

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

**«МИРЭА – Российский технологический университет»**

**РТУ МИРЭА**

**Отчет по выполнению практического задания номер два**

**Тема: Внешние структуры данных: текстовый и двоичный файлы**

**Дисциплина: Структуры и алгоритмы обработки данных**

Выполнил студент Михайлюк Д. С.

Группы ИНБО-07-21

Москва 2022

**СОДЕРЖАНИЕ**

[1 Задание 3](#_Toc114164723)

[1.1 Требования 3](#_Toc114164724)

[1.2 Текстовый пример 4](#_Toc114164725)

[1.3 Реализация приложения 4](#_Toc114164726)

[*1.3.1* Функционал приложения 4](#_Toc114164727)

[*1.3.2* Код основной программы 8](#_Toc114164728)

[1.4 Результаты тестирования 9](#_Toc114164729)

# Задание

Разработать программу, управления текстовым файлом.

## Требования

1. Реализация ввода-вывода на основе файловых потоков С++: ofstream, ifstrim.
2. Имя физического файла вводится пользователем и передается в функции обработки через параметр.
3. При открытии файла выполнять контроль его существования и открытия.

Примечание. При отладке программы можете имя физического файла определить через константу.

1. Разработать функции для выполнения операций над текстовым файлом.
2. создание текстового файла средствами текстового редактора кодировки ASCII, содержащего десятичные числа по несколько чисел на строке;
3. вывод содержимого текстового файла;
4. добавление новой записи в конец файла;
5. прочитать значение числа, указав его порядковый номер в файле, и вернуть его значение;
6. определить количество чисел в файле.
7. Разработать приложение и выполнить тестирование всех функций. Приложение должно содержать диалоговый интерфейс на основе текстового меню.
8. Контроль открытия и существования файла выполнить в основной программе перед вызовом функции. Перед закрытием файла, проверить отсутствие ошибок ввода и вывода (метод good)/
9. Создать модуль и перенести в него все отлаженные функции. Исключить функции из приложения. Отладить приложение, подключив к нему модуль с функциями.
10. Разработать функции для реализации дополнительных операций, определенных вариантом и сохранить их в модуле с остальными функциями.( Персональный вариант номер 14)
11. Выполнить тестирование приложения в полном объеме.

## Текстовый пример

Копия содержания текстового файла на примере 20 записей:

54 215 dcxv 87 5

azc 966 745 52 58

dsf sad 5 345 3

sdkf df 6 sd 367

## Реализация приложения

### Функционал приложения

#pragma once

#include <fstream>

#include <iostream>

#include <string>

#include <vector>

//#pragma warning(disable : 4996)

using namespace std;

/\* создание текстового файла средствами текстового редактора кодировки ASCII, содержащего десятичные числа по несколько чисел на строке

Предусловие: в качестве параметра должна быть передана строка с .txt на конце

Постусловие: В конце в папке с программой находится текстовый файл с введенным

Ранее названием и с символами вводимыми во время работы функции \*/

void newFile(string filename)

{

ofstream tst(filename);

int num;

int countOnLine = 0;

cout << "Enter numbers:\n";

cin >> num;

while (num)

{

tst << num;

countOnLine++;

if (countOnLine == 5)

{

countOnLine = 0;

tst << "\n";

}

else

{

tst << " ";

}

cin >> num;

}

cout << "Write succesfully!\n";

}

/\* вывод содержимого текстового файла

Предусловие: в качестве параметра должна быть передана строка с .txt на конце

Постусловие: В конце в консоли будет выведено все наполнение файла \*/

void printContent(string filename)

{

ifstream tst(filename);

cout << "Content of file:\n";

while (!tst.eof())

{

string a;

tst >> a;

cout << a << " ";

}

}

/\* добавление новой записи в конец файла

Предусловие: в качестве параметра должна быть передана строка с .txt на конце

Постусловие: В конце файла будет добавлено введенное наполнение \*/

void addToEnd(string filename)

{

ofstream tst(filename, ios::app);

string num;

int countOnLine = 0;

cout << "Enter content:\n";

cin >> num;

while (num != "0")

{

tst << num;

countOnLine++;

if (countOnLine == 5)

{

countOnLine = 0;

tst << "\n";

}

else

{

tst << " ";

}

cin >> num;

}

cout << "Write succesfully!\n";

}

/\* чтение значение числа, указав его порядковый номер в файле, и возвращение его значения

Предусловие: в качестве параметра должна быть передана строка с .txt на конце, номер числа

Постусловие: В конце в консоль будет выведено число, номер которого запрашивали\*/

void printNumByCordinate(string filename, int n)

{

ifstream tst(filename);

int f = 0;

while (!tst.eof())

{

int num = 1;

string a;

tst >> a;

for (int i = 0; i < a.size(); i++)

{

if (!isdigit(a[i]))

{

num = 0;

break;

}

}

if (num)

{

f++;

}

if (f == n)

{

cout << a;

break;

}

}

}

/\* создание нового файла из значений исходного, добавив к каждому числу наибольшее и наименьшее из чисел файла исходного файла.

