§1.1题目描述

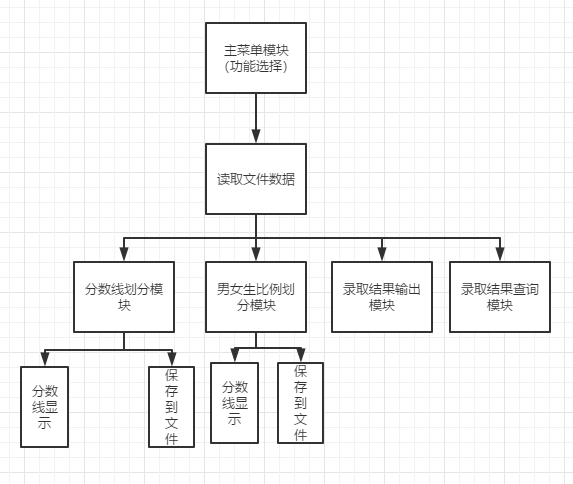
设计一个高考招生录取程序。划定高考各批次录取分数线，结果用屏幕显示，并且输出到文件中。统计各批次中男女生比例，结果用屏幕显示并输出到文件中。

确定每名学生的录取情况和录取批次；根据考生考号或姓名可以查询录取情况和成绩。

§1.2算法设计

设计思想：该系统应该包含的信息包括学生的考号、姓名、性别和考试成绩，将这些数据编成结构体。第一个功能的实现只需对一千个结构体中成绩高低进行排序，然后按各批次录取比例提取分数，作为各个批次的录取分数线。第二个功能的实现只需对排序后的一千个结构体中各批次的男女生人数进行统计，第三个功能的实现只需对每个数据与录取分数线相比较，属于哪个分数线被录取为哪个批次。第四个功能的实现需要利用字符串比较函数，确定并显示要查询的学生全部信息。

功能模块图：



§1.3程序设计及代码分析

int main() {

int n, m;

char c;

m = 1;

while (m) {

puts("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

puts("\t1.查询录取分数线；2.查询各批次男女比例");

puts("\t3.查看各批次录取情况：4.输入考生考号或姓名查询录取情况\*\*\*\*");

puts("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

printf("输入选择：");

scanf("%d", &n);

getchar();

switch (n) {

case 1:

Function1();

break;

case 2:

Function2();

break;

case 3:

Function3();

break;

case 4:

Function4();

break;

printf("继续选择Y，退出选择N");

scanf("%c", &c);

if (c == 'y' || c == 'Y') {

m = 1;

}

else m = 0;

printf("\n");

}

}return 0;

}

主函数使用while循环对主菜单复用，switch来接受用户选择的功能命令。

void Function1() {

//struct student stu[1000];

int i=0, j, k;

struct student t;

FILE \*fp= fopen("f1.txt","r");

if (fp == NULL) {

printf("open error!\n");

return 0;

}

for (k = 0; k < 1000; k++) {

fscanf(fp,"%d%s%s%d",&stu[k].num,stu[k].name,stu[k].sex,&stu[k].score);

}

fclose(fp);

for (j = 0; j < 999; j++) {

for (i = 0; i < 999 - j; i++) {

if (stu[i].score > stu[i + 1].score) {

t = stu[i];

stu[i] = stu[i + 1];

stu[i + 1] = t;

}

}

}

printf("一类本科分数线：\n");

printf("%d\n", stu[899].score);

printf("二类本科分数线：\n");

printf("%d\n", stu[749].score);

printf("三类本科分数线：\n");

printf("%d\n", stu[549].score);

printf("四类高职高专分数线：\n");

printf("%d\n", stu[399].score);

FILE \*out;

out = fopen("f2.txt", "w");

if (out == NULL)

{

printf("Can't open the file!\n");

}

fprintf(out, "一类本科分数线：\n%d\n", stu[899].score);

fprintf(out, "二类本科分数线：\n%d\n", stu[749].score);

fprintf(out, "三类本科分数线：\n%d\n", stu[459].score);

fprintf(out, "四类高职高专分数线：\n%d\n", stu[399].score);

fclose(out);

printf("信息已输出到文档中\n");

