《C++程序设计基础》作品

# 一，疫情人员管理

## 完整代码：

#include<bits/stdc++.h>

using namespace std;

class Student

{

public:

char name[200];

char ID[200];

char phone[200];

char color[200];

char f[200];

Student\* next;

Student() :next(nullptr) {}

Student(char\* Name, char\* Id, char\* Phone, char\* Color, char\* F)

{

strcpy(ID, Id);

strcpy(name, Name);

strcpy(phone, Phone);

strcpy(color, Color);

strcpy(f, F);

next = nullptr;

}

void creat(char\* Name, char\* Id, char\* Phone, char\* Color, char\* F)

{

strcpy(ID, Id);

strcpy(name, Name);

strcpy(phone, Phone);

strcpy(color, Color);

strcpy(f, F);

next = nullptr;

}

Student(Student& n)

{

strcpy(ID, n.ID);

strcpy(name, n.name);

strcpy(phone, n.phone);

strcpy(color, n.color);

strcpy(f, n.f);

next = nullptr;

}

};

class Link {

public:

Student\* head;

Link() {head = new Student();}

void addStudentTail(Student\* s)

{

Student\* p = head;

while (p->next != nullptr)

{

p = p->next;

}

Student\* newStudent = new Student(\*s);

p->next = newStudent;

}

void traverse()

{

Student\* p = head->next;

while (p != nullptr)

{

cout << p->name << " " << p->ID << " " << p->phone << " " << p->color << " " << p->f << endl;

p = p->next;

}

cout << endl;

}

bool func(char\* x, char\* y)

{

for (int i = 0; i < 2; i++)

{

if (x[i] != y[i])return false;

}

return true;

}

void getColor(char\* na)

{

Student\* p = head->next;

while (p != nullptr)

{

if (func(p->name,na))

{

cout << p->color << " " << p->f << endl;

break;

}

p = p->next;

}

}

void displayUndoing()

{

Student\* p = head->next;

while (p != nullptr)

{

if (p->f[0] == -73)cout << p->name << " " << p->ID << " " <<" "<<p->phone<<" "<< p->color << " " << p->f << endl;

p = p->next;

}

}

//绿码：-62,-52,-62,-21,\0

bool green(char\* x)

{

if (x[0] != -62)return false;

if (x[1] != -52)return false;

if (x[2] != -62)return false;

if (x[3] != -21)return false;

return true;

}

//红码： -70,-20,-62,-21,'\0'

bool red(char\* x)

{

if (x[0] != -70)return false;

if (x[1] != -20)return false;

if (x[2] != -62)return false;

if (x[3] != -21)return false;

return true;

}

//黄码：-69,-58,-62,-21,\0

bool yellow(char\* x)

{

if (x[0] != -69)return false;

if (x[1] != -58)return false;

if (x[2] != -62)return false;

if (x[3] != -21)return false;

}

void setfile()

{

Student\* p = head->next;

ofstream foutgreen("绿码.txt");

ofstream foutyellow("黄码.txt");

ofstream foutred("红码.txt");

while (p->next != nullptr)

{

if (green(p->color))

{

foutgreen << p->name << " " << p->ID << " " << p->phone << " " << p->color << " " << p->f << endl;

}

if (red(p->color))

{

foutred << p->name << " " << p->ID << " " << p->phone << " " << p->color << " " << p->f << endl;

}

if (yellow(p->color))

{

foutyellow << p->name << " " << p->ID << " " << p->phone << " " << p->color << " " << p->f << endl;

}

p = p->next;

}

}

};

int main()

{

ifstream fin("疫情管理人员2.txt");

if (!fin)

{

cout << "文件打开失败" << endl;

exit(1);

}

//2 8 11 2 2

Link link;

char c[1153];

fin.getline(c, sizeof(c));

int i = 0;

Student s[8];

while (i < 8)

{

char name[1024];

char ID[1024];

char phone[1024];

char color[1024];

char f[1024];

fin.getline(name, sizeof(name), ' ');

fin.getline(ID, sizeof(ID), ' ');

fin.getline(phone, sizeof(phone), ' ');

fin.getline(color, sizeof(color), ' ');

fin.getline(f, sizeof(f), '\n');

s[i].creat(name, ID, phone, color, f);

link.addStudentTail(&s[i]);

//link.traverse();

i++;

}

int p;

cout << "功能：" << endl;

cout << "(1)根据姓名查询学生的健康码和核酸检测状态" << endl;

cout << "(2)显示所有未做核酸检测的学生信息" << endl;

cout << "请输入想实现功能的序号：" << endl;

cin >> p;

//根据学生姓名查询学生健康码和核酸检测状态

if (p == 1)

{

char ch[200];

cout << "请输入查询学生名字：" << endl;

cin >> ch;

link.getColor(ch);

cout << endl;

}

//显示所有未做核酸检测的学生信息

if(p==2)link.displayUndoing();

//根据健康码状态将人员数据储存到文件

link.setfile();

fin.close();

