**华中科技大学计算机科学与技术学院**

**C语言课程设计报告**



题目: 学生班级管理信息系统

专 业：计算机科学与技术

班 级：CS1606

学 号：U201614683

姓 名：薛旭升

成 绩：

指导教师：王多强

**完成日期：** 2016年 9 月 27 日

目录

[一、系统需求分析 4](#_Toc16333)

[1.录入功能： 4](#_Toc2732)

[2.查询功能 4](#_Toc1339)

[3.插入功能 5](#_Toc7291)

[4.统计功能 5](#_Toc14496)

[二、总体设计 6](#_Toc6767)

[1.文件操作 6](#_Toc32380)

[2.数据查询 7](#_Toc29021)

[3.数据插入 8](#_Toc18771)

[4.数据统计 9](#_Toc18070)

[三、数据结构设计 10](#_Toc14990)

[四、详细设计 13](#_Toc27091)

[1.主函数 13](#_Toc3884)

[2.查询系统 13](#_Toc18329)

[3.插入系统 18](#_Toc7960)

[4.统计系统 18](#_Toc26554)

[5.录入功能 21](#_Toc27686)

[五、系统实现 22](#_Toc8410)

[六、运行测试与结果分析 23](#_Toc26116)

[七、总结 28](#_Toc12578)

[八、参考文献 29](#_Toc29345)

[九、指导教师评语 30](#_Toc11657)

[附录一 31](#_Toc1750)

[附录二 67](#_Toc1107)

# 一、系统需求分析

学生管理系统主要功能有四个：录入功能、查询功能（包括删除、修改功能）、插入功能、统计功能。

## 1.录入功能：

录入功能要求从三个不同文件中分别读出年级信息表、班级信息表、学生信息表，并建立一个三级十字交叉链表，将读出的三个信息表储存到该十字交叉链表中。年级信息表对应一级链表，班级信息表对应二级链表，学生信息对应三级链表。三个信息表分别包括以下信息：

1. 年级信息表：

年级编号、入学时间、入学人数、毕业人数、年级辅导员姓名、辅导员联系电话、年级学生会主席姓名；

1. 班级信息表：

班级编号、专业名称、入学人数、入学平均年龄、毕业人数、班长姓名、班长联系电话、班主任姓名、班主任联系电话；

1. 学生信息表：

学号、姓名、性别、籍贯、出生年月、联系电话、入学分数、毕业与否、毕业去向。

2.查询功能：

查询功能要求可以进行年级查询、班级查询、学生查询三种查询操作以及对查询结果可以进行删除、修改操作。

年级查询有3种查询方式（即可以输入3类数据）：年级编号、入学时间、入学人数。年级编号查询还要求可以模糊查询，入学时间、入学人数查询还要求可以区间查询。

班级查询有4种查询方式（即可以输入4类数据）：班级编号、专业、人数、班主任姓名。班级编号、专业、班主任姓名可以进行模糊查询，人数可进行区间查询。

学生查询有6种查询方式（即可以输入6类数据）：学生姓名、专业、入学时间、年龄、毕业与否、毕业去向。姓名、专业、毕业去向还要求可进行模糊查询，入学时间、年龄还要求可以进行区间查询。

除此以外，这三种查询要求可以组合查询。

对于查询结果，应将该条结果的所有信息呈列表形式、逐条输出。

删除操作须将对应记录的所有信息删除，若查询出的结果为多条，删除功能需支持批量删除。

修改操作即将对应记录的所有信息重新输入赋值。

3.插入功能：

插入功能包括插入一个新年级、插入一个新班级、插入一个新同学。在信息表中插入一条新信息即在对应链表中插入一个新结点，而且该功能要求插入的新纪录位置不固定，而是根据排序方式插入，一般按年级编号/班级编号/学号首字母顺序排列。

插入时最好输入新纪录的所有信息。

4.统计功能：

统计功能可统计指定期限内信息以及无期限信息，若无期限则统计所有记录中满足统计要求的记录个数。

统计功能也可分为年级统计、班级统计、学生统计：

1. 年级统计包括：年级班数、入学人数、毕业人数、未毕业人数，对于统计结果，可指定按年级编号、入学人数、未毕业人数排序输出；
2. 班级统计只统计未毕业人数，对于统计结果，须按人数从大到小排序输出；

3）学生统计包括：毕业后某企业工作学生数量、某时间段出生学生数量，对于企业统计应按人数从大到小排序输出，对于出生时间统计应按年级排序输出。

# 二、总体设计

整个程序大致分为四个部分：文件操作、数据查询、数据插入、数据统计，如图2.1：

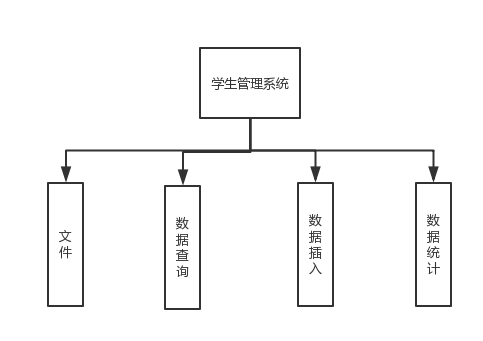


图2.1 程序总结构

1.文件操作：

文件包括创建、录入、保存功能，三个功能都先对第三级链表学生链表进行处理，由低到高，使年级链表的处理中运用班级链表的子函数，班级链表的处理中运用学生链表的子函数，最后创建、录入、保存功能的实现直接调用年级链表。