Предусловие: в качестве параметра должна быть передано 2 строки с .txt на конце

Постусловие: В конце будет создан новый файл из значений исходного, добавив к каждому числу наибольшее и наименьшее из чисел файла исходного файла.\*/

void createNew(string filename, string newFile)

{

ifstream tst(filename);

int min = 9999999;

int max = 0;

while (!tst.eof())

{

int isNum = 1;

string a;

tst >> a;

for (int i = 0; i < a.size(); i++)

{

if (!isdigit(a[i]))

{

isNum = 0;

break;

}

}

if (isNum and !tst.eof())

{

int num = stoi(a);

if (num < min)

{

min = num;

}

if (num > max)

{

max = num;

}

}

}

ifstream tst2(filename);

ofstream newf(newFile);

int countOnLine = 0;

while (!tst2.eof())

{

int isNum = 1;

string a;

tst2 >> a;

for (int i = 0; i < a.size(); i++)

{

if (!isdigit(a[i]))

{

isNum = 0;

break;

}

}

if (isNum and !tst2.eof())

{

int num = stoi(a) + min + max;

newf << num;

countOnLine++;

if (countOnLine == 5)

{

countOnLine = 0;

newf << "\n";

}

else

{

newf << " ";

}

}

else

{

newf << a;

countOnLine++;

if (countOnLine == 5)

{

countOnLine = 0;

newf << "\n";

}

else

{

newf << " ";

}

}

}

}

/\* определение количества чисел в файле

Предусловие: в качестве параметра должна быть передана строка с .txt на конце

Постусловие: В конце в консоль будет выведено – количество чисел в файле\*/

void countNums(string filename)

{

ifstream tst(filename);

int f = 0;

while (!tst.eof())

{

int num = 1;

string a;

tst >> a;

for (int i = 0; i < a.size(); i++)

{

if (!isdigit(a[i]))

{

num = 0;

break;

}

}

if (num and !tst.eof())

{

f++;

}

}

cout << f;

}

### Код основной программы

#include "file.h"

int main()

{

string filename;

cout << "Please, enter file name with extension .txt ath the end:\n";

cin >> filename;

ifstream tst(filename);

if (!tst.is\_open())

{

cout << "Error opening the file";

return 1337;

}

cout << "File opened succesfully\n";

cout << "What do you want to do?\n 1 - Create new file with numbers in it\n 2 - Print content from file";

cout << "\n 3 - Add text to a file\n 4 - Print specific num in file by its cordinate \n ";

cout << "5 - Count numbers in file\n 6 - Copy existing file, adding min and max nums to every number\n";

int isCreate;

cin >> isCreate;

string newсFile;

switch (isCreate)

{

case 1:

newFile(filename);

break;

case 2:

printContent(filename);

break;

case 3:

addToEnd(filename);

break;

case 4:

int n;

cout << "Enter cordinate you want to get:\n";

cin >> n;

printNumByCordinate(filename, n);

break;

case 5:

countNums(filename);

break;

case 6:

cout << "Enter name of new file:\n";

cin >> newсFile;

createNew(filename, newсFile);

default:

break;

}

if (tst.good())

{

cout << "\nAll gone well!\n";

}

tst.close();

