**按以下要求编写程序**

**请各位考生从老师处获取数据文件info.txt，然后将该数据文件手动保存在D盘根目录下。**

**题目说明：**

**目前新冠肺炎病毒检测主要手段是“核酸检测”，它是针对新冠病毒基因组中开放读码框1ab（open reading frame 1ab，ORF1ab）和核壳蛋白（nucleocapsid protein，N）进行检测。其检测结果称为Ct值，是指通过RT-PCR技术实时监测反应管中荧光经历多少循环数达到预定的阈值。那么如何根据Ct值判断是否阳性？《新型冠状病毒肺炎防控方案》第七版之前规则如下：**

1. **阴性：无Ct值或Ct≥40。**
2. **阳性：Ct值<37，可报告为阳性。**
3. **灰区：Ct值在37-40之间，建议重复实验。**

**文件info.txt中存储了若干人员的核酸检测Ct值。每个人员的信息占一行，每行包括身份证号（其中如果有字母X，一定是大写）和核酸检测Ct值（一定是整数），二者之间用空格分开。**

**请按要求依次完成如下操作：**

1. **编写函数readInfo，读取出文件中的所有数据并存在struct info类型的infos向量中；**
2. **编写函数showInfo，显示第1步所生成向量的前5条数据，（如果不足5条则全部显示）显示格式如下：**

|  |  |
| --- | --- |
| ***身份证号（占20列，左对齐）*** | ***Ct值(占5列，右对齐)*** |

1. **编写函数delInfo，删除infos向量中身份证号码长度不合法的数据（中华人民共和国第二代身份证号码应该为18位）；**
2. **编写函数SortInfo，按照Ct值从小到大对数据进行排序，当Ct相同时按照身份证号的ASCII码从小到大排序；**
3. **编写函数groupInfo统计出阳性、灰区和阴性的人数并用向量返回；**
4. **编写函数writeInfo将排序后的Ct值在灰区的人员信息全部输出到D盘根文件夹下的文本文件out.txt中。输出时每行一条数据，先输出身份证号再输出Ct值，二者之间用英文逗号分开；**
5. **编写函数找出身份证号码校验位不正确的人员，并把身份证号码存储在string类型的向量res中。中国人民共和国第二代居民身份证号码是18位编码，第18位是校验码。校验规则如下：**

* **将身份证号码前面的17位数分别乘以不同的系数。从第1位到第17位的系数分别为：7、9、10、5、8、4、2、1、6、3、7、9、10、5、8、4、2**
* **将这17位数字和系数分别相乘的结果相加**
* **用上面累加结果除以11，取余数。**
* **余数只可能有0、1、2、3、4、5、6、7、8、9、10这11个数字。其分别对应的最后一位身份证的号码为1、0、X、9、8、7、6、5、4、3、2。**