结构如图2.2：

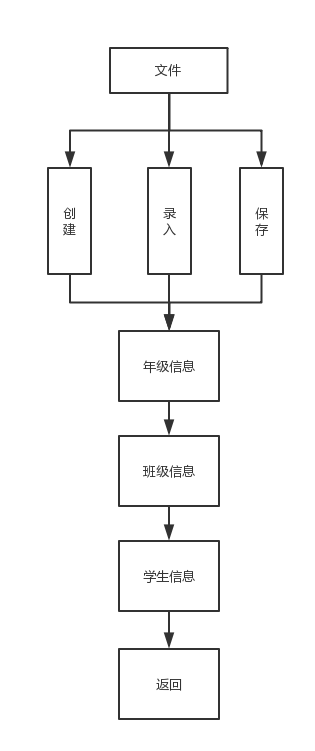


图2.2 文件操作结构图

2.数据查询:

查询功能要求可以进行年级查询、班级查询、学生查询三种查询操作以及对查询结果可以进行删除、修改操作。

年级查询有3种查询方式，班级查询有4种查询方式，学生查询有6种查询方式，除此以外，这三种查询要求可以组合查询。所以，对每种查询方式建立一个子函数，共16个子函数，在查询系统主函数中直接引用。

删除、修改功能各写三个函数，分别删除、修改年级信息、班级信息、学生信息。

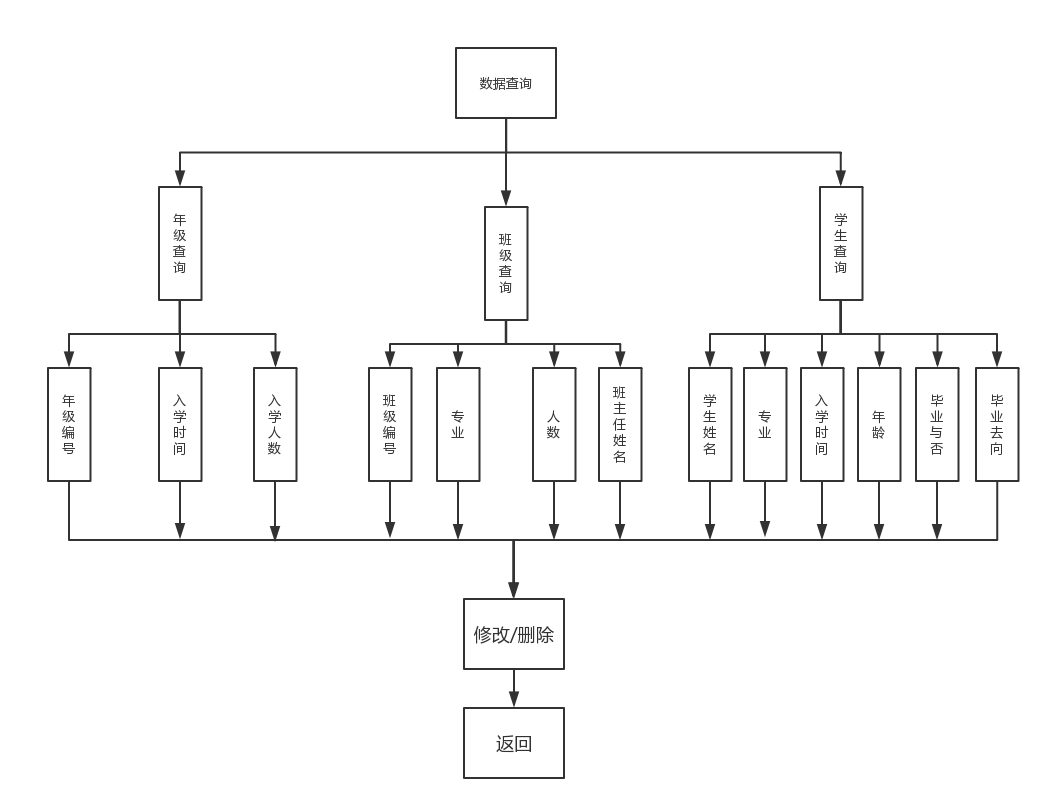


图2.3 数据查询的结构图

3.数据插入，如图2.4：

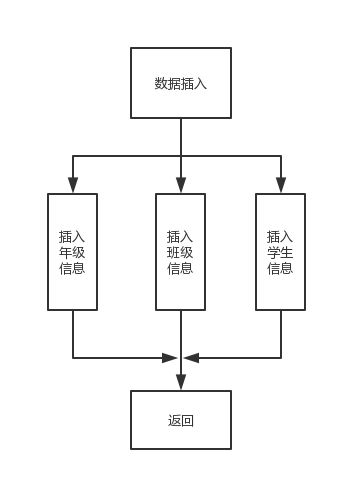


图2.4 数据插入结构图

4.数据统计：

数据统计也包括年级统计，班级统计，学生统计。每种统计又有不同的排序方式。

对于年级统计，应根据选的排序方式，先对年级链表排序，然后直接遍历年级链表统计数据并输出即可；对于班级统计与学生统计，应再声明其他结构，对满足条件的班级/学生建立链表，然后按照排序方式对新建立的链表排序并输出。

