Учебный курс: Технологии программирования

Кафедра: ИС Факультет: ИТиП

Лабораторная работа №10-11-12. Стандартная библиотека.

Выполнил: Трофимов В.А. Группа: 1511

Преподаватель: Повышев В.В.

# Задание

Выполнить задания согласно варианту. При выводе на экран и в файл использовать **только потоки ввода-вывода стандартной библиотеки**. Для описания массивов и прочих структур данных **использовать только структуры данных из стандартной библиотеки**. По возможности, использовать std::string для хранения строковых переменных. Для выполнения всех заданий (сортировка, поиск и проч.) – использовать алгоритмы из <algorithm>.

Необходимо **обрабатывать стандартные исключения** (например, при работе с файлами).

Во всех вариантах числа (или другие элементы) считываются из файла. Обработанные данные записываются в другой файл, имя которого запрашивается у пользователя с консоли. То есть после окончания обработки данных, их необходимо записать в файл.

Считать числа в массив. Спросить у пользователя в каком порядке его отсортировать (по возрастанию или убыванию). Отсортировать.

Считать числа в список. Удалить все четные или нечетные числа (спросить у пользователя, какие именно).

В файле № 1 находятся записи о соответствии номеров телефона (7 цифр) и фамилии. Например:

5671234 Ivanov

3214567 Petrov

9871234 Sidorov

Считать их в ассоциативный массив (std::map).

В файле № 2 находятся номера телефонов. Например:

1112233 9871234 5671234 5556688

Осуществить поиск фамилий из файла №1, соответствующих телефонам из файла №2. Найденные фамилии записать в выходной файл. В данном примере:

Sidorov

Ivanov

# Входные/Выходные данные

## Консоль

PROBLEM A.

31 elements were succesfully read from file.

How would you like to sort your numbers (Increase(Y)/Decrease(N))?

Okay. I'll sort them by decrease

Please enter a filename for saving result: 123.txt

Saving complete succesfully.

PROBLEM F.

31 elements were succesfully read from file.

How would you like to modify your data (delete all even numbers(Y)/delete all od

d numbers(N))?

Okay. I'll delete all even numbers.

Please enter a filename for saving result: 456.txt

Saving complete succesfully.

Problem G.

3 elements were succesfully read from file.

4 elements were succesfully read from file.

Please enter a filename for saving result: 789.txt

Saving complete succesfully.

## Файл inputA.txt

3 1 4 1 5 9 2 6 5 3 5 8 9 7 9 3 2 3 8 4 6 6 4 3 3 8 3 2 7 9 5

## Файл outputA.txt

9 9 9 9 8 8 8 7 7 6 6 6 5 5 5 5 4 4 4 3 3 3 3 3 3 3 2 2 2 1 1

## Файл inputB.txt

3 1 4 1 5 9 2 6 5 3 5 8 9 7 9 3 2 3 8 4 6 6 4 3 3 8 3 2 7 9 5

## Файл outputB.txt

3 1 1 5 9 5 3 5 9 7 9 3 3 3 3 3 7 9 5

## Файл inputC.txt

5671234 Ivanov

3214567 Petrov

9871234 Sidorov

## Файл queryC.txt

1112233 9871234 5671234 5556688

## Файл outputC.txt

Sidorov

Ivanov

# Исходный текст

## Файл main.cpp

#include <algorithm>

#include <cstdlib>

#include <fstream>

#include <iostream>

#include <conio.h>

#include <string>

#include <vector>

#include <list>

#include <map>

#include <exception>

using namespace std;

bool isYes(char c) {

return (c == 237) || (c == 205) || (c == 121) || (c == 89) || (c == -83) || (c == -115);

}

bool isNo(char c) {

return (c == 210) || (c == 242) || (c == 110) || (c == 78) || (c == -30) || (c == -110);

}

bool increaseCompare(int left, int right) {

return left < right;

}

bool decreaseCompare(int left, int right) {

return left > right;

}

bool isRestricted(char c) {

return (c == '\\') || (c == '/') || (c == ':') || (c == '\*') ||

(c == '?') || (c == '"') || (c == '<') || (c == '>') || (c == '|');

}

bool isFilename(string s) {

if (s.length() == 0) return false;

else {

bool space = true; char c;

for (unsigned i = 0; i < s.length(); i++) {

c = s.at(i); if (c != ' ') space = false;

if (isRestricted(c)) return false;

}

if (space) return false;

else return true;

}

}

bool isOdd(int i) {

return (i % 2 == 1);

}

bool isEven(int i) {

return (i % 2 == 0);