}

## Результаты тестирования

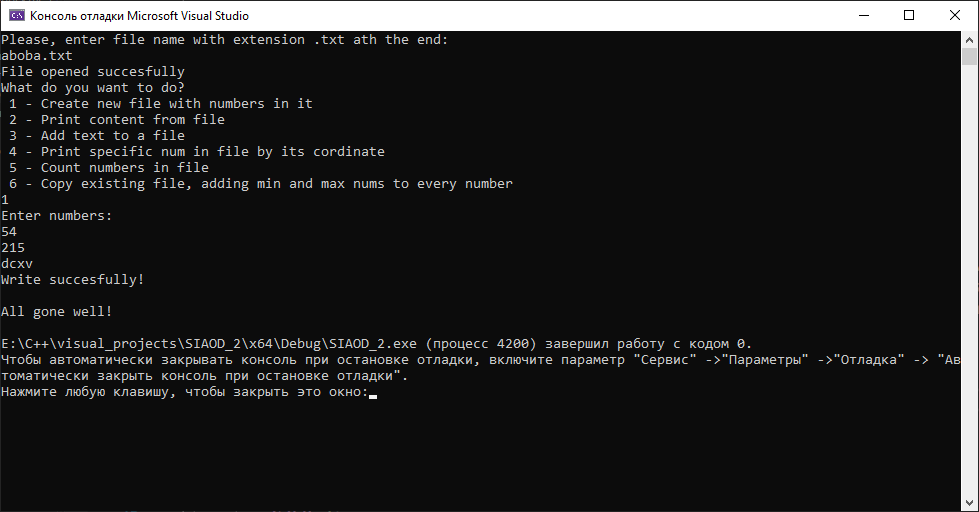


Рисунок 1 – Результат тестирования метода создания файла

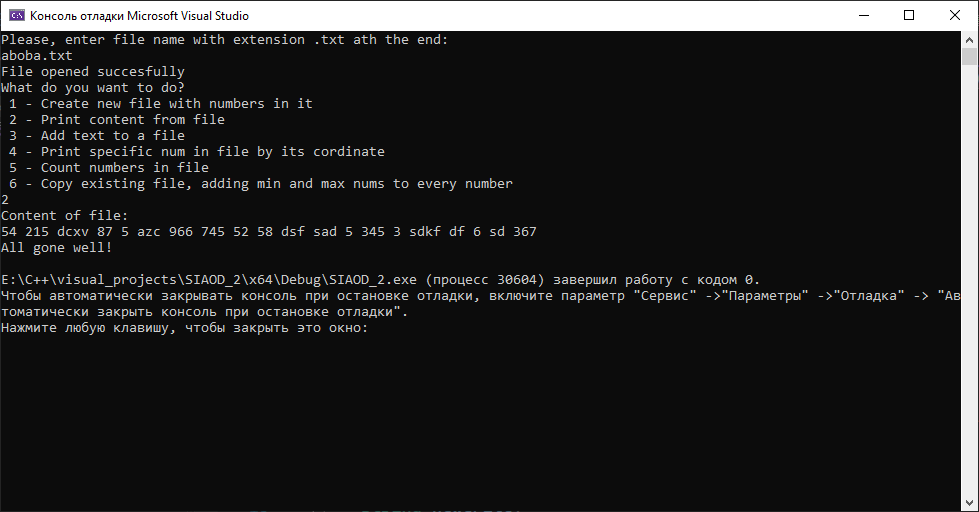


Рисунок 2 – Результат тестирования метода вывода содержимого файла

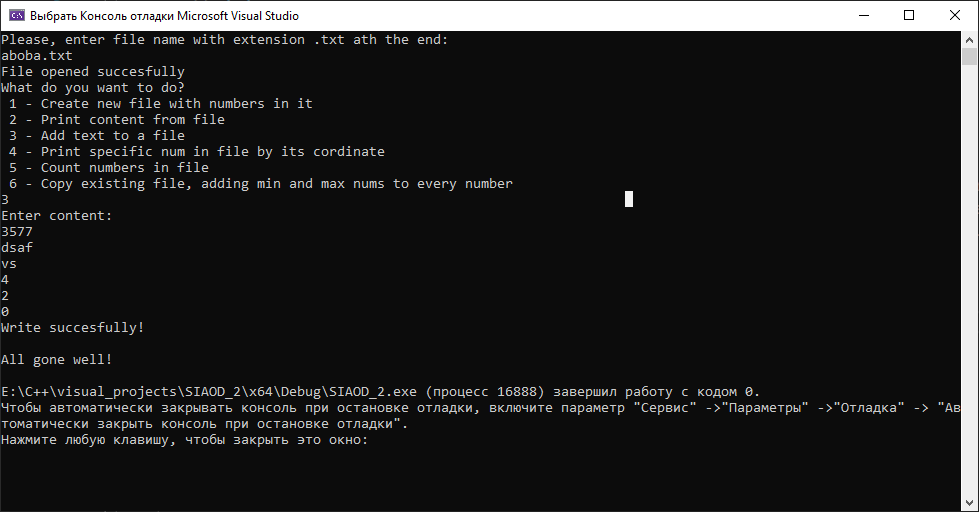


Рисунок 3 – Результат тестирования метода добавления контента в конец файла

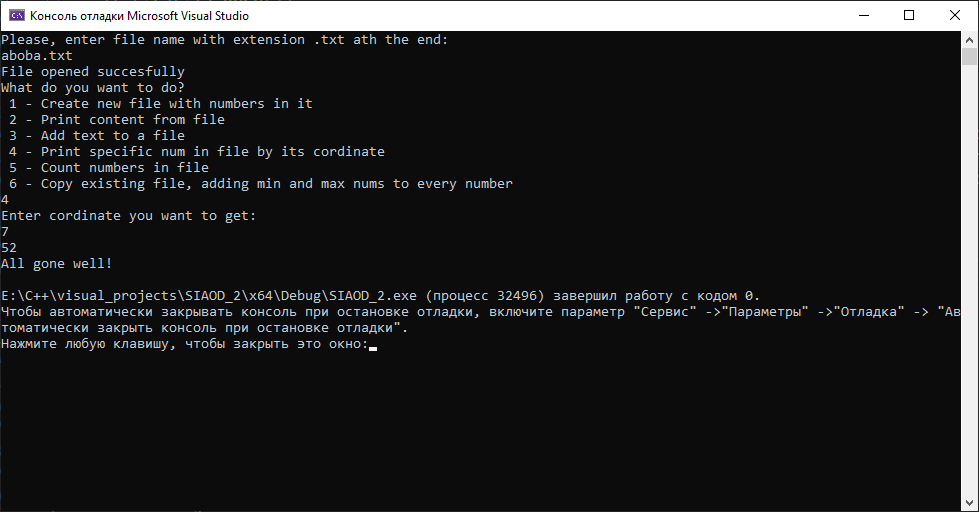


Рисунок 4 – Результат тестирования метода вывода числа по его номеру в файле

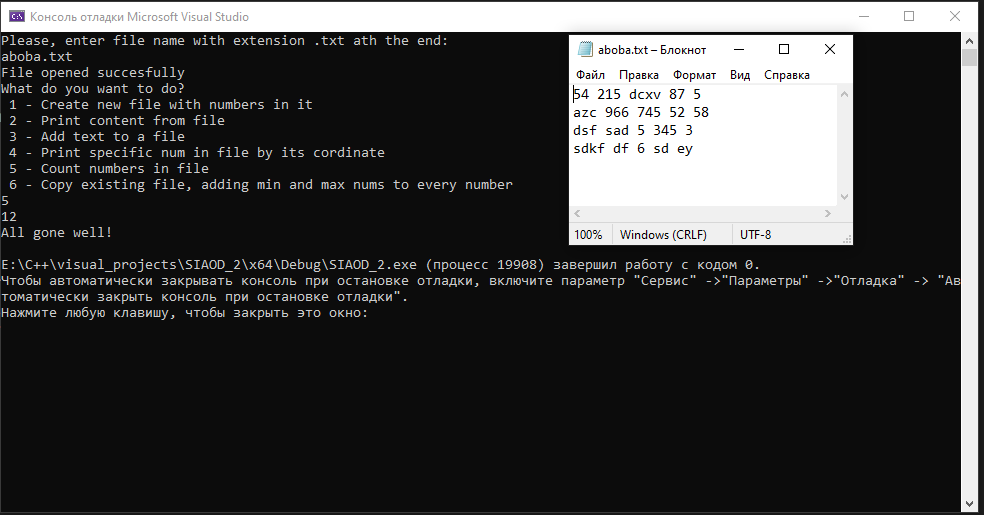


Рисунок 5 – Результат тестирования метода подсчета чисел в файле

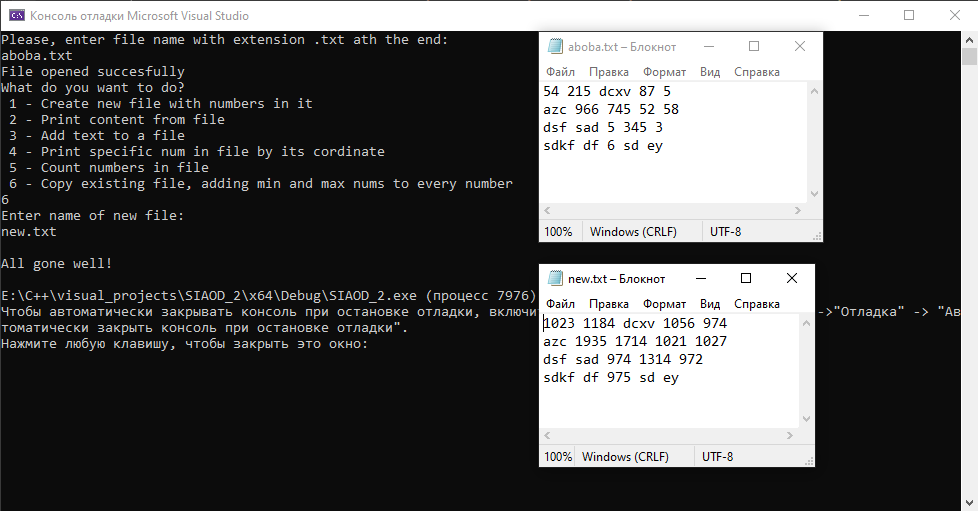


Рисунок 6 – Результат тестирования метода создания нового файла с прибавлением к всем числам минимального и максимального числа

# Задание

Разработать программу управление двоичными файлами с записями фиксированной длины.

## Требования

1. Разработать структуру записи двоичного файла согласно варианту задания.
2. Подготовить тестовые данные в текстовом файле с кодировкой ASCII, в соответствии со структурой записи варианта. При открытии файла выполнить контроль его существования и открытия.

Примечание. Реализация операций по чтению данных из файла будет проще, если значение для каждого поля записи размещать на отдельной строке текстового редактора.

1. Имя файла вводит пользователь.
2. При открытии файла обеспечить контроль существования и открытия файла.
3. При применении механизма прямого доступа к записи файла выполнить контроль присутствия записи с заданным номером в файле.