}

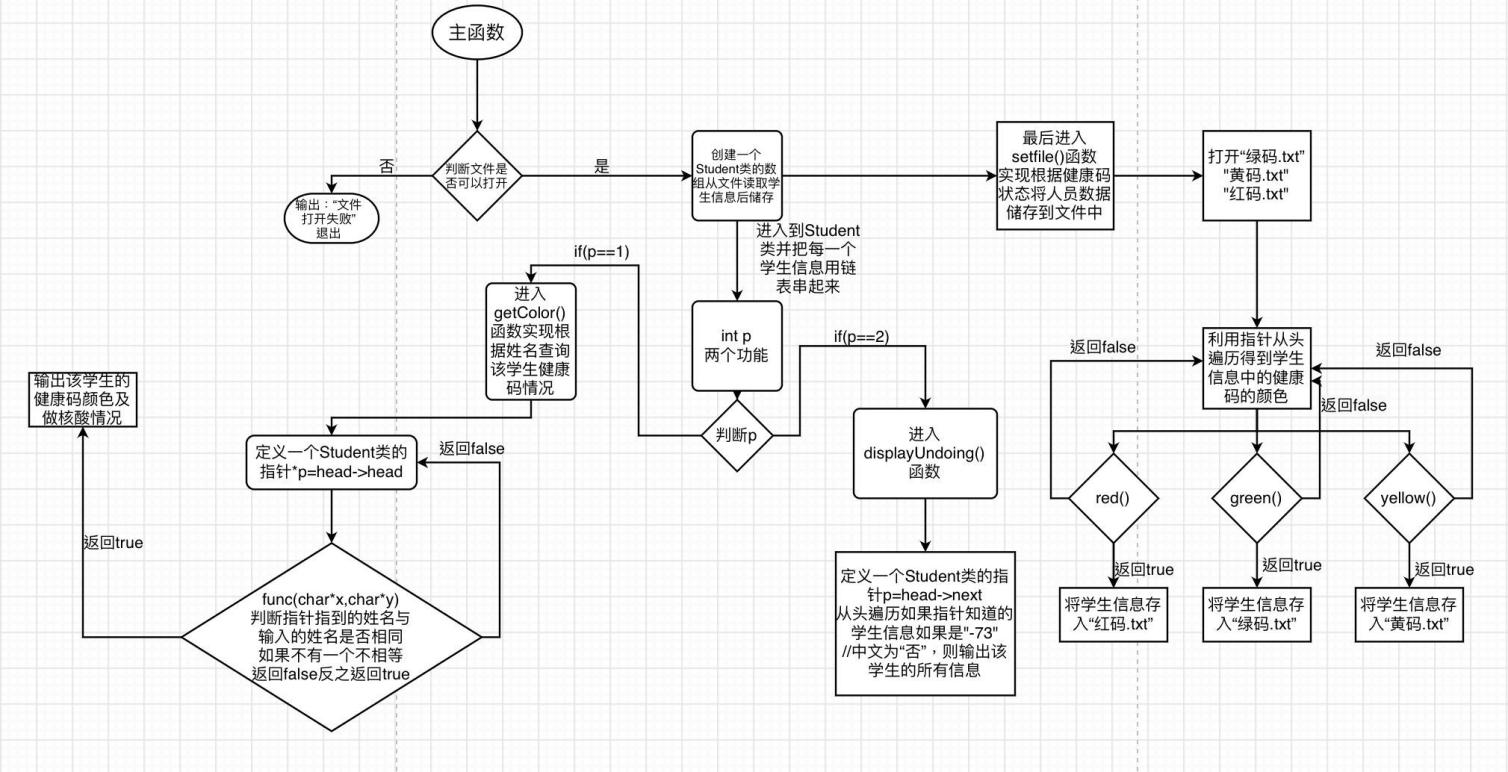
## 核心实现原理

1. 从文件中读取一个学生的信息后，用字符数组储存学生信息，getline()每次读取到空格，最后读取到换行符‘\n’，最后讲读取到的信息储存到Student类里，并加入到链表中，将每一个学生信息都串起来
2. 在Student类里使用strcpy()赋值到Student()类里。
3. 第一个功能：根据姓名查询学生健康码和核酸状况；通过我写的getColor(char\*)函数传入想要查询的学生姓名，运用链表指针，遍历寻找进入到func()函数判断符合条件的学生，输出该学生的健康码颜色以及是否做核酸。
4. 第二个功能：显示所有未做核酸检测的学生信息；进入displayUndoing()函数，仍是用链表指针持续遍历，为否的学生输出该学生信息，通过断点遍历发现中文“否”字符值为-73因此(p->f[0]==-73)判断
5. 第三个功能：根据健康码状态将人员数据储存到各个文件；通过写三个判断健康码颜色，通过断点遍历发现红码，绿码，黄码在字符数组中分别为以下形式储存：

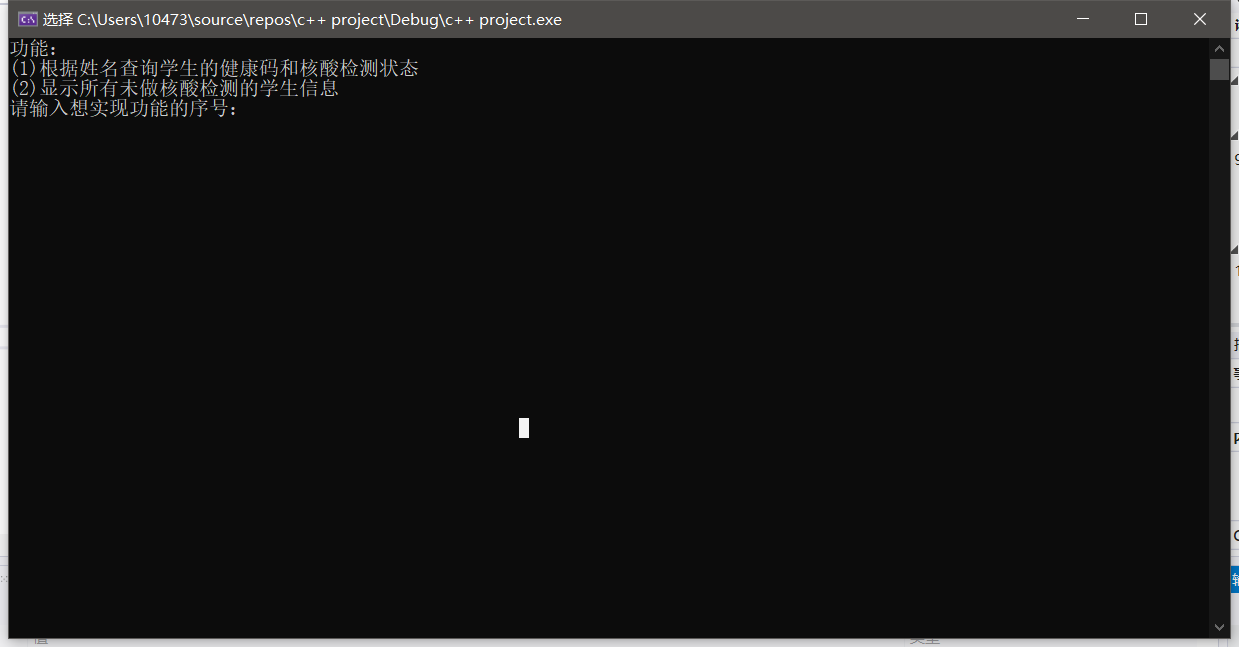
绿码：-62,-52,-62,-21,\0; 红码： -70,-20,-62,-21,'\0'；绿码：-62,-52,-62,-21,\0