数据统计结构图，如图2.4：

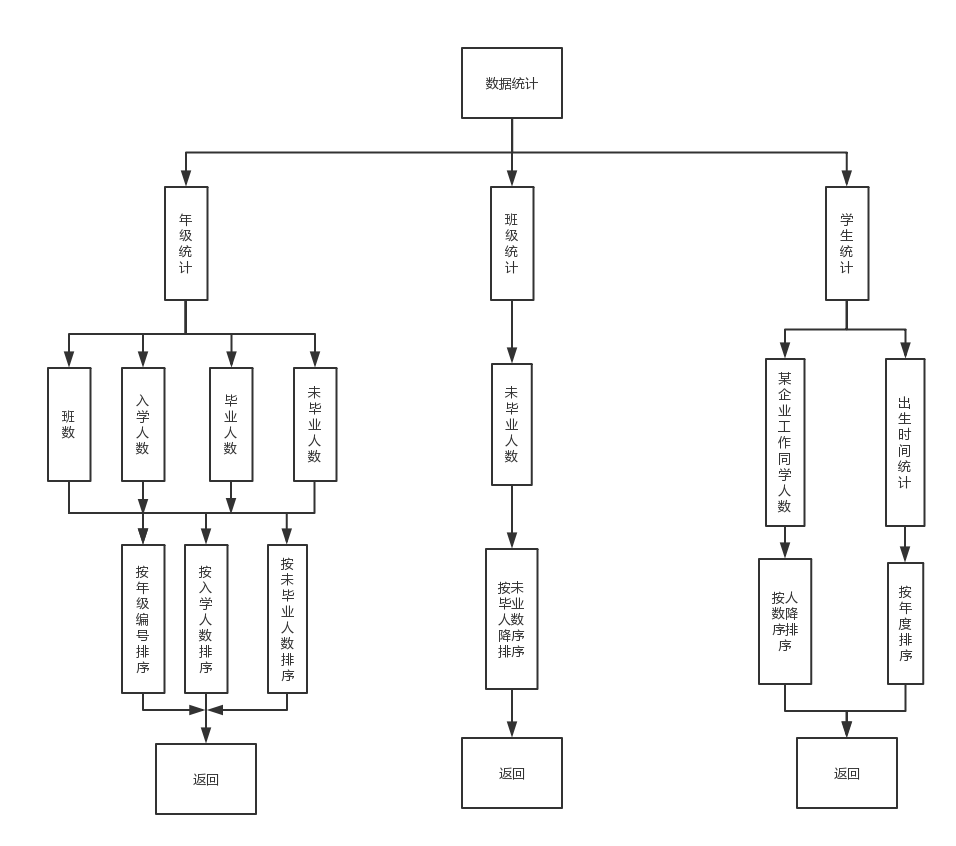
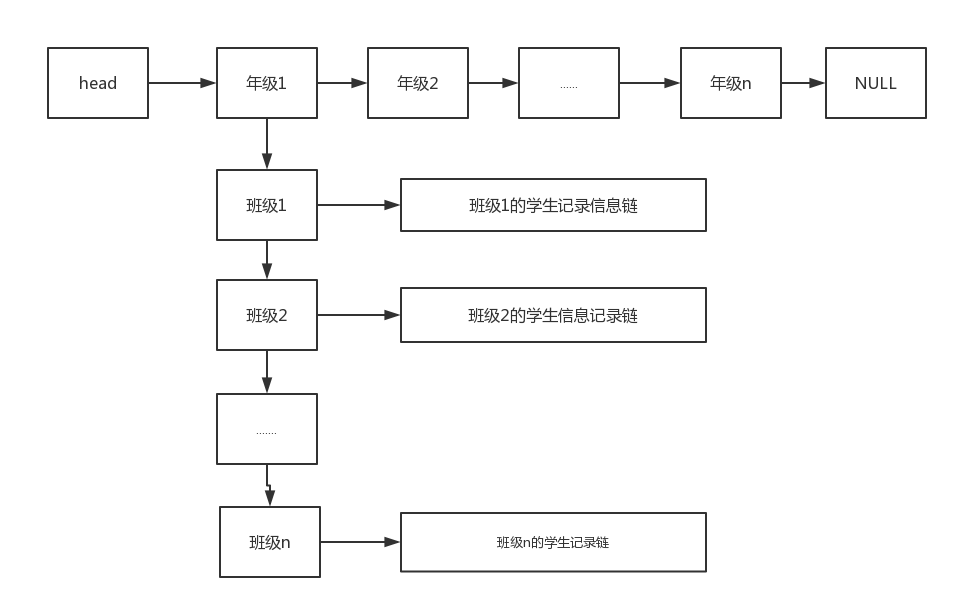


图2.4 数据统计结构图

# 三、数据结构设计

系统所操作的数据分三类，使用三级的狮子交叉链表，链表使用单向链表，其数据结构图如下：



三个结构及学生信息表内容如下：

1.年级信息表：

年级编号、入学时间、入学人数、毕业人数、年级辅导员姓名、辅导员联系电话、年级学生会主席姓名；

//年级信息结构

struct Grade{

char CSNo[10];//年级编号

int time;//入学时间

int num\_enter;//入学人数

int num\_graduate;//毕业人数

char name[10];//年级辅导员姓名

char telephone[20];//辅导员电话

char name\_chairman[10];//学生会主席

struct Class \*classes; //班级信息

struct Grade \*next;

};

2.班级信息表：

班级编号、专业名称、入学人数、入学平均年龄、毕业人数、班长姓名、班长联系电话、班主任姓名、班主任联系电话；

//班级信息结构

struct Class{

char CNo[8];//班级编号

char major[15];//专业名称

int num\_enter;//入学人数

float ave\_of\_age;//入学平均年龄

int num\_graduate;//毕业人数

char monitor[10];//班长姓名

char telephone[20];//班长电话

char teacher[10];//班主任姓名

char telephone\_teacher[20];//班主任电话

struct Student \*student;//学生信息

struct Class \*next;