4. Разработать функции для выполнения операций:

* преобразование тестовых данных из текстового файла в двоичный файл;
* сохранение данных двоичного файла в текстовом, так, чтобы используя их можно было восстановить двоичный файл;
* вывод всех записей двоичного файла;
* доступ к записи по ее порядковому номеру в файле, используя механизм прямого доступа к записи в двоичном файле;
* удаление записи с заданным значением ключа, выполнить путем замены на последнюю запись.
* манипулирование записями в двоичном файле согласно дополнительным операциям, определенным в варианте;

1. Сохраните функции в новом модуле.
2. Разработать приложение, демонстрирующее выполнение всех операций, подключив к нему модуль с функциями.

Выполнить тестирование приложения, продемонстрировав выполнение всех операций.

## Формулировка задания варианта

Персональный вариант номер 14:

|  |  |
| --- | --- |
| Структура записи | Справочная межгорода: код города, название города, страна. |
| Доп. операция | 1. Сформировать в текстовом файле таблицу кодов указанной страны. В строке файла должны храниться: название города и код. 2. Обновить код города по названию города. |

## Тестовый файл

93795 Tokyo Japan

15817 Delhi India

94924 Shanghai China

833 Dhaka Bangladesh

24059 Mexico-City Mexico

## Реализация приложения

### Структура записи двоичного файла и ее размер в байтах

Размер структуры в двоичном формате: 88 байт.

2\*40 для двух переменных строкового типа.

1\*8 под 1 переменную целочисленного типа.