因此,利用这些数值判断每位学生健康码，存入相应的文件中。

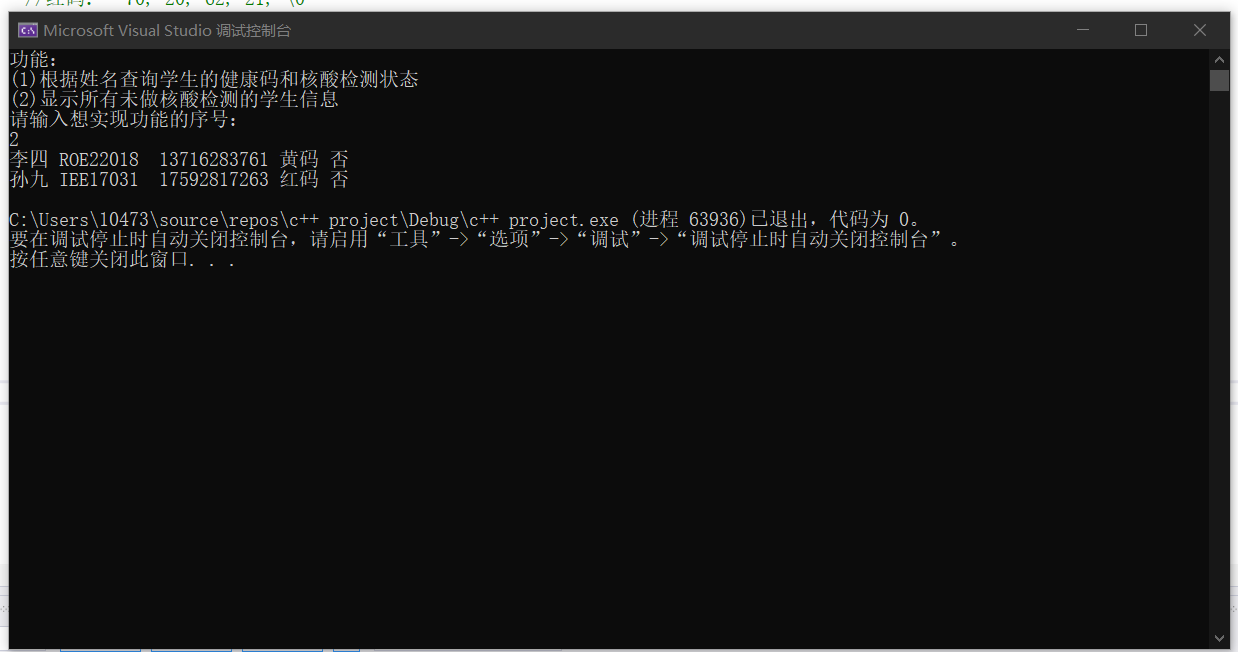
## 流程设计



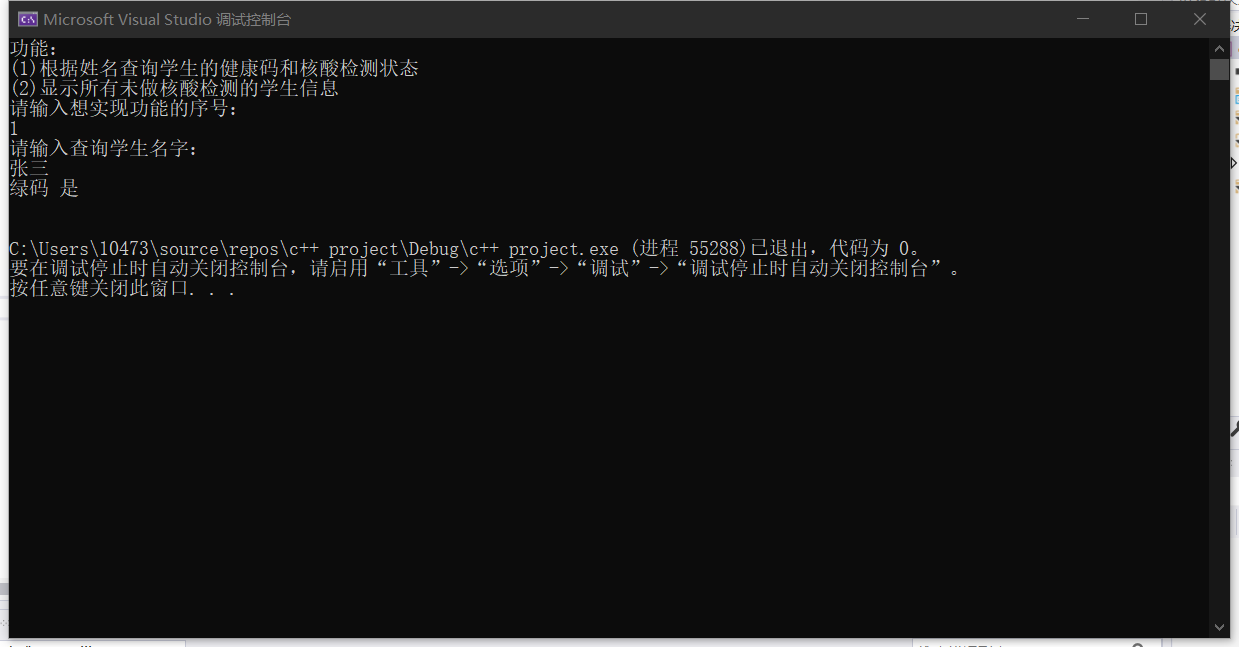
## 结果截图及说明

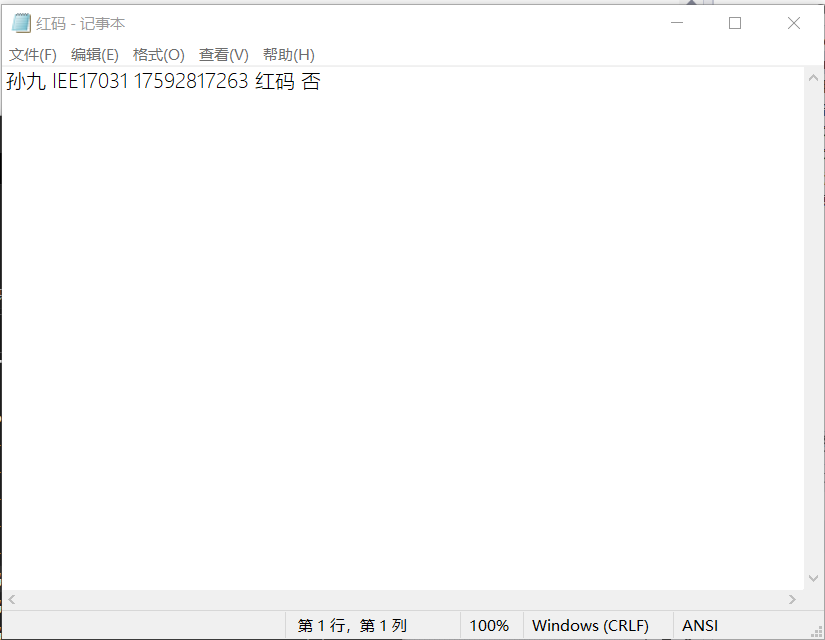


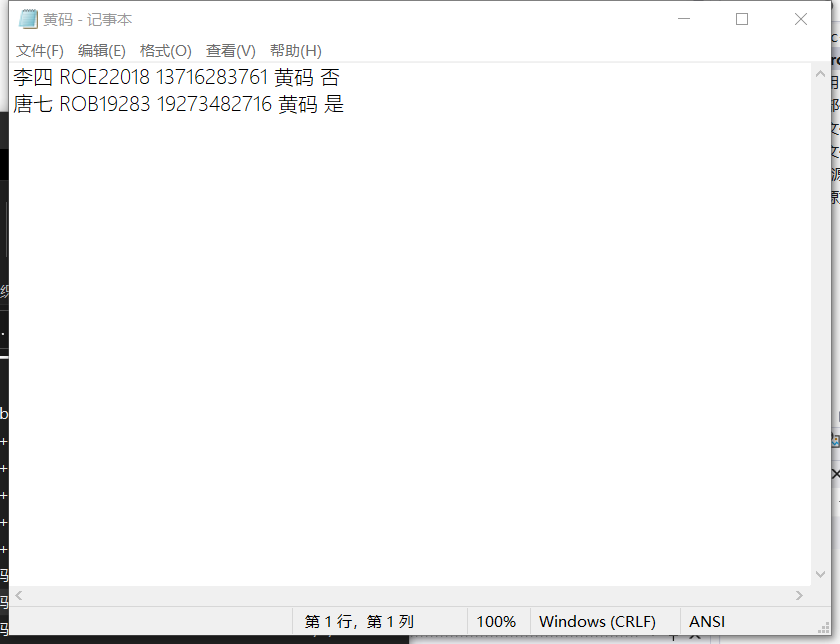
输入1 张三

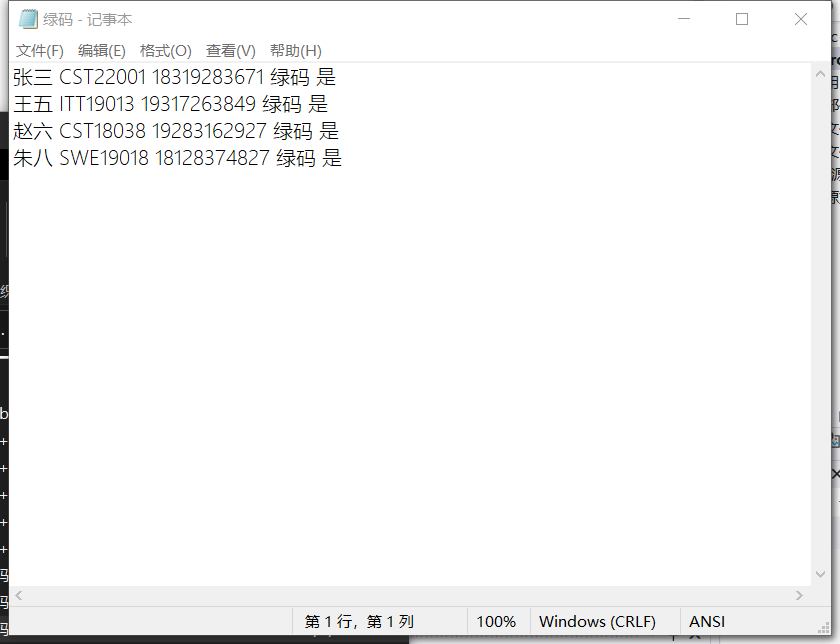


输入2



红码文件：

黄码文件：

绿码文件：

## 设计体会

写完成就感满满，真棒！

# 二、实现一个文本处理类TXTManager

## 完整代码

#include<bits/stdc++.h>

using namespace std;

typedef long long int ll;

ll cnt\_all = 0;

ll cnt\_s = 0;

ll cnt\_ASCII = 0;

ll cnt\_hang = 0;

ll cnt\_word = 0;

ll cnt\_words = 0;

class TXTManager

{

public:

//1,文本字符数(ASCII)的统计

void getASCII()

{

ifstream fin("youth.txt");

if (!fin)exit(1);

char a;

while (fin >> a)

{

if ((int)(a) >= 0 && (int)(a) <= 127)cnt\_ASCII++;

}

cout << "文本字符数(ASCII):" << cnt\_ASCII << endl;

fin.close();

}

//2,行数统计

void getHang()

{

ifstream fin("youth.txt");

if (!fin)exit(1);

string c;

while (fin.peek() != EOF)

{

getline(fin, c);

cnt\_hang++;

}

cout << "行数:" << cnt\_hang << endl;

fin.close();

}

//3,单词个数统计

void getword()

{

ifstream fin("youth.txt");

if (!fin)exit(1);

char w[12000];

while (fin.peek() != EOF)

{

fin.getline(w, sizeof(w), '\n');

//fin.getline(w, sizeof(w), ' ');

cnt\_words+=getwords(w);

}

cout << "单词个数:"<<cnt\_words << endl;

fin.close();

}

//4,某个字符的出现次数(区分大小写)

void searchChar()

{

ifstream fin("youth.txt");

if (!fin)exit(1);

char s;

cout << "请输入想要查询的字符：" << endl;

cin >> s;

char a;

while (fin >> a)

{

if (a == s)cnt\_s++;

//cout << a;

}

cout << s << "字符出现的次数:" << cnt\_s << endl;

fin.close();

}

//5,某个单词的出现次数(不区分大小写)

void searchWord()

{

ifstream fin("youth.txt");

if (!fin)exit(1);

char w[1025];

char y[1025];

int i = 0;

cout << "请输入想查询单词：" << endl;

cin >> y;

for (int i = 0; i < getstrlen(y); i++)

{

y[i] = tolower(y[i]);

}

while (fin.peek() != EOF)

{

fin.getline(w, sizeof(w), '\n');

cnt\_word+=func(w,y);

}

cout << y << "单词出现的次数:" << cnt\_word << endl;

fin.close();

}

//判断是否相同

bool judge(string w, char\* y)

{

for (int i = 0; i < w.length(); i++)

{

if (w[i] != y[i])return false;

}

return true;

}

//计算每行有多少个单词出现

int func(char\* x,char\*y)

{

int cnt = 0;

string s;

for (int i = 0; i < 1024; i++)

{

x[i] = tolower(x[i]);

if (x[i] >= 'a' && x[i] <= 'z')

{

s += x[i];

}

else if(x[i]==' '||x[i]=='\n')

{

if (judge(s, y))cnt++;

s.erase();

}

if (x[i] < 0)return cnt;

}

}

//计算字符数组中有效字符数

int getstrlen(char\*x)

{

int cnt = 0;

for (int i = 0; i < 1024; i++)

{

cnt++;

if (x[i] == '\0')return cnt;

if (x[i] == '\n')return cnt;

}

}

//计算每行有多少个单词

int getwords(char\*s)

{

int cnts = 0;

for (int i = 0; i < 12000; i++)

{

if (s[i] == ' ')cnts++;

if (s[i] == '\0')

{

cnts++;

return cnts;

}

}

}

};

int main()

{

TXTManager t;

int num;

cout << "功能：" << endl;

cout << "(1)文本字符数(ASCII)的统计" << endl;

cout << "(2)行数统计" << endl;

cout << "(3)单词个数统计" << endl;

cout << "(4)某个字符的出现次数(区分大小写)" << endl;

cout << "(5)某个单词的出现次数(不区分大小写)" << endl;

cin >> num;

if (num == 1)t.getASCII();

if (num == 2)t.getHang();

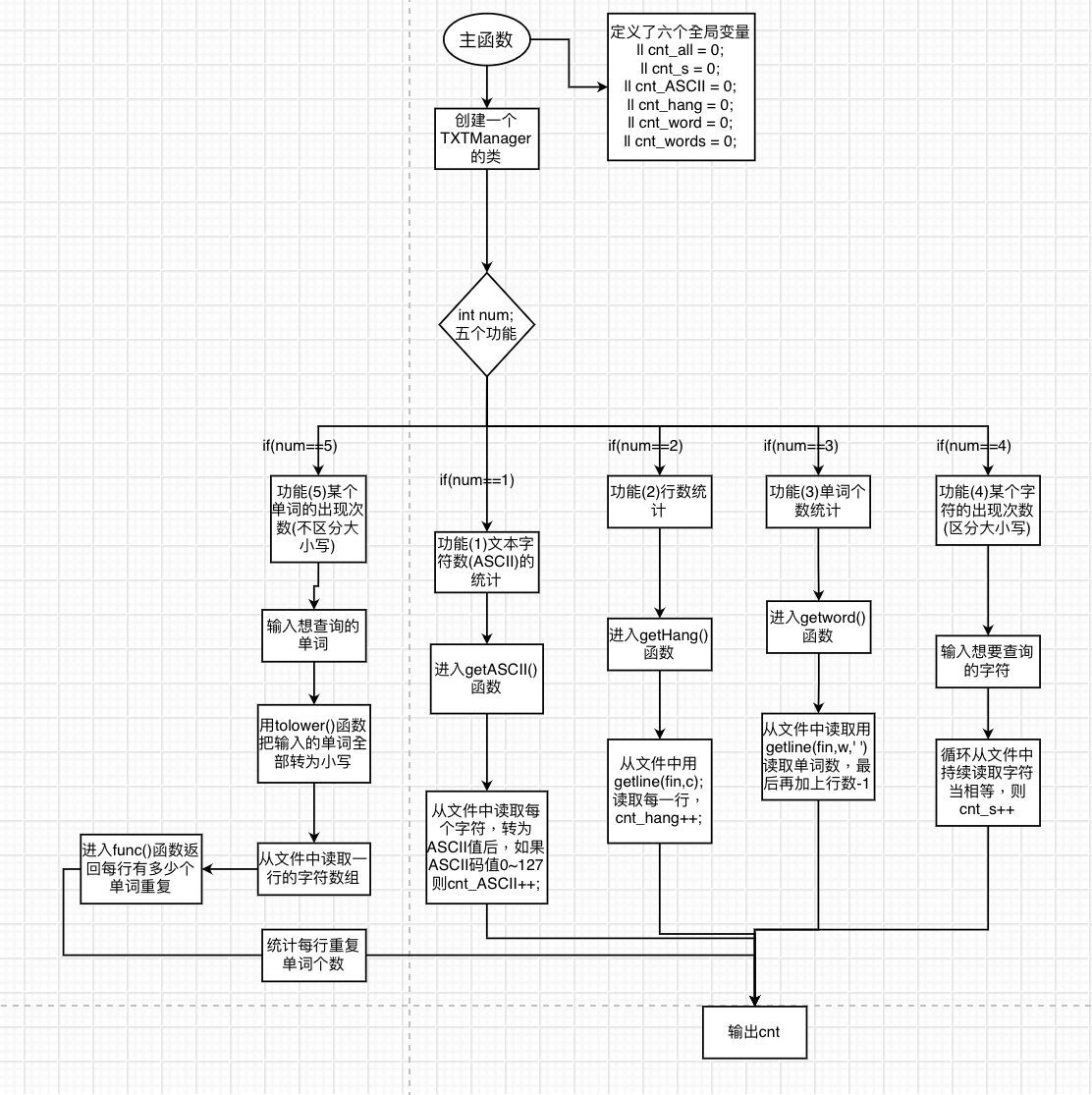
if (num == 3)t.getword();

if (num == 4)t.searchChar();

if (num == 5)t.searchWord();