};

3.学生信息表：

学号、姓名、性别、籍贯、出生年月、联系电话、入学分数、毕业与否、毕业去向。

//学生信息结构

struct Student{

char SNo[10];//学号

char name[10];//学生姓名

char sex;//性别 （1：男；2：女）

char native\_place[10];//籍贯

char year[10];//出生年月

char telephone[20];//联系电话

float score;//入学成绩

int graduate;//是否毕业（0：没毕业； 1：毕业）

char graduate\_place[10];//毕业去向

struct Student \*next;

};

# 四、详细设计

1.主函数流程图，如图4.1：

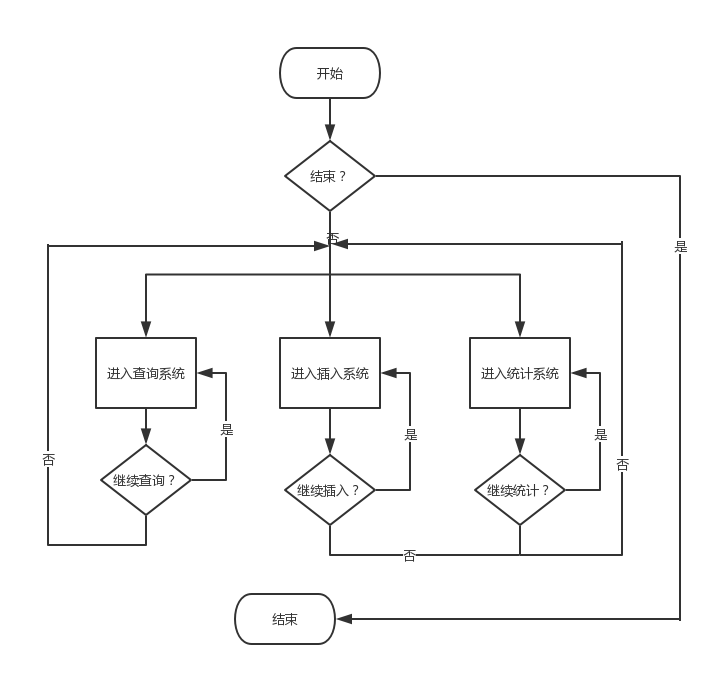