### Изображение структуры двоичного файла с записями фиксированной длины

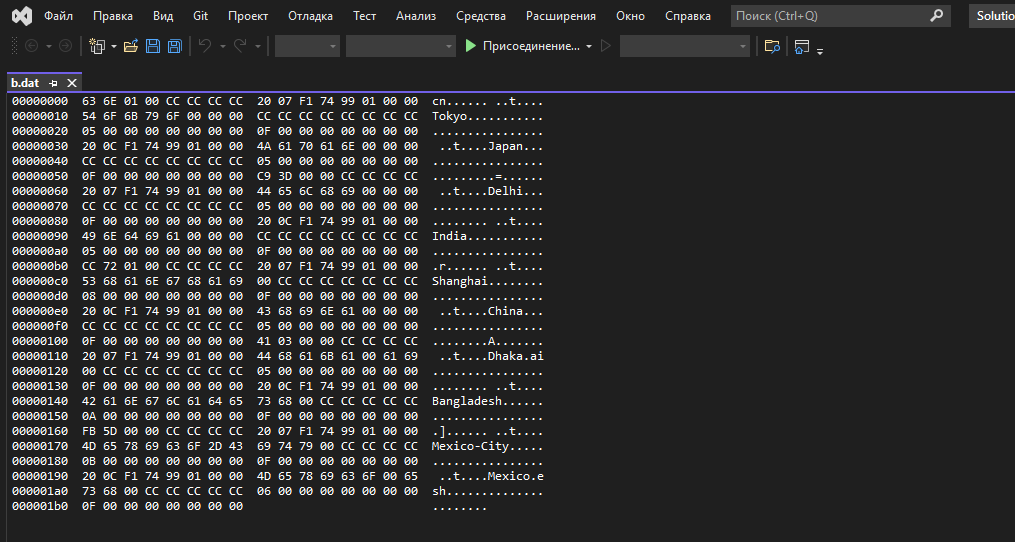


Рисунок 1 - Изображение структуры двоичного файла с записями фиксированной длины

### Функционал приложения

#pragma once

#include <fstream>

#include <iostream>

#include <string>

#include <vector>

#include <filesystem>

using namespace std;

struct City

{

int numCode = 0;

string city;

string country;

};

/\* вывод всех записей двоичного файла

Предусловие: в качестве параметра должна быть передана строка с .txt на конце

Постусловие: В конце будет выведено все содержимое двоичного файла\*/

void printBin(string binName)

{

City tp;

ifstream binFile(binName, ios::in | ios::binary);

binFile.read((char\*)&tp, sizeof(City));

cout << "Containtment of binary file:\n";

while (!binFile.eof())

{

cout << tp.numCode << " ";

cout << tp.city << " ";

cout << tp.country << endl;

binFile.read((char\*)&tp, sizeof(City));

}

cout << "all printed!";

binFile.close();

}

/\* преобразование тестовых данных из текстового файла в двоичный файл

Предусловие: в качестве параметра должна быть передана строка с .txt на конце

Постусловие: В конце будет создан двоичный файл с указанным названием\*/

void createBin(string filename)

{

ifstream scan(filename);

ofstream binaryFile;

string n;

cout << "Enter name of a binary file with .dat: \n";

cin >> n;

binaryFile.open(n, ios::out | ios::binary);

if (!binaryFile)

{

cout << "Binary file not created";

return;

}

string cntr, cty;

int nc;

City tmp;

while (!scan.eof())

{

scan >> nc;

tmp.numCode = nc;

scan >> cty;

tmp.city = cty;

scan >> cntr;

tmp.country = cntr;

binaryFile.write((char\*)&tmp, sizeof(City));

}

if (binaryFile.good())

{

cout << "\nAll gone well!\n";

binaryFile.close();

scan.close();

}

}

/\* 1. Сформировать в текстовом файле таблицу кодов указанной страны. В строке файла должны храниться: название города и код.

Предусловие: в качестве параметра должна быть передана строка с .txt на конце

Постусловие: В конце из данных двоичного файла будет сформирован новый с городами только указанной страны\*/

void createTxt(string binName)

{

City tp;

string newFile;

cout << "Enter name of new file: \n";

cin >> newFile;

ofstream newTxt(newFile);

ifstream binFile(binName, ios::in | ios::binary);

binFile.read((char\*)&tp, sizeof(City));

while (!binFile.eof())

{

newTxt << tp.numCode << " ";

newTxt << tp.city << " ";

newTxt << tp.country << endl;

binFile.read((char\*)&tp, sizeof(City));

}

cout << "all copied!";

binFile.close();

}

/\* доступ к записи по ее порядковому номеру в файле, используя механизм прямого доступа к записи в двоичном файле

Предусловие: в качестве параметра должна быть передана строка с .txt на конце

Постусловие: В конце выведена запись о городе по его номеру\*/

void findInBin(string binName)

{

City tp;

int key;

cout << "Enter key of city you want to find:\n";

cin >> key;

ifstream binFile(binName, ios::in | ios::binary);

binFile.read((char\*)&tp, sizeof(City));

while (!binFile.eof())

{

if (tp.numCode == key)

{

cout << tp.numCode << " ";

cout << tp.city << " ";

cout << tp.country << endl;

binFile.close();

return;

}

binFile.read((char\*)&tp, sizeof(City));

}

cout << "Couldn't Find city with this key!";

binFile.close();

}/\* удаление записи с заданным значением ключа, выполнить путем замены на последнюю запись

Предусловие: в качестве параметра должна быть передана строка с .txt на конце

Постусловие: В конце удалена запись с указанным номером, путем замены ее на последний\*/

void delByKey(string filename)

{

City tmp;

int key, nc;

string t1, t2;

cout << "Enter a key of city you want to delete: \n";

cin >> key;

ifstream scan(filename);

while (!scan.eof())

{

scan >> nc;

tmp.numCode = nc;

scan >> t1;

tmp.city = t1;

scan >> t2;

tmp.country = t2;