}

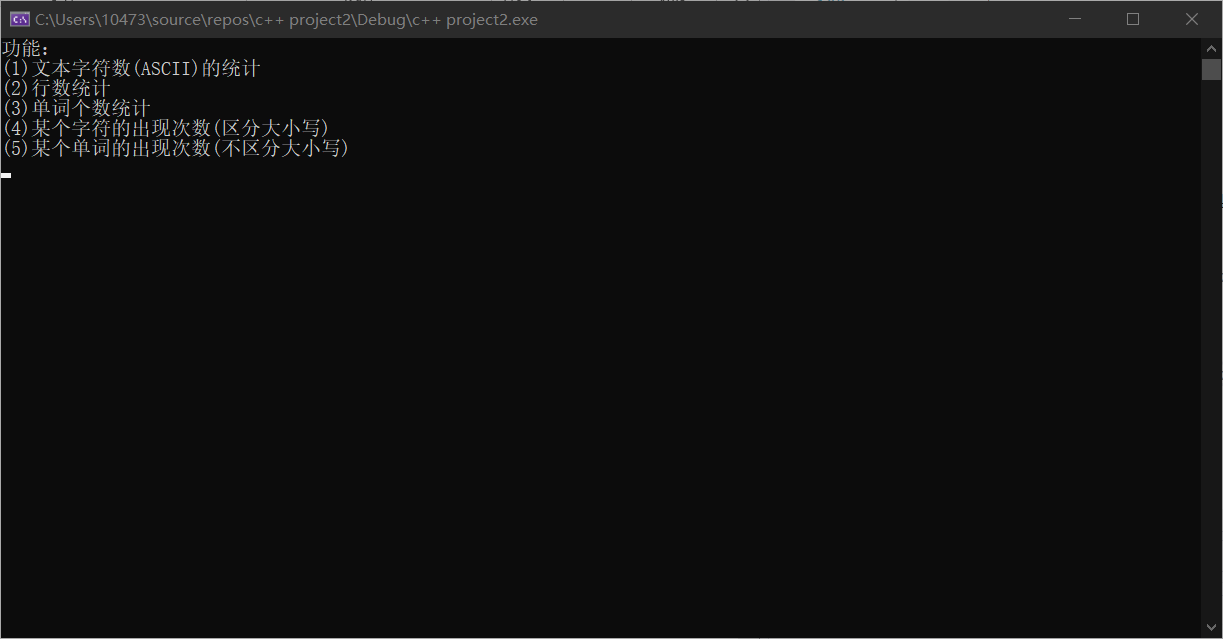
## 流程设计



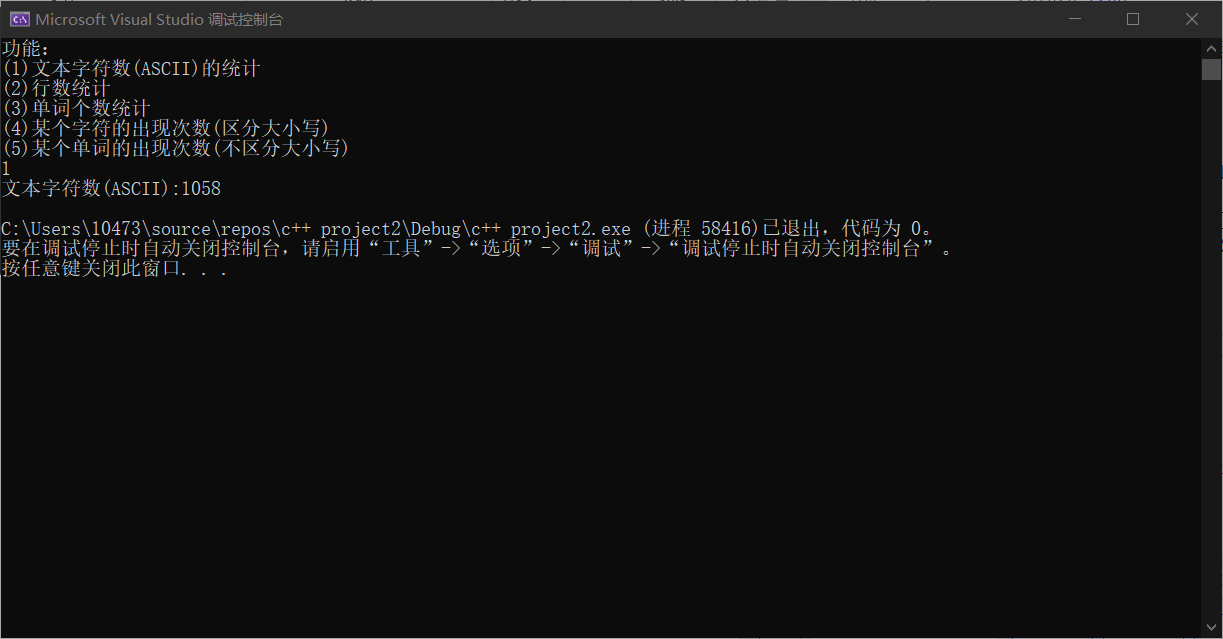
## 核心实现原理

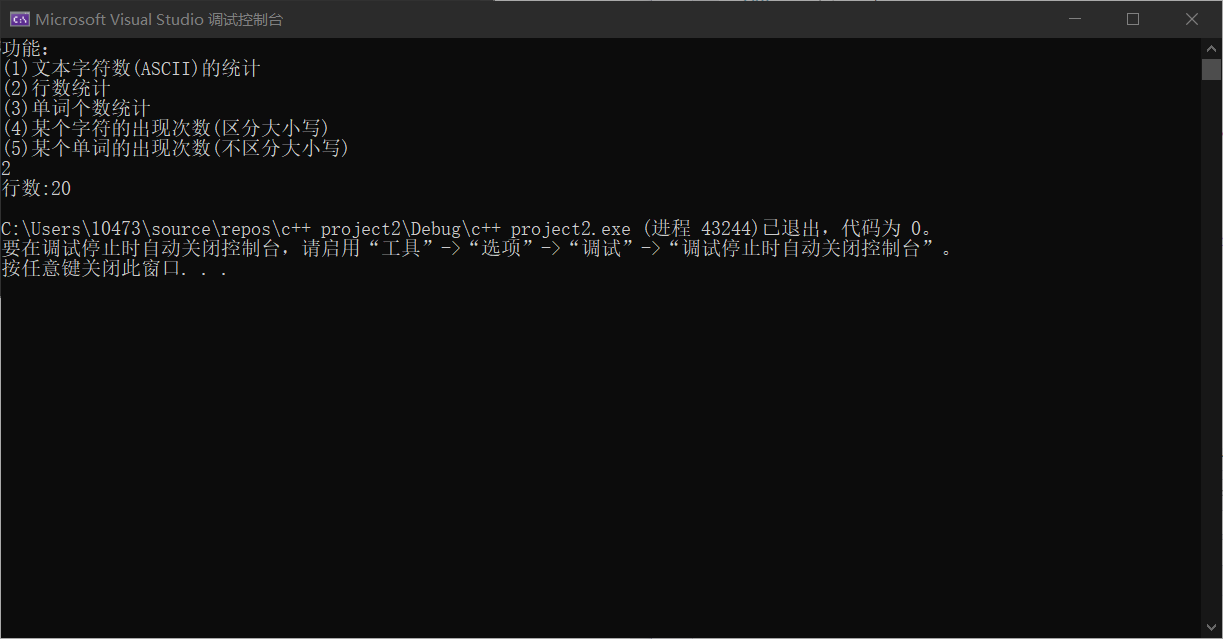
1. 功能(1)文本字符数(ASCII)的统计:用while(fin>>a)读取一个字符,判断每个字符是否在ASCII范围内;
2. 功能(2)行数统计:用while(fin.peek()!=EOF)退出循环,每次循环读取一行,cnt\_hang++;
3. 功能(3)单词个数统计:仍是while(fin.peek()!=EOF)退出循环,每次循环读取一行,进入getwords(char\*)函数获取每一行的单词个数，每个字符判断只要是’ ’cnt++;或者’\0’就cnt++;且返回cnt值;
4. 功能(4)某个字符的出现次数:从文件读取每一个字符与输入字符对比，相同就cnt\_s++;
5. 功能(5)某个单词的出现次数:首先是从文件中读取一行的字符;进入func(char\*,char\*)函数返回每行符合输入单词的个数;进入循环，先把传入一行的字符用tolower()函数转成小写，如果改字符为’a’~’z’则存入字符串中，若读取行字符为’ ’||’\n’就进入judge(string,char\*)判断文件这一行所得单词是否跟输入单词一样，是返回true，cnt++;否则返回false；然后用erase()函数清空字符数。经过遍历得出当一行有效字符读取完后会出现负数，因此当读取字符为负数是返回cnt的值。