图4.1 主函数流程图

## 2.查询系统

1）年级查询，以年级编号查询为例，如图4.2.1

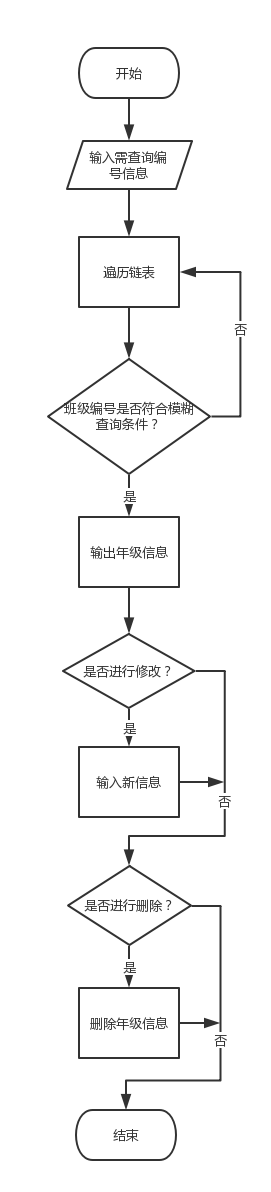


图4.2.1 年级编号查询流程图

1. 班级查询，以班级编号查询为例，如图4.2.2：

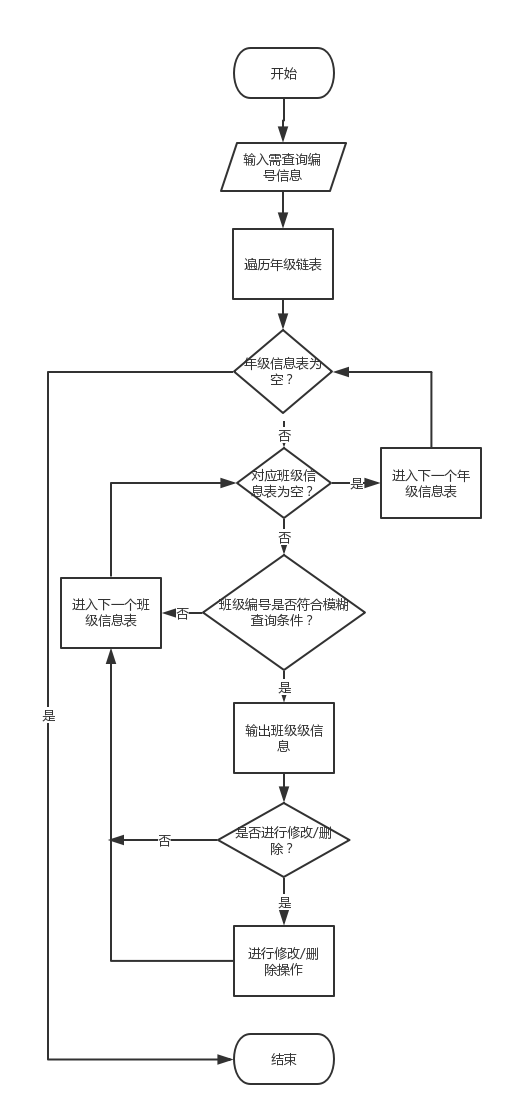


图4.2.2 班级编号查询流程图

1. 学生查询，以入学时间以及毕业去向为例，如图4.2.3：

若按入学时间查询，在遍历年级链表时就应该进行判断是否满足条件；

若查询毕业去向，只需在遍历学生链表时进行判断是否满足条件即可。

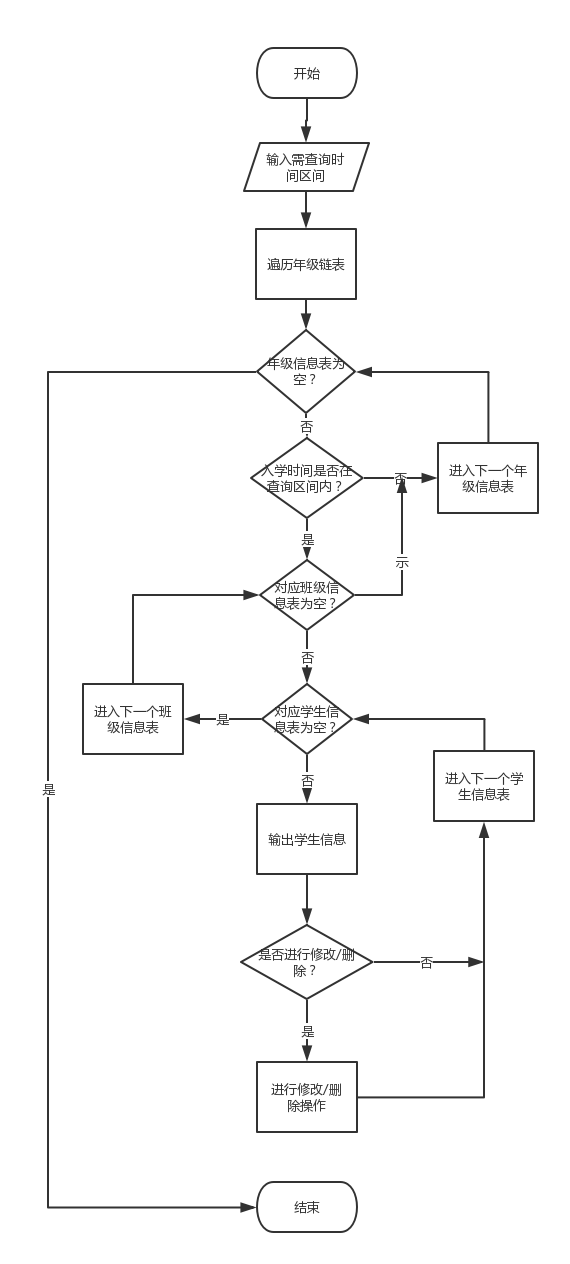


图4.2.3.a 学生入学时间查询流程图