}

第一个功能的实现只需对一千个结构体中成绩高低进行排序，然后按各批次录取比例提取分数，作为各个批次的录取分数线。

void Function2() {

FILE \*fp;

fp = fopen("f1.txt", "r");

struct student t;

int i, j, m;

for (i = 0; i < 1000; i++) {

fscanf(fp,"%d%s%s%d", &stu[i].num, stu[i].name, stu[i].sex, &stu[i].score);

}

for (j = 0; j < 999; j++) {

for (i = 0; i < 999 - j; i++) {

if (stu[i].score > stu[i].score) {

t = stu[i];

stu[i] = stu[i + 1];

stu[i + 1] = t;

}

}

}

int a, b, c, d, e, f, g, h;

a = b = c = d = e = f = g = h = 0;

for (m = 899; m <= 999; m++) {

if (strcmp(stu[m].sex, "男") == 0)

a++;

}

b = 100 - a;

FILE \*out;

out = fopen("f2.txt", "w");

if (out == NULL)

{

printf("Can't open the file!\n");

}

fprintf(out, "一类本科男生：女生：\n%d %d\n", a, b);

printf("一类本科男生：女生：\n");

printf("%d %d\n", a, b);

for (m = 749; m < 899;m++) {

if (strcmp(stu[m].sex, "男") == 0)

c++;

}

d = 150 - c;

fprintf(out, "二类本科男生：女生：\n%d %d\n", c, d);

printf("二类本科男生：女生：\n");

printf("%d %d\n", c, d);

for (m = 549; m < 749; m++) {

if (strcmp(stu[m].sex, "男") == 0)

e++;

}

f = 200 - e;

fprintf(out, "三类本科男生：女生：\n%d %d\n",e, f);

printf("三类本科男生：女生：\n");

printf("%d %d\n", e, f);

for (m = 399; m < 549; m++) {

if (strcmp(stu[m].sex, "男") == 0)

g++;

}

h = 200 - g;

fprintf(out, "高职高专男生：女生：\n%d %d\n", g, h);

printf("高职高专男生：女生：\n");

printf("%d %d\n", g, h);

fclose(out);

printf("信息已输出到文档中\n");

}

第二个功能的实现只需对排序后的一千个结构体中各批次的男女生人数进行统计，然后输出到屏幕和文档。

void Function3() {

FILE \*fp;

fp = fopen("f1.txt", "r");

struct student t;

int i, j;

for (i = 0; i < 1000; i++) {

fscanf(fp, "%d%s%s%d", &stu[i].num, stu[i].name, stu[i].sex, &stu[i].score);

}

for (j = 0; j < 999; j++) {

for (i = 0; i < 999 - j; i++) {

if (stu[i].score > stu[i].score) {

t = stu[i];

stu[i] = stu[i + 1];

stu[i + 1] = t;

}

}

}

for (i = 0; i < 1000; i++) {

if (stu[i].score > stu[899].score)

printf("%s被本一学校录取\n", stu[i].name);

if(stu[i].score>stu[749].score&&stu[i].score<stu[899].score)

printf("%s被本二学校录取\n", stu[i].name);

if (stu[i].score > stu[549].score&&stu[i].score < stu[749].score)

printf("%s被本三学校录取\n", stu[i].name);

if (stu[i].score > stu[399].score&&stu[i].score < stu[549].score)

printf("%s被高职高专学校录取\n", stu[i].name);

else if (stu[i].score < stu[399].score)

printf("%s未被录取\n", stu[i].name);

}

}

第三个功能的实现只需对每个数据与录取分数线相比较，属于哪个分数线被录取为哪个批次。

void Function4() {

FILE \*fp;

fp = fopen("f1.txt", "r");

struct student t;

int i, j, h,k;

for (i = 0; i < 1000; i++) {

fscanf(fp, "%d%s%s%d", &stu[i].num, stu[i].name, stu[i].sex, &stu[i].score);

