНЕВАРТОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ФАКУЛЬТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И МАТЕМАТИКИ

**Контрольная работа по предмету «Языки и методы программирования»**

**Вариант 19.**

Выполнил:

студент гр. 3312 ИВТ

*Лысенко Олеся Евгеньевна*

Нижневартовск – 2024

Задание 1

Дан массив целых чисел из n элементов, заполненный случайным образом числами из заданного пользователем промежутка. Найти максимальный из четных элементов и возвести его в квадрат. Отсортировать получившийся массив и вывести его на экран и в файл.  
Исходный код (Visual Studio c++)

//1.Дан массив целых чисел из n элементов, заполненный случайным образом числами из заданного пользователем промежутка.

// Найти максимальный из четных элементов и возвести его в квадрат. Отсортировать получившийся массив и вывести его на экран и в файл.

#include <iostream>

#include <fstream>

#include <vector>

#include <algorithm>

#include <random>

#include <ctime>

using namespace std;

int main() {

setlocale(LC\_ALL, "Rus");

srand(time(nullptr));

int n;

cout << "Введите размер массива: ";

cin >> n;

int lower\_bound, upper\_bound;

cout << "Введите нижнюю и верхнюю границы промежутка через пробел: ";

cin >> lower\_bound >> upper\_bound;

int\* arr = new int [n];

for (int i = 0; i < n; ++i) {

arr[i] = rand() % (upper\_bound + 1 - lower\_bound) + lower\_bound;

cout << arr[i] << " ";

}

int max\_even = 0;

for (int i = 0; i < n; ++i) {

if (arr[i] % 2 == 0 && arr[i] > max\_even) {

max\_even = arr[i];

}

}

if (max\_even == 0) {

cout << "В массиве нет четных чисел." << endl;

delete[] arr;

return 0;

}

max\_even \*= max\_even;

for (int i = 0; i < n - 1; ++i) {

for (int j = 0; j < n - i - 1; ++j) {

if (arr[j] > arr[j + 1]) {

swap(arr[j], arr[j + 1]);

}

}

}

cout << "Отсортированный массив:" << endl;

for (int i = 0; i < n; ++i) {

cout << arr[i] << " ";

}

cout << endl;

ofstream outfile("sorted\_array.txt");

if (outfile.is\_open()) {

for (int i = 0; i < n; ++i) {

outfile << arr[i] << " ";

}

outfile.close();

cout << "Отсортированный массив записан в файл sorted\_array.txt." << endl;

}

else {

cerr << "Не удалось открыть файл для записи." << endl;

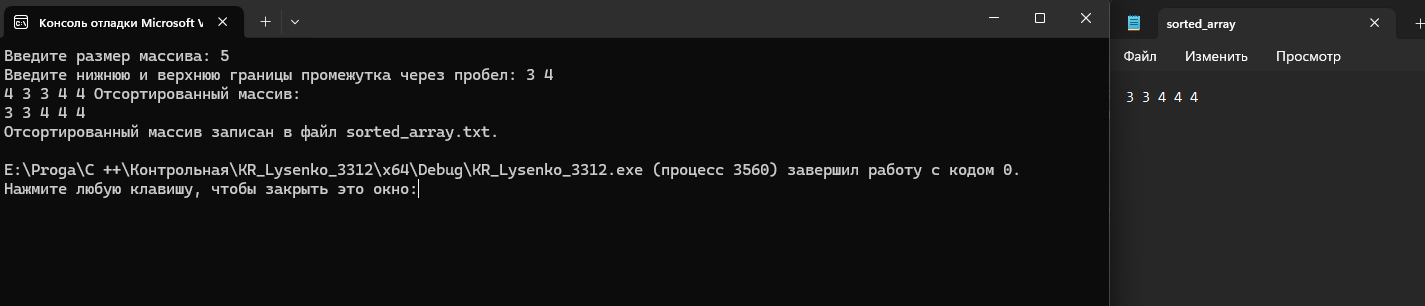
}

delete[] arr;

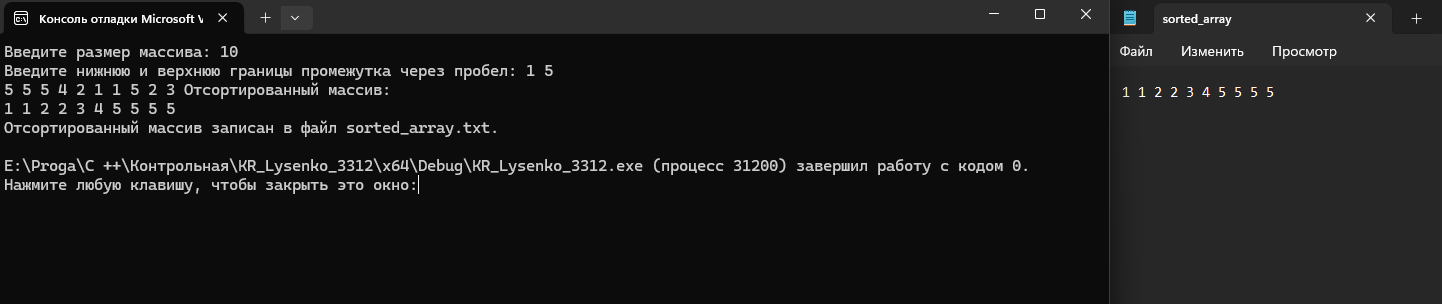
return 0;