## 结果截图及说明

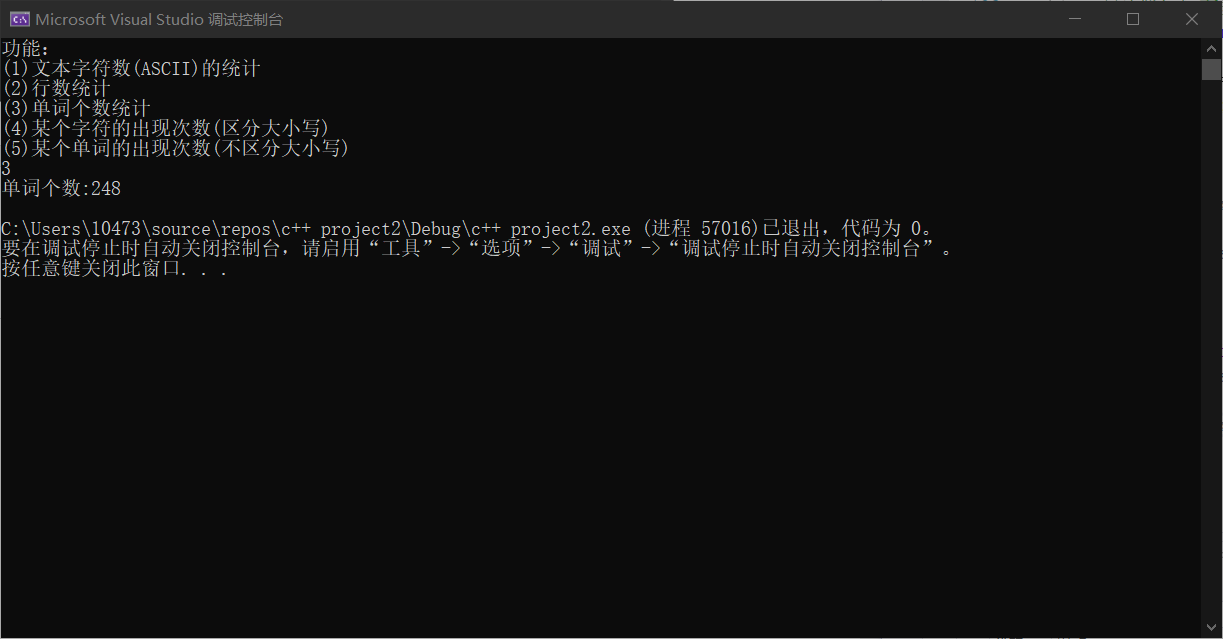


输入1：

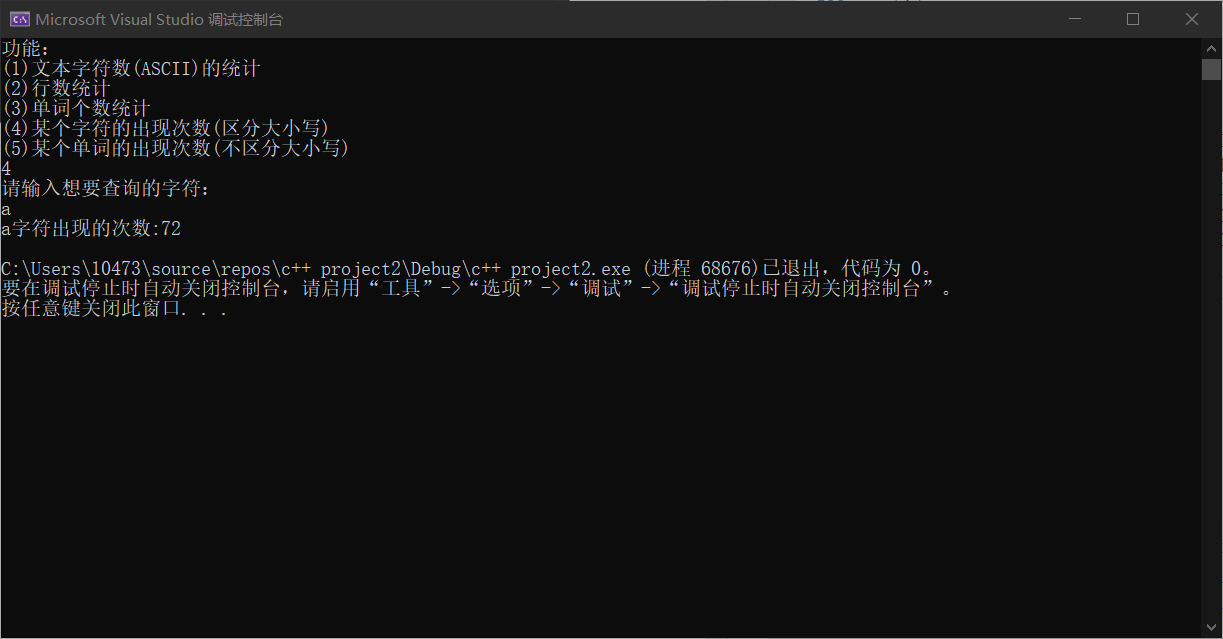


输入2：

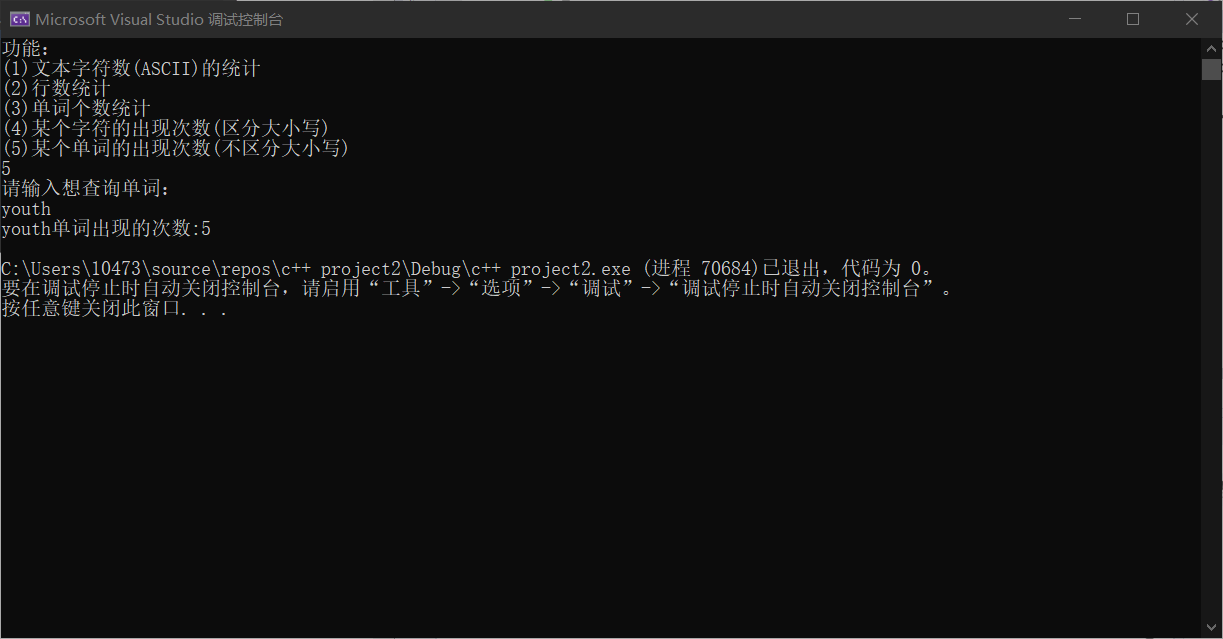
输入3：



输入4：a



输入5：youth



## 设计体会

很考验思维能力，运用多个函数，指针实现代码。

# 三、图像处理

## 完整代码

#include<bits/stdc++.h>

using namespace std;

class ImgProc

{

public:

int num;

void Gray()

{

ifstream fin("kiki.bmp", ios::binary);

ofstream fout("kiki\_2.bmp", ios::binary);

if (!fin || !fout)exit(1);

unsigned char buf[1024] = { 0 };

fin.read((char\*)buf, 54);

fout.write((char\*)buf, 54);

unsigned int w = 700, h = 700;

for (int i = 0; i < h; i++)

{

for (int j = 0; j < w; j++)

{

//Gray = R\*0.299 + G\*0.587 + B\*0.114

fin.read((char\*)buf, 3);

unsigned char t;

t = buf[0] \* 0.114 + buf[1] \* 0.587 + buf[2] \* 0.299;

buf[0] = buf[1] = buf[2] = t;

fout.write((char\*)buf, 3);

}

}

fin.close();

fout.close();

}

void masaike()

{

ifstream fin("kiki.bmp", ios::binary);

ofstream fout("kiki\_2.bmp", ios::binary);

if (!fin || !fout)exit(1);

unsigned char buf[1024] = { 0 };

fin.read((char\*)buf, 54);

fout.write((char\*)buf, 54);

unsigned int w = 700, h = 700;

unsigned char color[4900][3];//148

int m = 0;

int a, b;

for (int i = 0; i < h; i ++)

{

if (i % 10 == 0)

{

for (int j = 0; j < w; j++)

{

// fout.write((char\*)color, 3);

if(j%10==0)

{

a = i / 10 \* 3;

b = j / 10 ;

m = a \* 21 + b ;

fin.read((char\*)buf, 3);

color[m][0] = buf[0];

color[m][1] = buf[1];

color[m][2] = buf[2];

}

else

{

fin.read((char\*)buf, 3);

}

}

}

else

{

for (int j = 0; j < w; j++)

{