}

for (j = 0; j < 999; j++) {

for (i = 0; i < 999 - j; i++) {

if (stu[i].score > stu[i].score) {

t = stu[i];

stu[i] = stu[i + 1];

stu[i + 1] = t;

}

}

}

char m[20];

puts("\*\*\*\*1.根据考号查询：2.根据姓名查询\*\*\*\*");

printf("输入选择：");

scanf("%d", &h);

getchar();

switch (h) {

case 1:{

printf("请输入考生考号");

scanf("%ld", &h);

for (i = 0; i < 1000; i++)

if (h == stu[i].num) {

printf("ld %s %s %d\n", stu[i].num, stu[i].name, stu[i].sex, stu[i].score);

if (stu[i].score > stu[899].score)

printf("%s被本一学校录取\n", stu[i].name);

if (stu[i].score > stu[749].score&&stu[i].score < stu[899].score)

printf("%s被本二学校录取\n", stu[i].name);

if (stu[i].score > stu[549].score&&stu[i].score < stu[749].score)

printf("%s被本三学校录取\n", stu[i].name);

if (stu[i].score > stu[399].score&&stu[i].score < stu[549].score)

printf("%s被高职高专学校录取\n", stu[i].name);

else if (stu[i].score < stu[399].score)

printf("%s未被录取\n", stu[i].name);

break;

}

if (h != stu[i].num)

printf("输入的考号有误\n");

}

break;

case 2: {

printf("输入考生姓名：");

scanf("%s", &m);

for(k=0;k<1000;k++)

if (strcmp(m, stu[k].name) == 0) {

printf("ld %s %s %d\n", stu[k].num, stu[k].name, stu[k].sex, stu[k].score);

if (stu[k].score > stu[899].score)

printf("%s被本一学校录取\n", stu[k].name);

if (stu[k].score > stu[749].score&&stu[k].score < stu[899].score)

printf("%s被本二学校录取\n", stu[k].name);

if (stu[k].score > stu[549].score&&stu[k].score < stu[749].score)

printf("%s被本三学校录取\n", stu[k].name);

if (stu[k].score > stu[399].score&&stu[k].score < stu[549].score)

printf("%s被高职高专学校录取\n", stu[k].name);

else if (stu[k].score < stu[399].score)

printf("%s未被录取\n", stu[k].name);

break;

}

if (strcmp(m,stu[k].name)!=0)

printf("输入的姓名有误\n");

break;