}

Запуск программы 1



Запуск программы 2



Задание 2

Дан двумерный массив размером n\*m, заполненный случайными числами. Найти строку с максимальным произведением элементов и в ней первый четный элемент и вывести их в файл  
Исходный код (Visual Studio c++)

// 2.Дан двумерный массив размером n\*m, заполненный случайными числами.

// Найти строку с максимальным произведением элементов и в ней первый четный элемент и вывести их в файл.

#include <iostream>

#include <fstream>

#include <ctime>

using namespace std;

int main() {

setlocale(LC\_ALL, "Rus");

srand(time(nullptr));

int n, m;

cout << "Введите количество строк и столбцов (n и m): ";

cin >> n >> m;

int\*\* matrixn = new int\* [n];

int\* matrixm = new int[m];

for (int i = 0; i < n; ++i) {

matrixn[i] = new int[m];

for (int j = 0; j < m; ++j) {

matrixn[i][j] = rand() % 100 + 1;

cout << matrixn[i][j] << " ";

}

cout << endl;

}

int max\_product = 0;

int max\_row\_index = -1;

int max\_even\_element = 0;

// Поиск строки с максимальным произведением элементов и первого четного элемента

for (int i = 0; i < n; ++i) {

int product = 1;

bool found\_even = false;

for (int j = 0; j < m; ++j) {

product \*= matrixn[i][j];

if (matrixn[i][j] % 2 == 0 && !found\_even) {

max\_even\_element = matrixn[i][j];

found\_even = true;

}

}

if (product > max\_product) {

max\_product = product;

max\_row\_index = i;

}

}

ofstream outfile("result.txt");

if (outfile.is\_open()) {

if (max\_row\_index != -1) {

outfile << "Строка с максимальным произведением элементов:" << endl;

for (int j = 0; j < m; ++j) {

outfile << matrixn[max\_row\_index][j] << " ";

}

outfile << endl;

outfile << "Первый четный элемент в этой строке: " << max\_even\_element << endl;

outfile << "Произведение элементов: " << max\_product << endl;

}

else {

outfile << "Массив пустой." << endl;

}

outfile.close();

cout << "Результат записан в файл result.txt." << endl;

}

else {

cerr << "Не удалось открыть файл для записи." << endl;

}

for (int i = 0; i < n; ++i) {

delete[] matrixn[i];

}

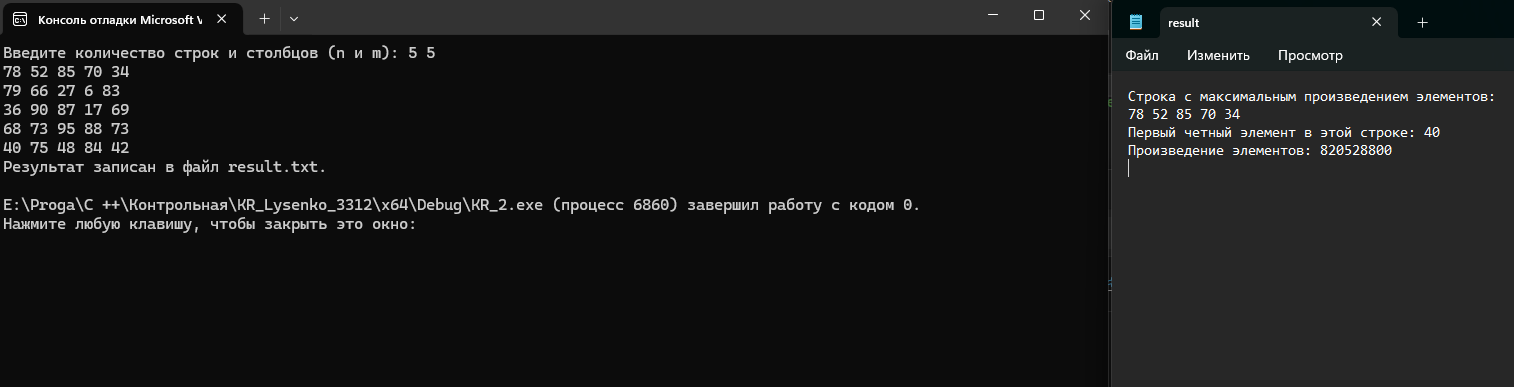
delete[] matrixn;

delete[] matrixm;