fin.read((char\*)buf, 3);

}

}

}

//fin.seekg(54, ios::cur);

for (int i = 0; i < h; i++)

{

for (int j = 0; j < w; j++)

{

//i/10\*3

int c, d, m;

c = i / 10 \* 3;

d = j / 10 ;

m = c \* 21 + d;

fin.read((char\*)buf, 3);

buf[0] = color[m][0];

buf[1] = color[m][1];

buf[2] = color[m][2];

fout.write((char\*)buf, 3);

}

}

fin.close();

fout.close();

}

void koutu()

{

ifstream fin("lotus.bmp", ios::binary);

ofstream fout("lotus\_out.bmp", ios::binary);

unsigned char buf[1024] = { 0 };

fin.read((char\*)buf, 54);

fout.write((char\*)buf, 54);

unsigned int w = 1200, h = 800;

for (int i = 0; i < h; i++)

{

for (int j = 0; j < w; j++)

{

fin.read((char\*)buf, 3);

//p过了

/\*if (buf[0] < 230 && buf[1]>130 && buf[2] > 100 && buf[2] < 220)

{

buf[0] = buf[1] = buf[2] = 255;

fout.write((char\*)buf, 3);

}\*/

//完美

if (buf[1] > buf[2] && buf[1] > buf[0])

{

buf[0] = buf[1] = buf[2] = 255;

fout.write((char\*)buf, 3);

}

else if (buf[2] > 100)

{

fout.write((char\*)buf, 3);

}

else if (buf[2] > buf[1] && buf[2] > buf[0] && buf[2] > 60)

{

fout.write((char\*)buf, 3);

}

else if (buf[2] > 100 && buf[0] > 100)

{

fout.write((char\*)buf, 3);

}

else if (buf[1] > 150 && buf[2] > 100)

{

fout.write((char\*)buf, 3);

}

else

{

buf[0] = buf[1] = buf[2] = 255;

fout.write((char\*)buf, 3);

}

}

}

fin.close();

fout.close();

}

};

int main()

{

ImgProc i;

cout << "请输入想实现功能序号：" << endl;

cout << "(1)图像灰度处理" << endl;

cout << "(2)图像马赛克处理" << endl;

cout << "(3)图像抠图处理" << endl;

int a;

cin >> a;

if (a == 1)i.Gray();