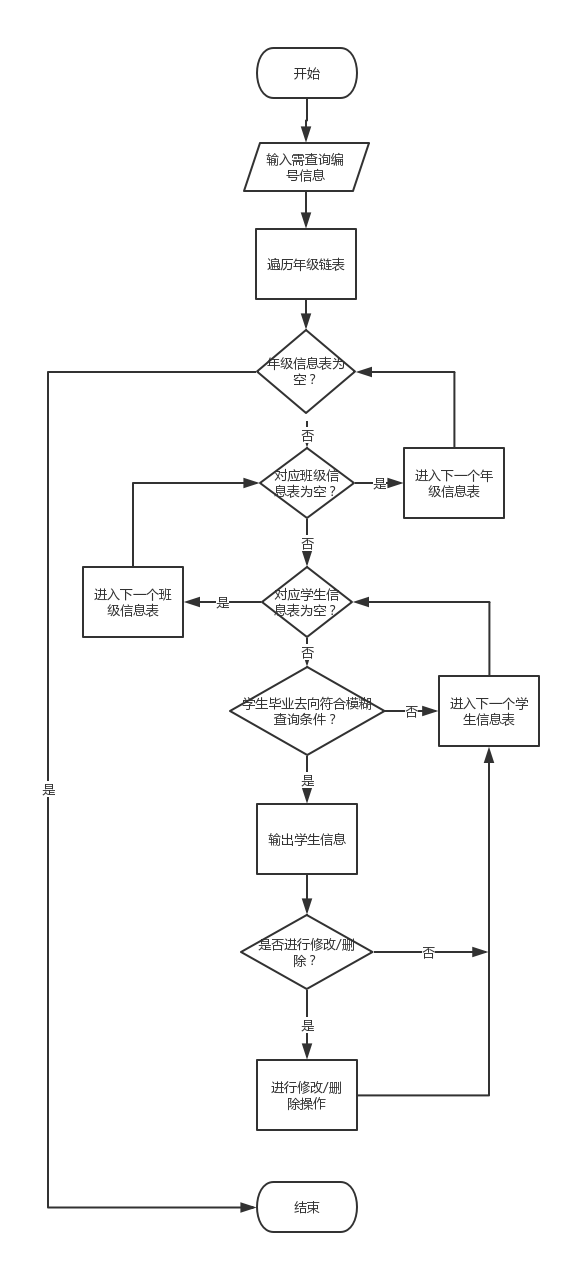


图4.2.3.b 学生毕业去向查询流程图

3.插入系统，如图4.3：

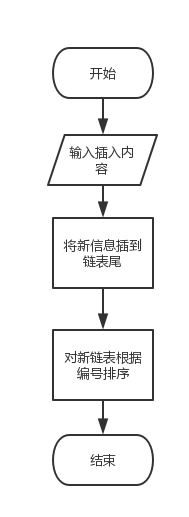


图4.3 插入功能流程图

## 4.统计系统

1）年级统计，如图4.4.1

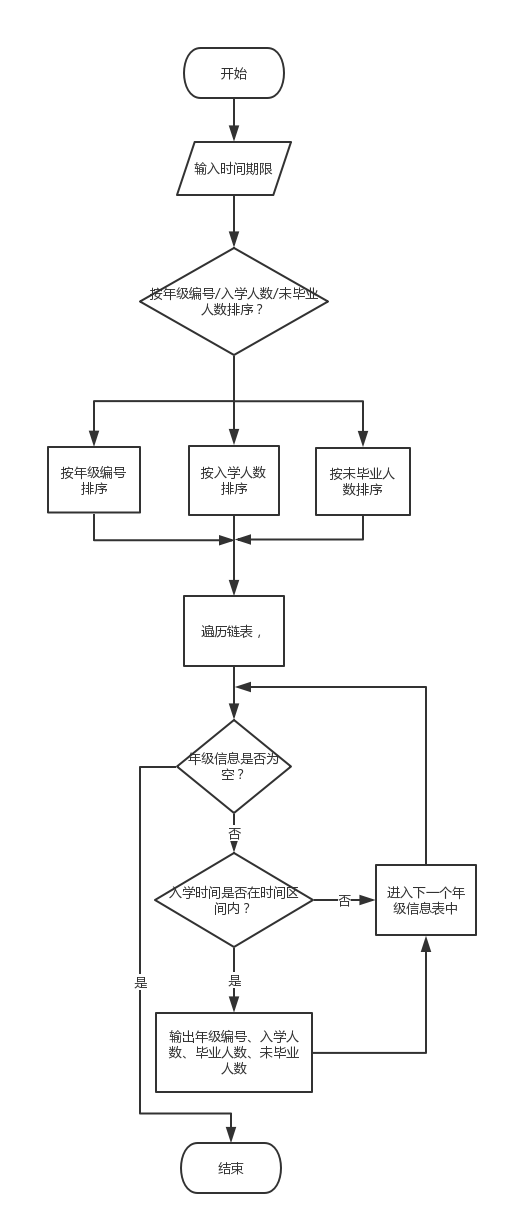


图4.4.1 年级统计流程图

2）企业统计，如图4.4.2：

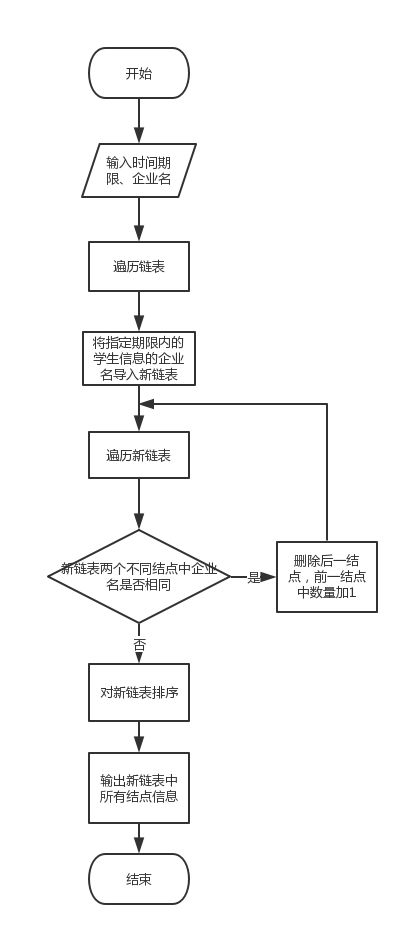


图4.4.2 企业统计流程图

5.录入功能，如图4.5：

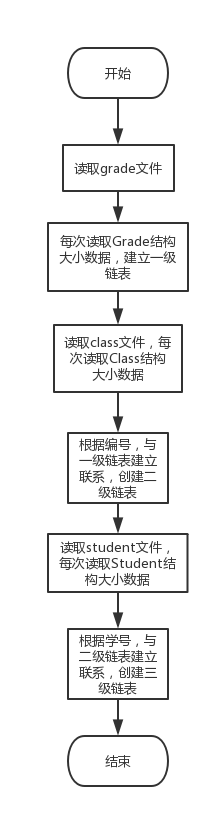


图4.5 录入功能流程图

# 五、系统实现

代码见附录1

主要函数名如下:

freetext //模糊查询函数

create\_grade //创建年级链表

load\_list //录入函数

save //保存函数

seek\_grade1 //年级编号查询

insert\_grade1 //年级插入

count1 //年级统计

revise\_grade //修改功能函数

delete\_grade //删除功能函数

\_seek //查询主函数

\_add //插入主函数

\_census //统计主函数

# 六、运行测试与结果分析

1. 主菜单界面，如图6.1

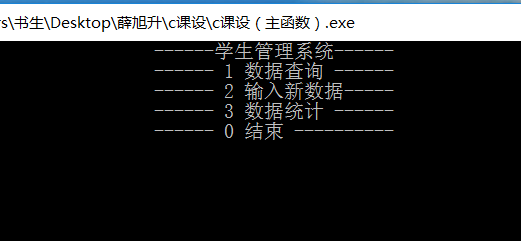


图6.1 主菜单界面

1. 查询系统

1）年级查询，以年级编号和入学时间为例，如图

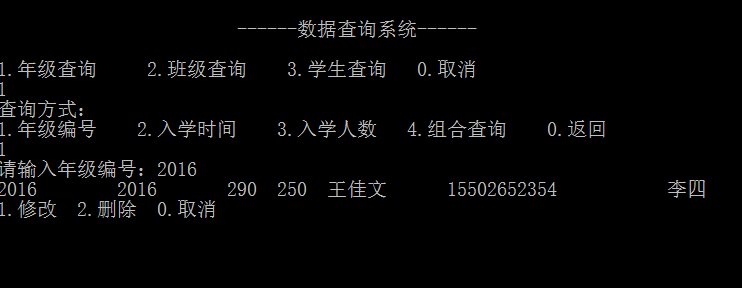


图6.2.1.a 年级编号查询

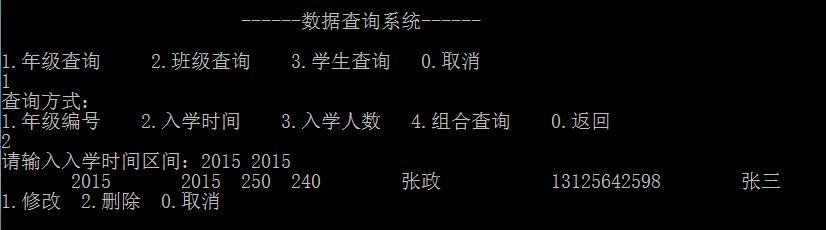


图6.2.1.b 入学时间查询

2）班级查询，以班级编号和人数为例，如图6.2.2



图6.2.2.a 班级编号查询

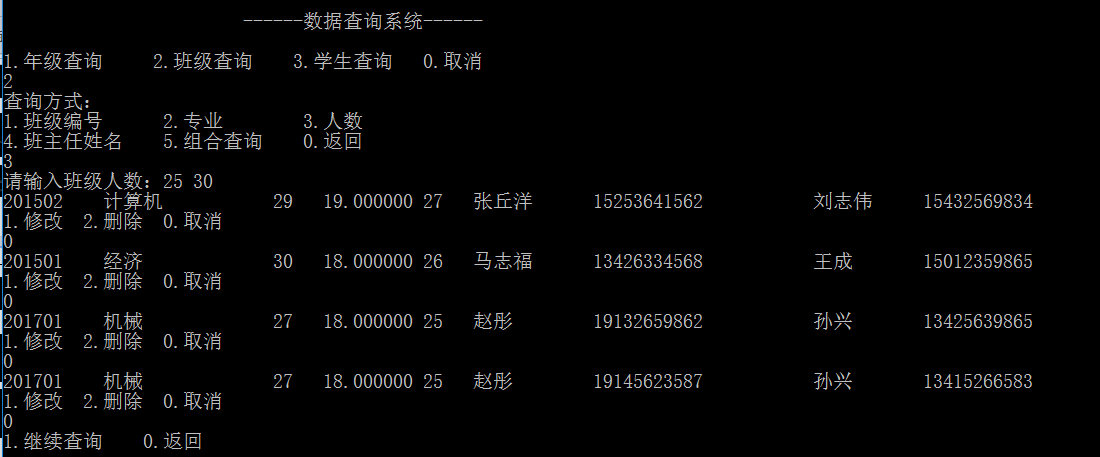


图6.2.2.b 班级人数查询

3）学生查询，以专业和入学时间为例，如图3.2.3：

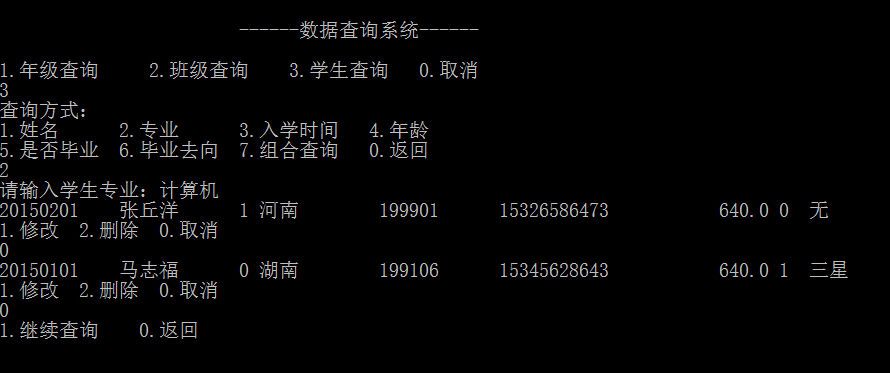


图6.2.3.a 学生专业查询



图6.2.3.b 学生入学时间查询

1. 修改功能，对2015届进行删除操作，如图6.3：

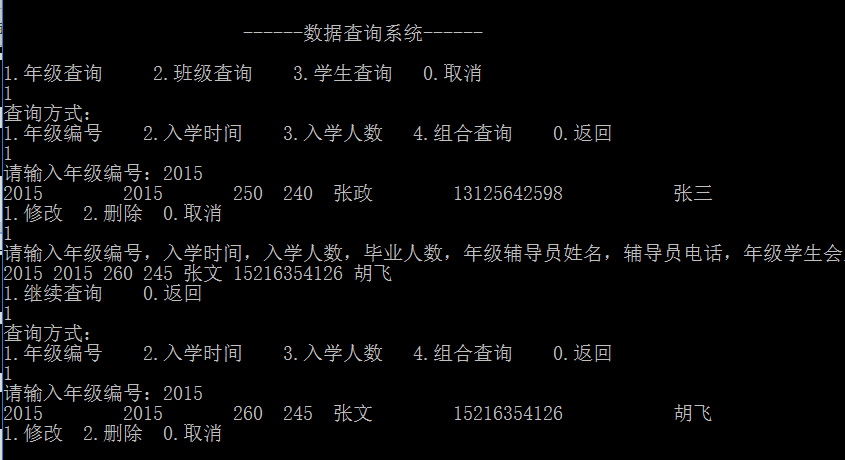
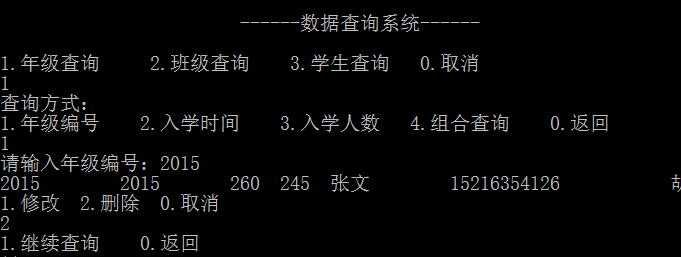