1. **编写一个重载函数showInfo显示res的全部结果到屏幕，一行一个身份证号，如果res为空，则显示“身份证号全部正确”。**

**程序中需要使用的结构体info定义如下:**

**struct info**

**{**

**string sfzh;//身份证号**

**int ct; //Ct值**

**};**

**main函数如下：**

**int main()**

**{**

**vector<struct info> infos;**

**if (readInfo("d:\\input.txt", infos) == false)**

**{**

**cout << "打开文件失败，请确认文件路径" << endl;**

**return 0;**

**}**

**cout << "合计有"<<infos.size()<<"条数据。";**

**cout << "前5条数据为：" << endl;**

**showInfo(infos);**

**int count = delInfo(infos);**

**cout << "删除了身份证不合法的" << count << "条，剩余" << infos.size() << "条" << endl;**

**sortInfo(infos);**

**cout << "排序后数据为：" << endl;**

**showInfo(infos,infos.size());**

**vector<int> counts = groupInfo(infos);**

**cout << "其中陽性：" << counts[0] << "人"<< endl;**

**cout << "其中灰区：" << counts[1] << "人" << endl;**

**cout << "其中阴性：" << counts[2] << "人" << endl;**

**if (writeInfo("d:\\out.txt", infos) == true)**

**{**

**cout << "写文件成功"<<endl;**

**}**

**else**

**{**

**cout << "打开文件失败"<<endl;**

**}**

**vector <string> res = findInvalidSfzh(infos);**

**cout << "身份证号错误检查结果如下：" << endl;**

**showInfo(res);**

**return 0;**

**}**

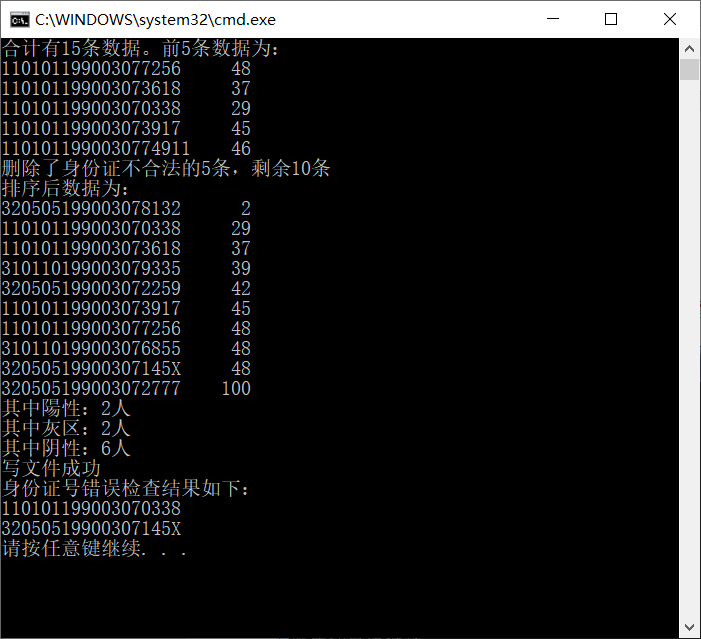
**注意: 不允许修改main函数,每修改一处,扣3分;**

**评分标准**

**（编程题满分为100分）**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **大项** | **子项** | **评分项** | **应得分** | **实得分** |
| **正**  **确**  **性**  **90分** | **结果（90分）**  **若编译链接不通过（本部分得0分）** | **readInfo函数** | **15** |  |
| **delInfo函数** | **10** |  |
| **showInfo函数** | **10** |  |
| **sortInfo函数** | **10** |  |
| **groupInfo** | **15** |  |
| **writeInfo函数** | **10** |  |
| **findInvalidSfzh函数** | **10** |  |
| **showInfo重载函数** | **10** |  |
| **上述各项都不得分** | **见编译子项** | **本项不得分** |
| **程序运行出现异常** | **-10** |  |
| **程序死循环** | **-10** |  |
| **修改main函数** | **-3（每处修改）** |  |
| **可**  **读**  **性**  **10分** | **缩进对齐（4分）** | **正确运用缩进对齐规则** | **4** |  |
| **有缩进对齐但不完全符合要求** | **3** |  |
| **没有使用缩进对齐规则** | **2** |  |
| **注释（3分）** | **有详细且正确的注释** | **3** |  |
| **有注释，但不够详细** | **2** |  |
| **完全没有注释** | **0** |  |
| **变量命名（3分）** | **变量命名有规则** | **3** |  |
| **变量命名有规则、但规则使用不一致** | **2** |  |
| **变量命名无规则** | **0** |  |

**程序运行结果样例截图(不一定是标准答案)：**



#include<iostream>

#include<iomanip>

#include<string>

#include<algorithm>

#include<cmath>

#include <vector>

#include <fstream>

#include <unordered\_map>

using namespace std;

struct info

{

string sfzh;//身份证号

int ct; //Ct值

};

//1

bool readInfo(string filename, vector<struct info>& infos)