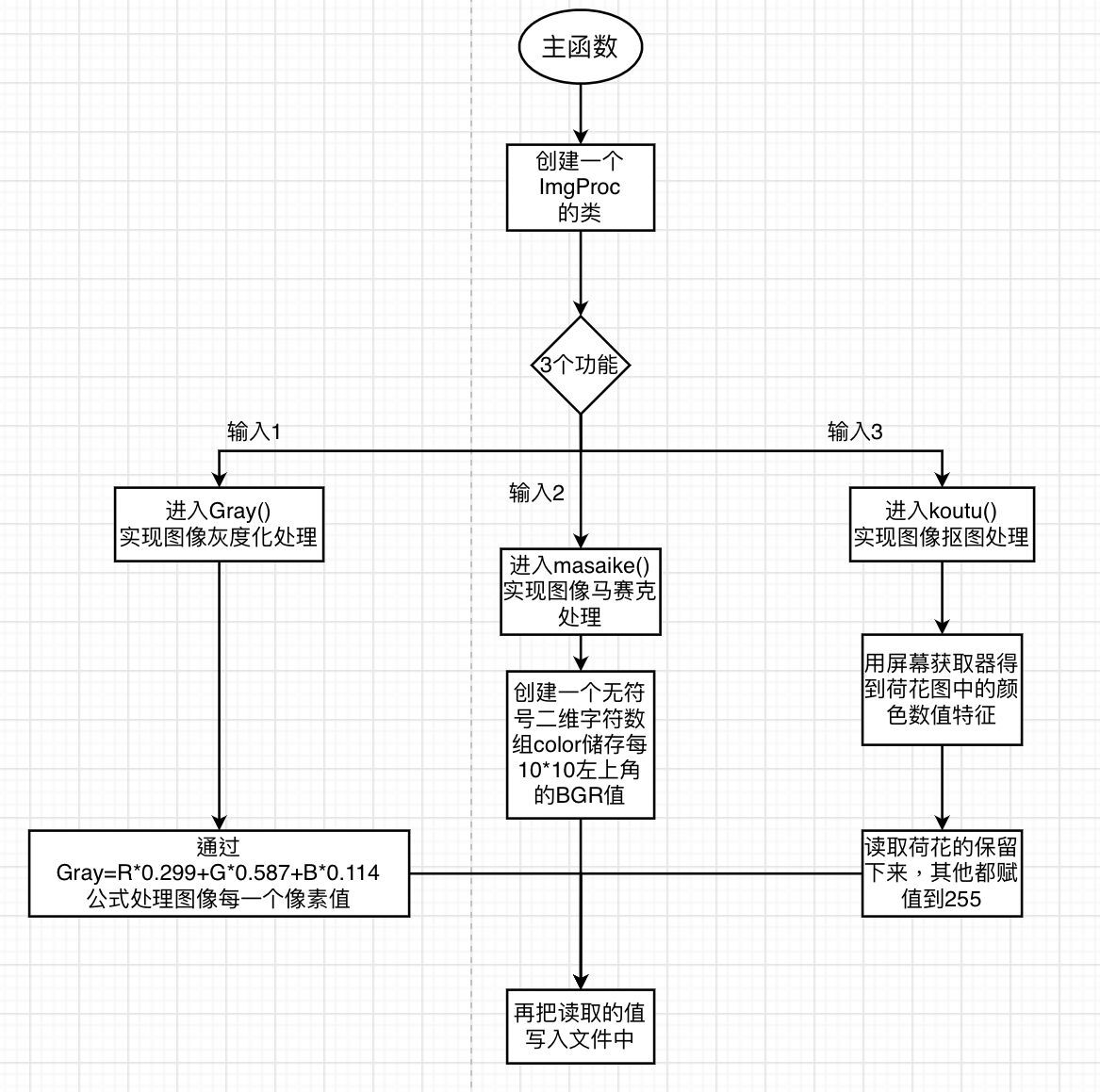
if (a == 2)i.masaike();

if (a == 3)i.koutu();

return 0;

}

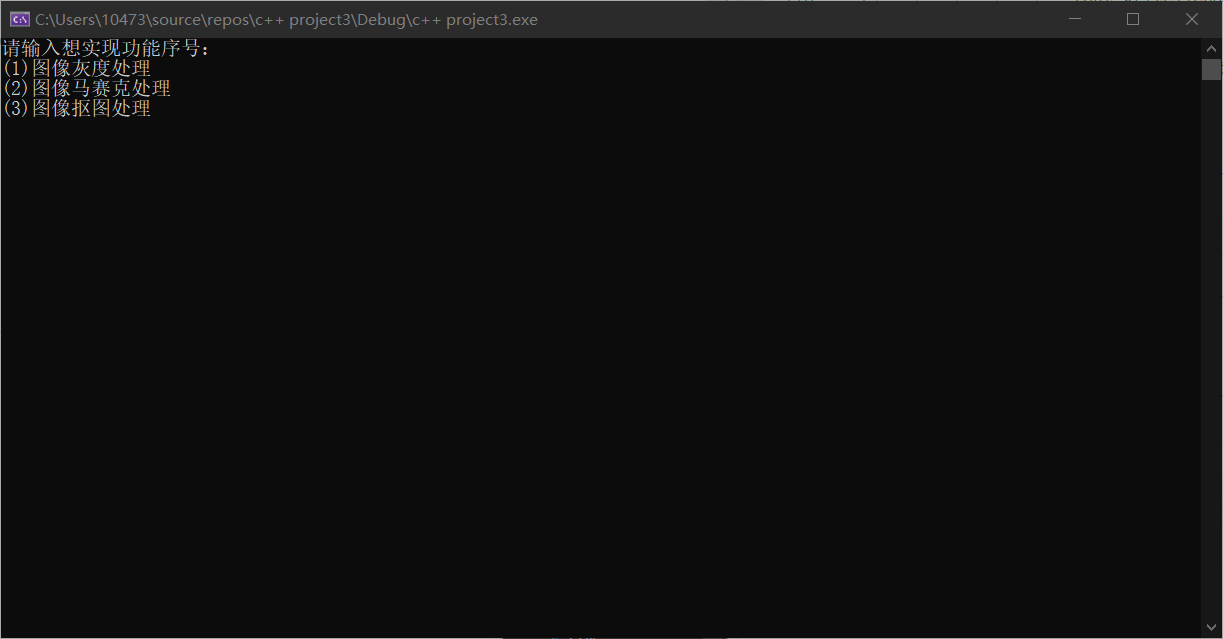
## 流程设计



## 核心实现原理

1. 功能(1)图像灰度处理：首先把前54位文件头读写入文件，进入循环，每次读取一个像素点的BGR三个值，通过Gray=R\*0.299 + G\*0.587 + B\*0.114公式处理后，写入文件中
2. 功能(2)图像马赛克处理：首先把前54位文件头读写入文件，利用无符号二位字符数组储存每10\*10的矩阵左上角的像素值，通过计算该图像w\*h=700\*700;因此通过循环，如果行数i%10==0,则进入列的循环，如果j%10==0；则通过计算得出以下公式a=i/10\*3;b=j/10;m=a\*21+b;则计算出的m储存到color[m][0]…中；其余就向后每次移动3位指针；最后再通过公式写入文件中c = i / 10 \* 3;d = j / 10 ;m = c \* 21 + d;
3. 功能(3)图像抠图处理：首先把前54位文件头读写入文件，通过屏幕颜色获取可知在荷花位置R的值都比G和B的值大且绿叶深色区域总体值会比较小且G的值比B.R大，根据这些判断深色就把BGR三个值改为255，荷花区域直接写入，最后实现抠图。

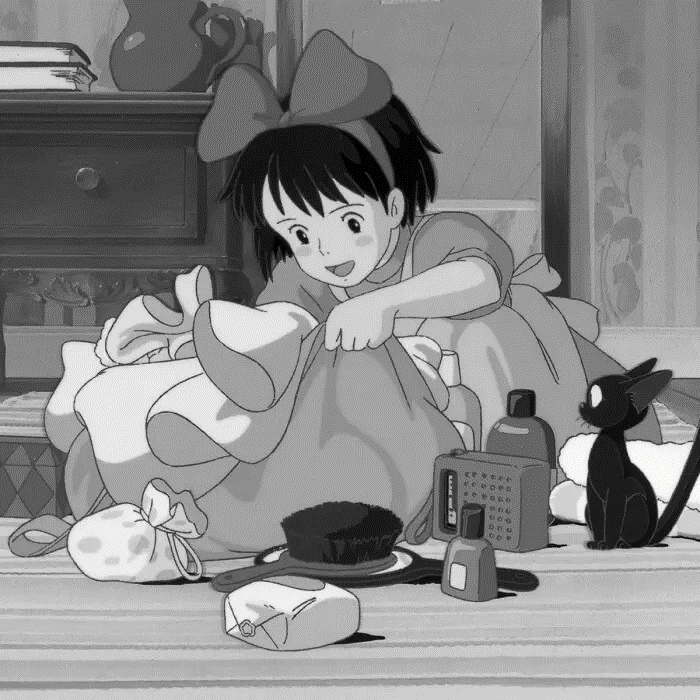
## 结果截图及说明



原图：



输入1：//功能(1)灰度处理：



输入2：马赛克处理



原图：



输入3：//抠图



## 设计体会

公式的总结，让我有一种很高能的感觉！很酷！