}

int main() {

//Problem A

cout << "PROBLEM A." << endl << endl;

vector<int> v; int temp; int count = 0; bool flagYes, flagNo, flag; string filename;

ifstream in;

try {

in.open("inputA.txt");

} catch(exception & e) {

cout << "Error occured during file opening: " << e.what() << endl;

return 0;

}

while(in.good()) {

in >> temp;

v.push\_back(temp);

count++;

} in.close();

if (count) {

cout << count << " elements were succesfully read from file." << endl;

cout << "How would you like to sort your numbers (Increase(Y)/Decrease(N))?" << endl;

char key; do {

key = getch();

flagYes = isYes(key); flagNo = isNo(key);

} while (!flagNo && !flagYes);

cout << "Okay. I'll sort them by " << (flagYes ? "increase." : "decrease") << endl;

stable\_sort(v.begin(), v.end(), (flagYes ? increaseCompare : decreaseCompare));

cout << "Please enter a filename for saving result: ";

do {

getline(cin, filename); flag = isFilename(filename);

if (!flag) cout << "You enter an invalid filename. Please try again." << endl;

} while (!flag);

ofstream out;

try {

out.open(filename, ostream::out);

} catch(exception & e) {

cout << "Error occured during file opening: " << e.what() << endl;

return 0;

}

for (unsigned i = 0; i < v.size(); i++) out << v.at(i) << " ";

out.close(); v.clear(); filename.clear();

cout << "Saving complete succesfully." << endl;

} else cout << "There are no succesfully read numbers. I will not save anything in the file." << endl;

//Problem F

cout << endl << endl << "PROBLEM F." << endl << endl;

list<int> l; count = 0;

try {

in.open("inputB.txt");

} catch(exception & e) {

cout << "Error occured during file opening: " << e.what() << endl;

return 0;

}

while(in.good()) {

in >> temp;

l.push\_back(temp);

count++;

} in.close();

if (count) {

cout << count << " elements were succesfully read from file." << endl;

cout << "How would you like to modify your data (delete all even numbers(Y)/delete all odd numbers(N))?" << endl;

char key; do {

key = getch();

flagYes = isYes(key); flagNo = isNo(key);

} while (!flagNo && !flagYes);

cout << "Okay. I'll delete all " << (flagYes ? "even " : "odd ") << "numbers." << endl;

l.remove\_if((flagYes ? isEven : isOdd)); l.resize(l.size());

cout << "Please enter a filename for saving result: ";

do {

getline(cin, filename); flag = isFilename(filename);

if (!flag) cout << "You enter an invalid filename. Please try again." << endl;

} while (!flag);

ofstream out;

try {

out.open(filename, ostream::out);

} catch(exception & e) {

cout << "Error occured during file opening: " << e.what() << endl;

return 0;

}

for (list<int>::iterator i = l.begin(); i != l.end(); i++) out << \*i << " ";

out.close(); l.clear(); filename.clear();

cout << "Saving complete succesfully." << endl;

} else cout << "There are no succesfully read numbers. I will not save anything in the file." << endl;

//Problem G

cout << endl << endl << "Problem G." << endl << endl;

map<long, string> m; string surname; count = 0; long phone;

try {

in.open("inputC.txt");

} catch(exception & e) {

cout << "Error occured during file opening: " << e.what() << endl;

return 0;

}

while(in.good()) {

in >> phone >> surname;

m[phone] = surname;

count++;

} in.close();

if (count) {

cout << count << " elements were succesfully read from file." << endl;

try {

in.open("queryC.txt");

} catch(exception & e) {

cout << "Error occured during file opening: " << e.what() << endl;

return 0;

}

vector<long> p; count = 0;

while(in.good()) {

in >> phone;

p.push\_back(phone);

count++;

} in.close();

if (count) {

cout << count << " elements were succesfully read from file." << endl;

cout << "Please enter a filename for saving result: ";

do {

getline(cin, filename); flag = isFilename(filename);

if (!flag) cout << "You enter an invalid filename. Please try again." << endl;

} while (!flag);

ofstream out;

try {

out.open(filename, ostream::out);

} catch(exception & e) {

cout << "Error occured during file opening: " << e.what() << endl;

return 0;

}

for (unsigned i = 0; i < p.size(); i++) {

map<long, string>::iterator it = m.find(p.at(i));

if (it != m.end()) out << (it->second) << endl;

}

out.close(); p.clear(); m.clear(); filename.clear();

cout << "Saving complete succesfully." << endl;

} else {

cout << "There are no succesfully read numbers. I will not save anything in the file." << endl;

return 0;

}

} else cout << "There are no succesfully read numbers. I will not save anything in the file." << endl;

return 0;

}