}

scan.close();

ifstream read(filename);

ofstream write("tmp" + filename);

City bruh;

while (!read.eof())

{

read >> nc;

bruh.numCode = nc;

read >> t1;

bruh.city = t1;

read >> t2;

bruh.country = t2;

if (!read.eof() and key != bruh.numCode)

{

write << bruh.numCode << " ";

write << bruh.city << " ";

write << bruh.country << endl;

}

else if (!read.eof() and key == bruh.numCode)

{

write << tmp.numCode << " ";

write << tmp.city << " ";

write << tmp.country << endl;

}

}

remove((char\*)&filename);

string oldname = "tmp" + filename;

rename((char\*)&oldname, (char\*)&filename);

if (write.good())

{

cout << "All gone well";

}

}

/\* 2. Обновить код города по названию города

Предусловие: в качестве параметра должна быть передана строка с .txt на конце

Постусловие: В конце код города будет обновлен на указанный\*/

void singleCntryTxt(string filename)

{

City tmp;

int nc;

string t1, t2, newName;

string cntry;

cout << "What contry?\n";

cin >> cntry;

cout << "Name of new file?\n";

cin >> newName;

ifstream scan(filename);

ofstream write(newName);

while (!scan.eof())

{

scan >> nc;

tmp.numCode = nc;

scan >> t1;

tmp.city = t1;

scan >> t2;

tmp.country = t2;

if (tmp.country == cntry)

{

write << tmp.numCode << " ";

write << tmp.city << endl;

}

}

if (write.good())

{

cout << "All gone well";

}

}

void updatecode(string filename)

{

int nc;

string t1, t2, newName;

int newc, oldc;

cout << "Old code?\n";

cin >> oldc;

cout << "New code?\n";

cin >> newc;

ifstream read(filename);

ofstream write("tmp" + filename);

City bruh;

while (!read.eof())

{

read >> nc;

bruh.numCode = nc;

read >> t1;

bruh.city = t1;

read >> t2;

bruh.country = t2;

if (bruh.numCode == oldc)

{

write << newc << " ";

write << bruh.city << " ";

write << bruh.country << endl;

}

else

{

write << bruh.numCode << " ";

write << bruh.city << " ";

write << bruh.country << endl;

}

}

remove((char\*)&filename);

string oldname = "tmp" + filename;

rename((char\*)&oldname, (char\*)&filename);

if (write.good())

{

cout << "All gone well";

}

}

### Код основной программы

#include "Binary.h"

int main()

{

string filename;

string dat;

cout << "Please, enter file name with extension .txt ath the end:\n";

cin >> filename;

ifstream tst(filename);

if (!tst.is\_open())

{

cout << "Error opening the file";

return 1337;

}

cout << "File opened succesfully\n";

cout << "What you want to do?\n 1 - create new binary file\n 2 - copy data from binary to .txt\n ";

cout << "3 - Print content from binary\n 4 - find city by name in binary\n 5 - delete city by key in binary\n ";

cout << "6 - create new .txt with cities only from 1 contry\n 7 - update city code\n";

int code;

cin >> code;

switch (code)

{

case 1:

createBin(filename);

break;

case 2:

cout << "Enter name of a binary file with .dat: \n";

cin >> dat;

createTxt(dat);

break;

case 3:

cout << "Enter name of a binary file with .dat: \n";

cin >> dat;

printBin(dat);

break;

case 4:

cout << "Enter name of a binary file with .dat: \n";

cin >> dat;

findInBin(dat);

break;

case 5:

delByKey(filename);

break;

case 6:

singleCntryTxt(filename);

break;

case 7:

updatecode(filename);

break;

case 8:

cout << sizeof(string);

break;

default:

break;

}

return 0;