}

}

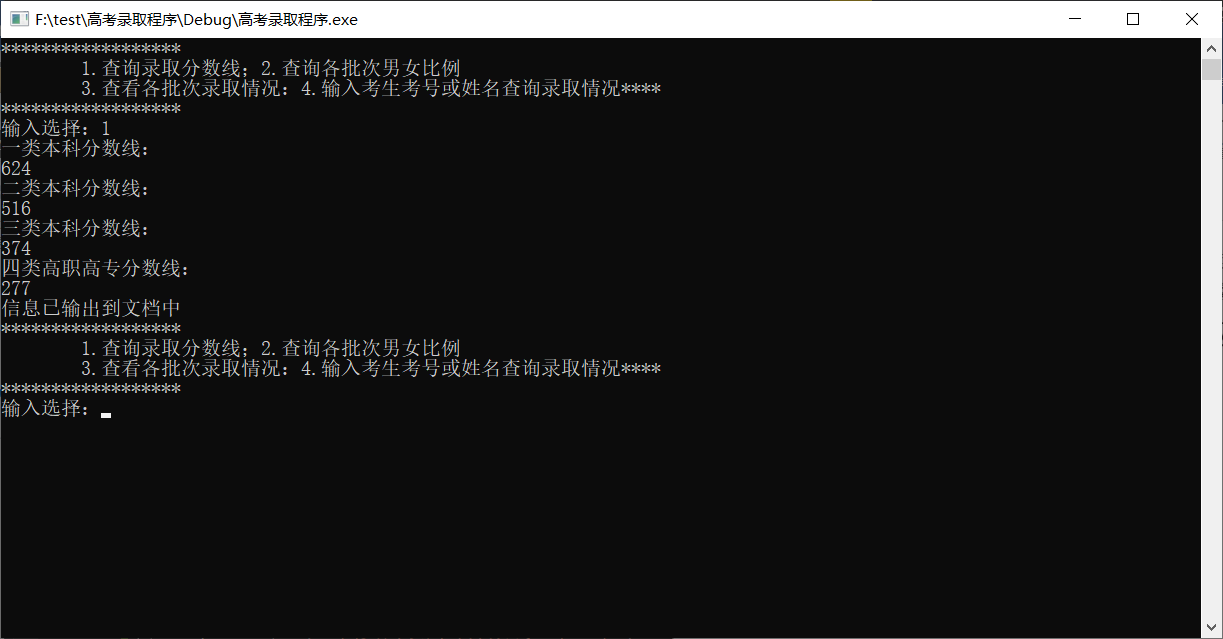
}

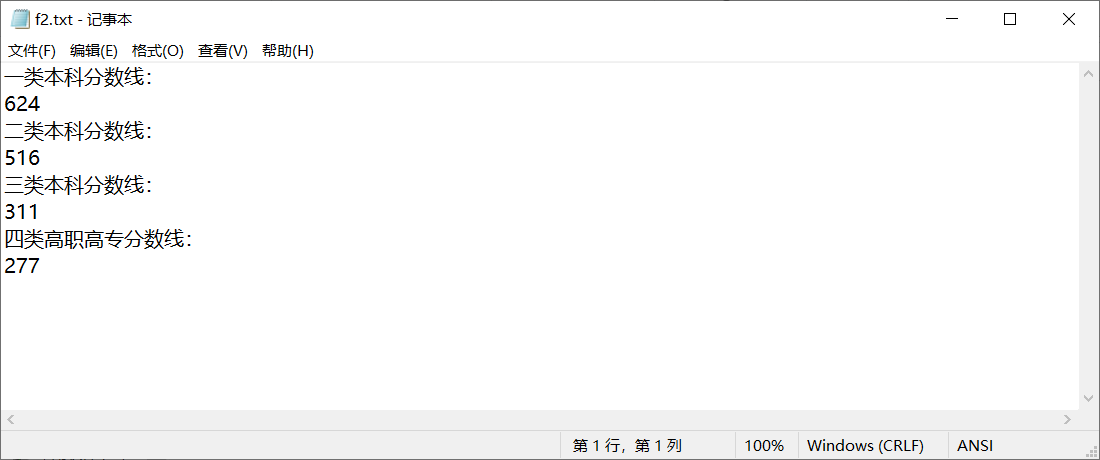
第四个功能的实现需要根据用户选择的功能，分别用学号和姓名对信息进行查找，其中姓名查询时需要利用字符串比较函数，确定并显示要查询的学生全部信息。