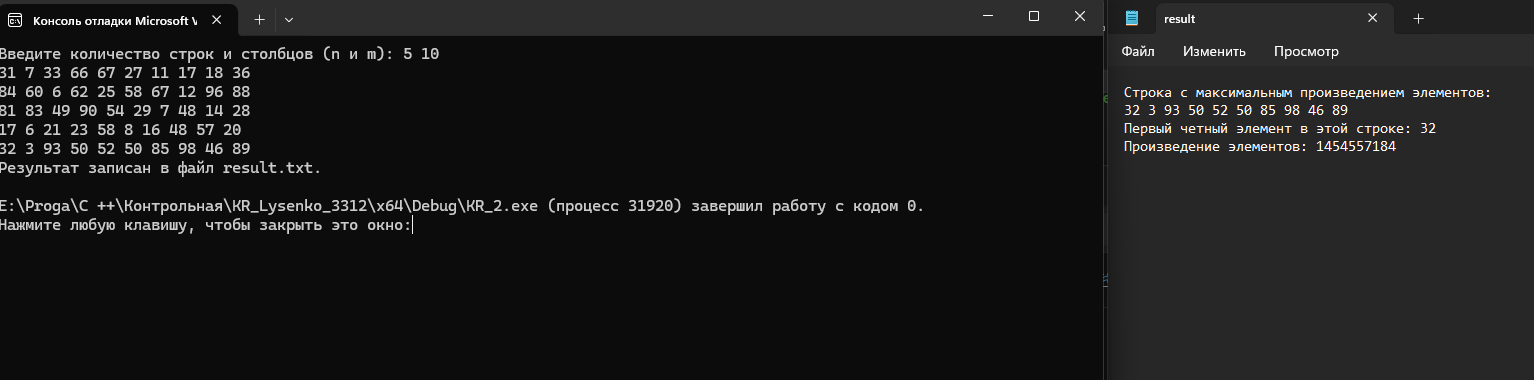
return 0;

}

Запуск программы 1



Запуск программы 2



Задание 3

Создать массив, элементами которого являются структуры – список команд «Что? Где? Когда?». В записи должны содержаться название, имя капитана, фамилия капитана, город, статус (школьная, студенческая, взрослая, корпоративная), рейтинг. Отсортировать записи по рейтингу. Запрограммировать возможность вывода всех команд из города, введенного пользователем, запись данных в файл и чтение из файла уже записанных данных.  
Исходный код (Visual Studio c++)

//.Создать массив, элементами которого являются структуры – список команд «Что? Где? Когда?».

// В записи должны содержаться название, имя капитана, фамилия капитана, город, статус (школьная, студенческая, взрослая, корпоративная), рейтинг.

// Отсортировать записи по рейтингу.

// Запрограммировать возможность вывода всех команд из города, введенного пользователем, запись данных в файл и чтение из файла уже записанных данных.

#include <iostream>

#include <fstream>

#include <algorithm>

#include <random>

#include <string>

#include <sstream>

#include <vector>

#include <ctime>

#include "Header.h"

using namespace std;

int main() {

srand(time(nullptr));

int arraySize;

cout << "Enter the number of commands: ";

cin >> arraySize;

Command\* commands = new Command[arraySize];

vector<string> names = readDataFromFile("names.txt");

vector<string> cities = readDataFromFile("cities.txt");

vector<string> statuses = readDataFromFile("statuses.txt");

vector<string> captain\_first\_names = readDataFromFile("first\_names.txt");

vector<string> captain\_last\_names = readDataFromFile("last\_names.txt");

for (int i = 0; i < arraySize; ++i) {

commands[i].name = getRandomValue(names);

commands[i].captain\_first\_name = getRandomValue(captain\_first\_names);

commands[i].captain\_last\_name = getRandomValue(captain\_last\_names);

commands[i].city = getRandomValue(cities);

commands[i].status = getRandomValue(statuses);

commands[i].rating = rand() % 10 + 1;

}

cout << "Commands before sorted:" << endl;

printCommands(commands, arraySize);

sort(commands, commands + arraySize, sortByRating);

cout << "After sorted Commands:" << endl;

printCommands(commands, arraySize);

string cityInput;

cout << "\nEnter a city to display commands from: ";

cin >> cityInput;

cout << "Commands from " << cityInput << ":" << endl;

for (int i = 0; i < arraySize; ++i) {

if (commands[i].city == cityInput) {

cout << "Name: " << commands[i].name << ", Captain: " << commands[i].captain\_first\_name << " "

<< commands[i].captain\_last\_name << ", City: " << commands[i].city << ", Status: " << commands[i].status

<< ", Rating: " << commands[i].rating << endl;