}

## Результаты тестирования

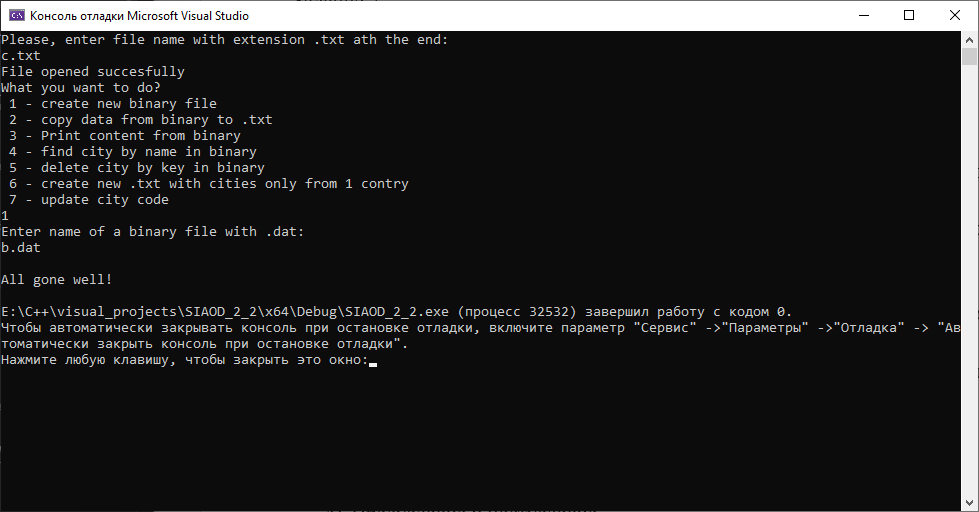


Рисунок 2 – результат тестирования метода создания двоичного файла

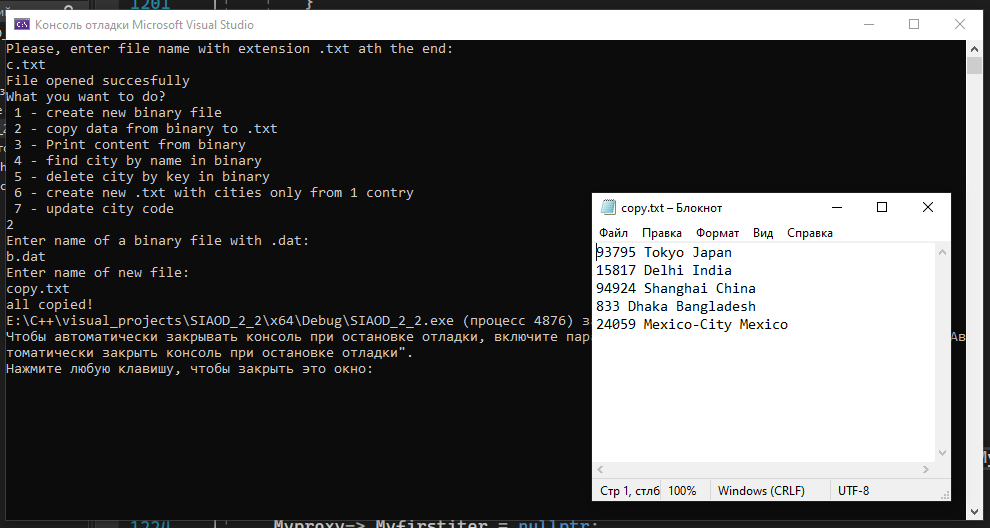


Рисунок 3 – результат тестирования метода создания текстовой копии двоичного файла