§1.4实验数据及运行效果截图

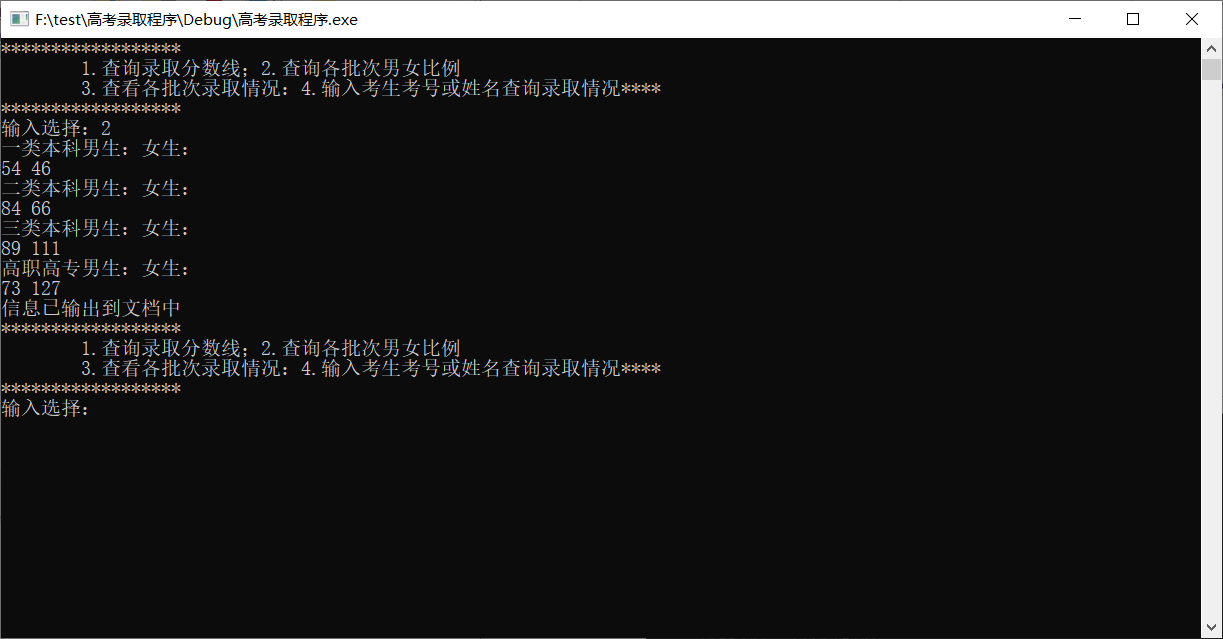
实验数据：f1.txt,f2.txt,t3.txt

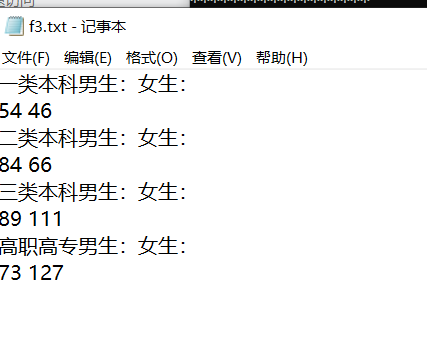
功能1：



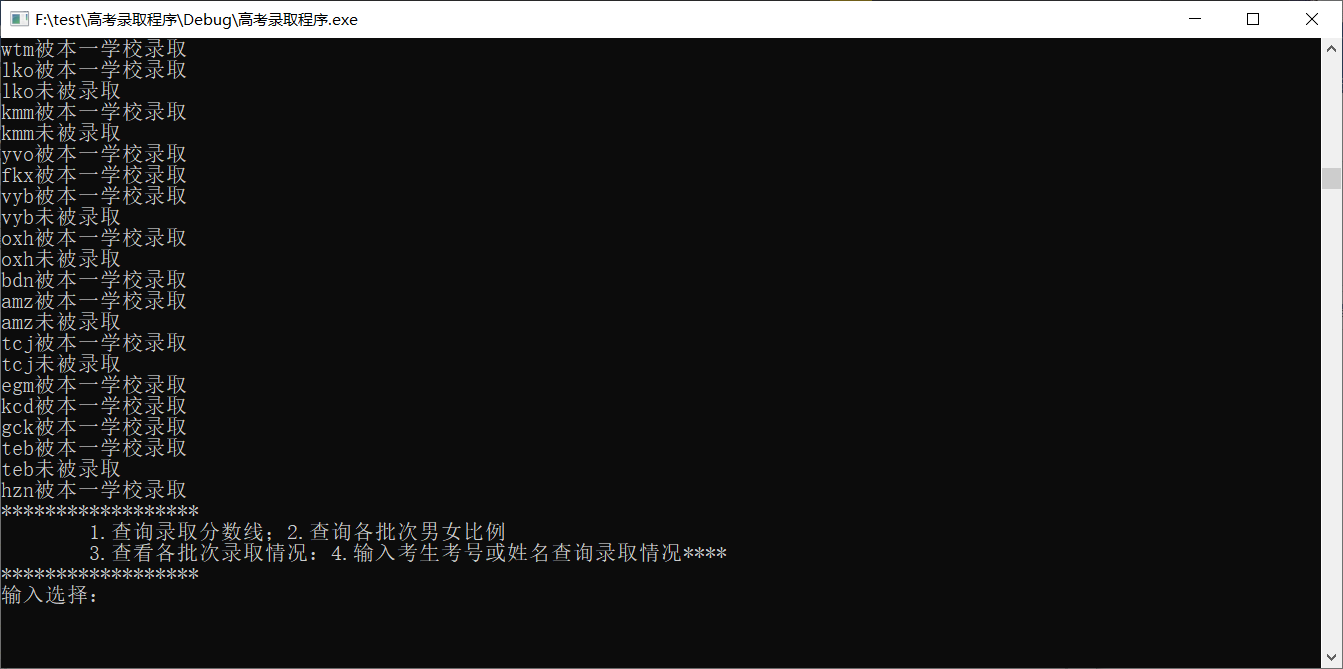