}

}

ofstream outFile("commands.txt");

if (outFile.is\_open()) {

for (int i = 0; i < arraySize; ++i) {

outFile << "[ " << i + 1 << " ]" << " commands\_name: " << commands[i].name << " | captain\_first\_name: " << commands[i].captain\_first\_name << " | captain\_last\_name: " << commands[i].captain\_last\_name << " | city: "

<< commands[i].city << " | status: " << commands[i].status << " | rating: " << commands[i].rating << endl;

}

outFile.close();

cout << "\nData written to file 'commands.txt'." << endl;

}

else {

cerr << "Unable to open file for writing." << endl;

}

delete[] commands;

return 0;

}

HEADER.H

#ifndef CITIES\_H

#define CITIES\_H

#include <string>

#include <vector>

using namespace std;

struct Command {

string name;

string captain\_first\_name;

string captain\_last\_name;

string city;

string status;

int rating;

};

vector<string> readDataFromFile(const string& filename);

string getRandomValue(const vector<string>& values);

void printCommands(const Command commands[], int size);

bool sortByRating(const Command& cmd1, const Command& cmd2);

#endif

SOURSE.CPP

#include "Header.h"

#include <fstream>

#include <iostream>

#include <algorithm>

#include <random>

#include <ctime>

vector<string> readDataFromFile(const string& filename) {

vector<string> data;

ifstream file(filename);

if (file.is\_open()) {

string value;

while (getline(file, value)) {

data.push\_back(value);

}

file.close();

}

else {

cerr << "Unable to open file " << filename << endl;

}

return data;

}

string getRandomValue(const vector<string>& values) {

if (!values.empty()) {

int index = rand() % values.size();

return values[index];

}

else {

cerr << "No values available." << endl;

return "";

}

}

void printCommands(const Command commands[], int size) {

for (int i = 0; i < size; ++i) {

cout << "Name: " << commands[i].name << ", Captain: " << commands[i].captain\_first\_name << " "

<< commands[i].captain\_last\_name << ", City: " << commands[i].city << ", Status: " << commands[i].status

<< ", Rating: " << commands[i].rating << endl;

}

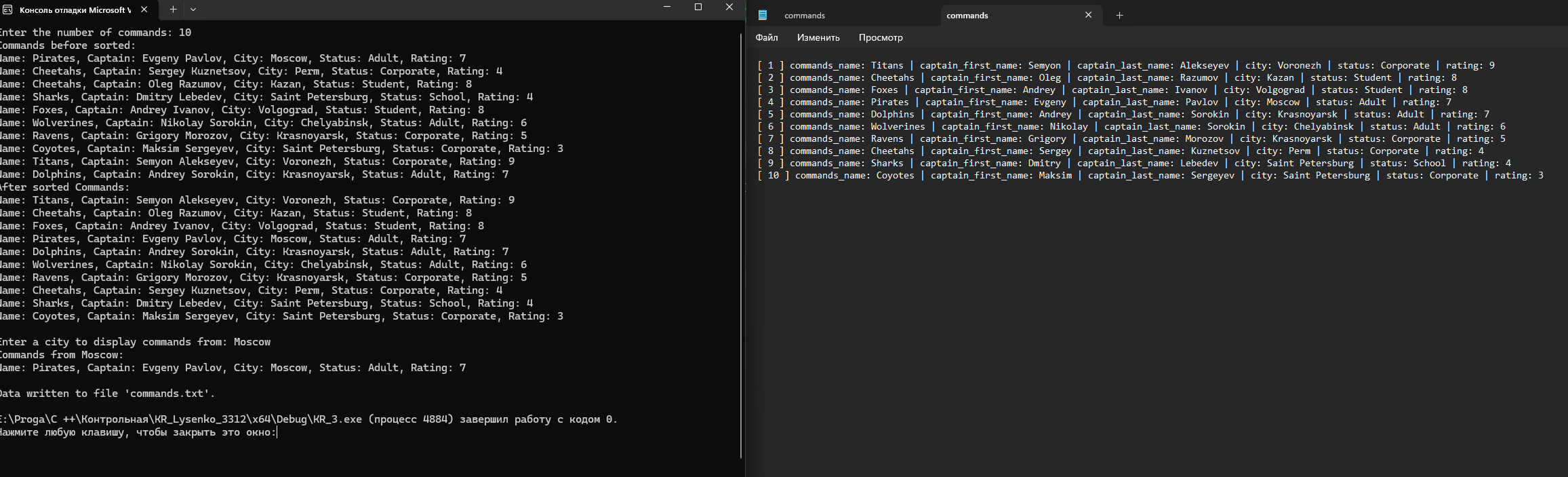
}

bool sortByRating(const Command& cmd1, const Command& cmd2) {

return cmd1.rating > cmd2.rating;

}

Запуск программы 1



Запуск программы 2