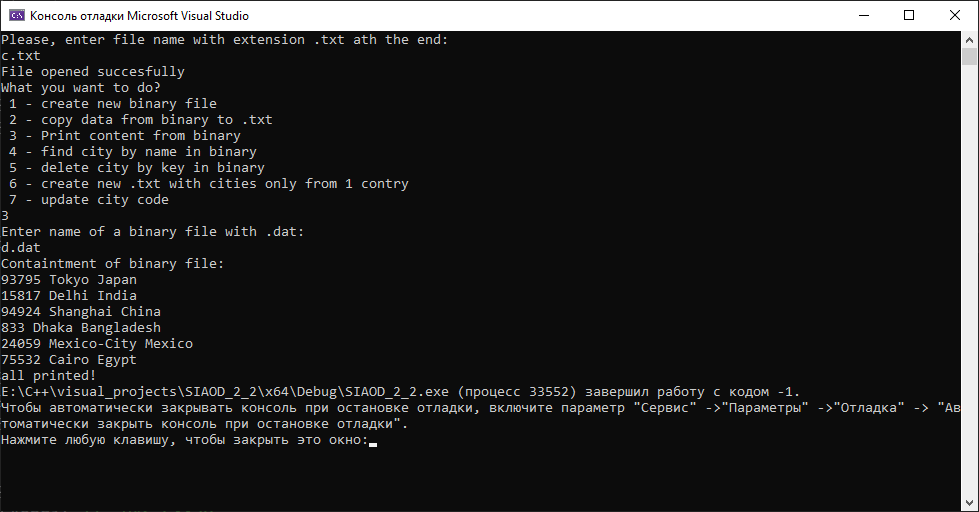


Рисунок 4 – результат тестирования метода вывода содержания двоичного файла

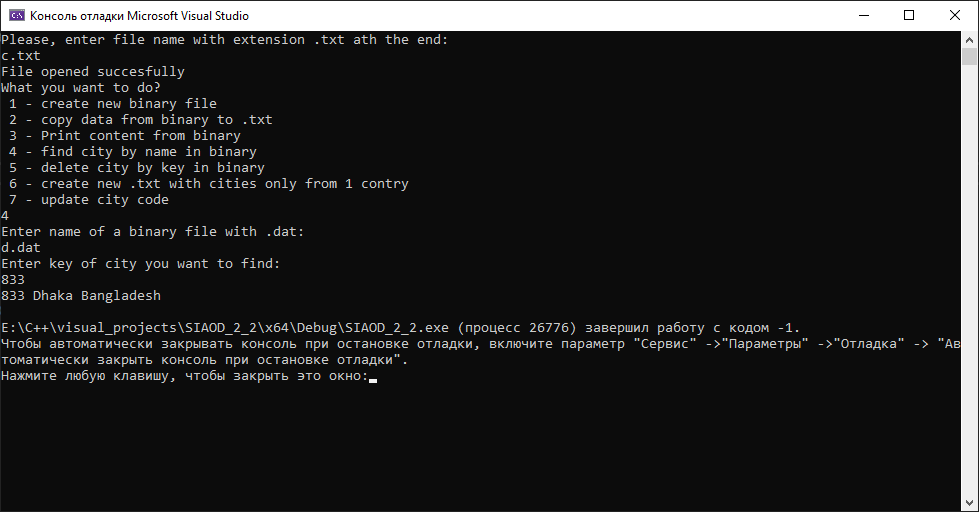


Рисунок 5 – результат тестирования метода вывода города по ключу

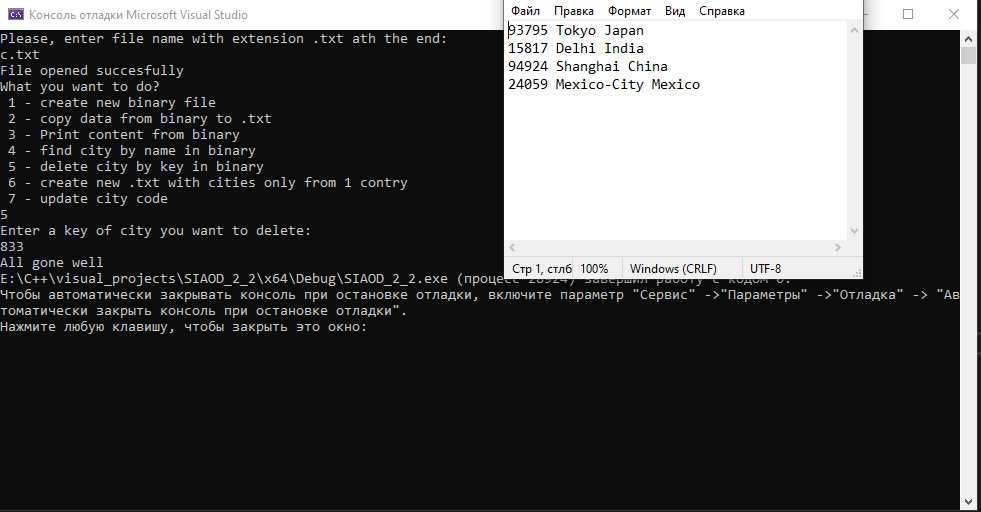


Рисунок 6 – результат тестирования метода удаления значения по ключу

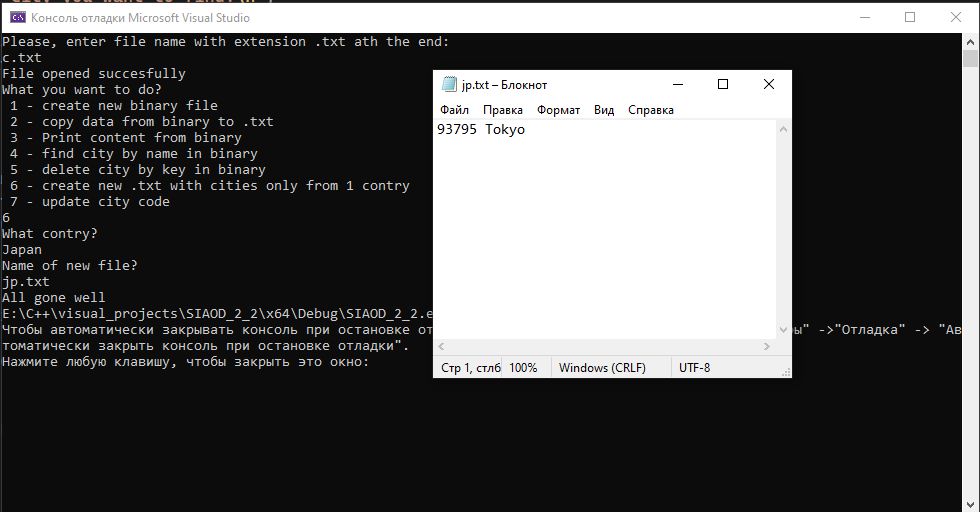


Рисунок 7 – результат тестирования метода создания нового файла с указанной страной

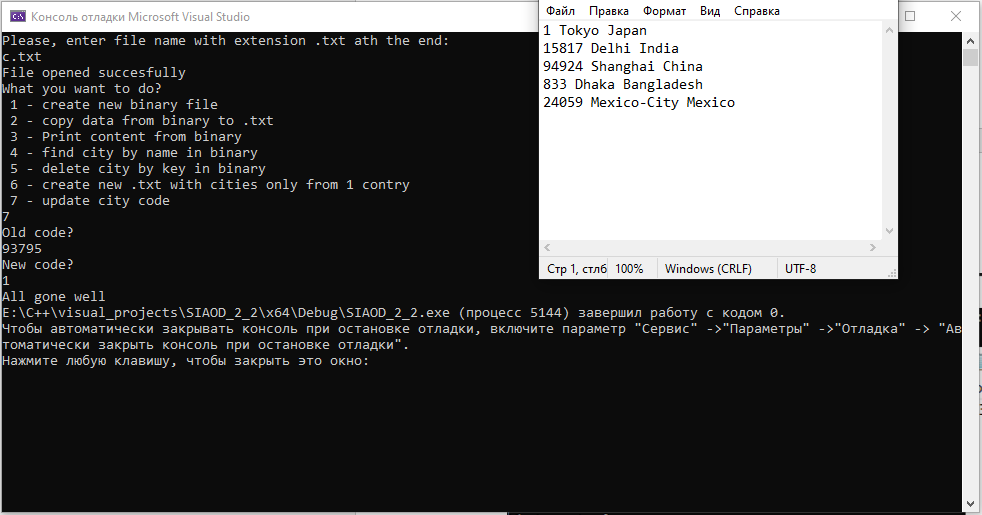


Рисунок 7 – результат тестирования метода обновления кода города