功能2：

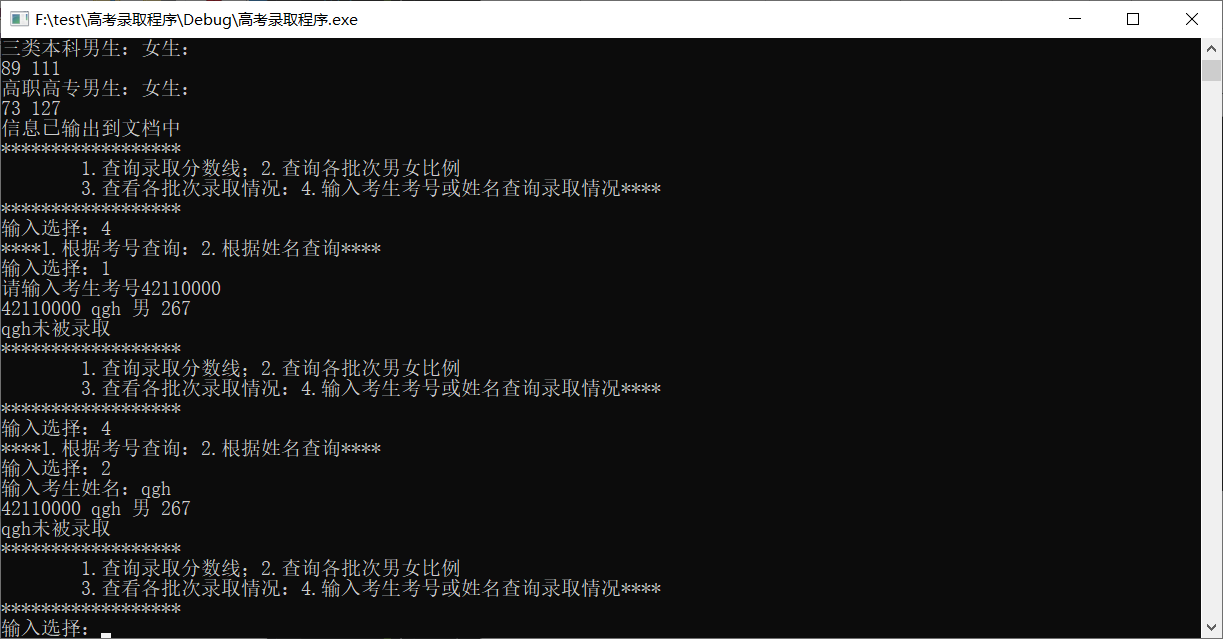




功能3：



功能4：



§1.5设计中出现的错误及解决方法

在对文件进行操作时，将fopen误写成了open，导致无法读取文件，经过单步调试发现问题所在，修改之后得以解决。