{

ifstream file(filename);

if (!file.is\_open()) //这个逻辑应该不成立

{

return false;

}

struct info temp;

while (file >> temp.sfzh >> temp.ct)

{

infos.push\_back(temp);

}

file.close();

return true;

}

//2

void showInfo(vector<struct info>& infos,int n=5)

{

for (int i = 0; i < n && i<infos.size(); i++)

{

cout << left << setw(20) << infos[i].sfzh

<< right << setw(5) << infos[i].ct << endl;;

}

}

//3

int delInfo(vector<struct info>& infos)

{

int count = 0;

for (int i = 0; i < infos.size(); )

{

if (infos[i].sfzh.size() != 18)

{

infos.erase(infos.begin() + i);

count++;

}

else

{

i++;

}

}

return count;

}

//4

bool compare(info a, info b)

{

if (a.ct != b.ct) return a.ct < b.ct;

else if (a.sfzh != b.sfzh)return a.sfzh < b.sfzh;

}

void sortInfo(vector<info>& infos)

{

sort(infos.begin(), infos.end(), compare);

}

//5

vector<int> groupInfo(vector<info>&infos)

{

vector <int> a(3);

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

{

if (infos[i].ct >= 40)

a[2]++;

else if (infos[i].ct < 40 && infos[i].ct >= 37)

a[1]++;

else

a[0]++;

}

return a;

}

//6

bool writeInfo(string new\_filename, vector<info>& infos)

{

ofstream outfile(new\_filename);

for (const auto& inf : infos)

{

if (inf.ct >= 37 && inf.ct < 40)

{

outfile << inf.sfzh << " " << inf.ct << endl;

}

}

outfile.close();

return true;

}

//7

vector<string> findInvalidSfzh(vector<info>& infos)

{

vector<string> res;

int coefficients[] = { 7, 9, 10, 5, 8, 4, 2, 1, 6, 3, 7, 9, 10, 5, 8, 4, 2 }; // 系数数组

char checkCodes[] = { '1', '0', 'X', '9', '8', '7', '6', '5', '4', '3', '2' }; // 校验码数组

for (const auto& info : infos)

{

int sum = 0;

for (int j = 0; j < 17; ++j) {

sum += (info.sfzh[j] - '0') \* coefficients[j]; // info.sfzh[j] - '0' 是将字符形式的数字转换为整数

}

int yu = sum % 11;

if (info.sfzh[17] != checkCodes[yu])

{

res.push\_back(info.sfzh);

}

}

return res;

}

//8 重载函数

void showInfo(vector<string>& res)

{

if (res.size() == 0) cout << "身份证号全部正确";

else

{

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

{

cout << res[i] << endl;

}

}

}

int main()

{

vector<struct info> infos;

if (readInfo("C:\\Users\\lenovo\\Desktop\\C++\\考试试题\\20220419\\input.txt", infos) == false)

{

cout << "打开文件失败，请确认文件路径" << endl;

return 0;

}

cout << "合计有" << infos.size() << "条数据。";

cout << "前5条数据为：" << endl;

showInfo(infos);

int count = delInfo(infos);

cout << "删除了身份证不合法的" << count << "条，剩余" << infos.size() << "条" << endl;

sortInfo(infos);

cout << "排序后数据为：" << endl;

showInfo(infos, infos.size());

vector<int> counts = groupInfo(infos);

cout << "其中陽性：" << counts[0] << "人" << endl;

cout << "其中灰区：" << counts[1] << "人" << endl;

cout << "其中阴性：" << counts[2] << "人" << endl;

if (writeInfo("d:\\out.txt", infos) == true)

{

cout << "写文件成功" << endl;

}

else

{

cout << "打开文件失败" << endl;

}

vector <string> res = findInvalidSfzh(infos);

cout << "身份证号错误检查结果如下：" << endl;

showInfo(res);

return 0;

}