图6.3 修改功能

可以看见，修改之后重新查询信息变更为新输入的信息，修改功能实现。

1. 删除功能，对2015届进行删除操作，如图6.4：



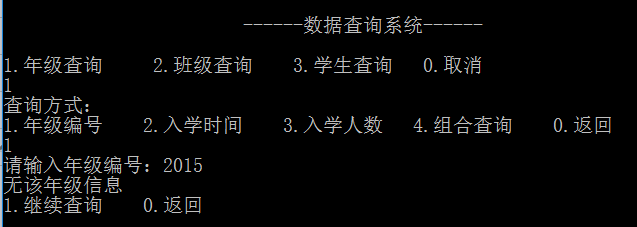


图6.4 删除功能

在进行删除操作后，在查询2015届信息输出的是“无该年级信息”，删除功能已实现。

5）统计功能

1）年级统计，如图6.5.1：

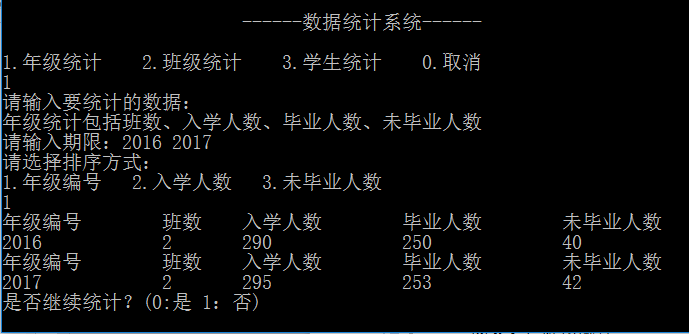


图6.5.1 年级统计

2）班级统计，如图6.5.2：

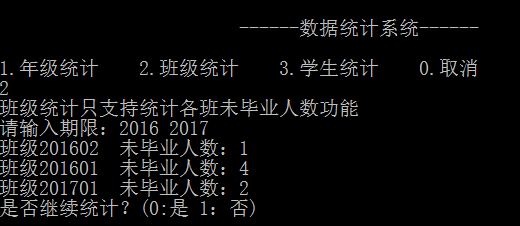


图6.5.2 班级统计

1. 企业统计，如图6.5.3:

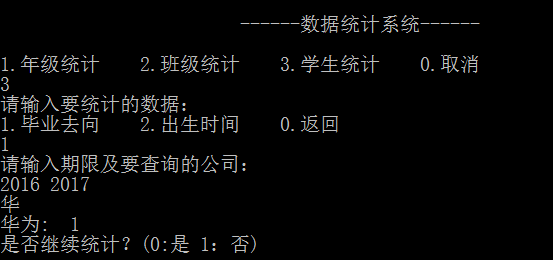


图6.5.3 企业统计

1. 出生时间统计，如图6.5.4：

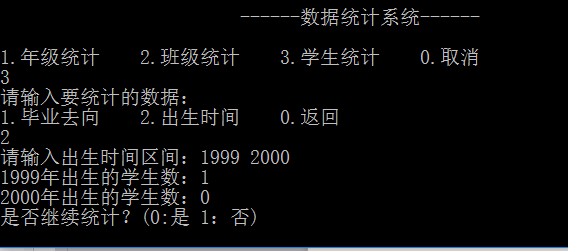


图6.5.4 出生时间统计

# 七、总结

这是我第一次写这么大型且比较复杂的c程序，上学期都是写一些小程序，其难度与代码量完全不成正比。

首先我想说的是我从这次课设设计中得到的教训。这次的c课设任务本来是暑假的任务，暑假就应该完成，但我太懒了，暑假未做，一直到开学前几天才着手准备，由于时间原因，我在做c课设中遇到了许多问题，而且c课设也做的特别简陋，本来想做图形界面，但最后只能写文字界面，我从这次事件中得到了深刻的教训，以后遇事一定尽早做完。

除此以外，我也从这次任务中收获了许多，虽然是最简单的代码，但写这种大型程序与小程序的感觉完全不一样，小程序最主要的是对基本概念得的了解熟悉，而大程序更考验编程者的全局思维、对整体构架的设计以及各种小程序的拼接，就算是最基础的代码，也会出很多难以解决的bug，在debug上花费的精力不比写代码少多少，所以完成整个任务的时间远比想象中的长。

由于代码量的巨额增加，在刚开始任务的时候我感觉无从下手，于是我在网上参考了往届学长学姐学的程序并与室友进行了讨论，最后终于了解了这次的任务，对程序的整体框架有了基本的构思。

由于我只是一个c语言初学者，尽管已经想好了基本结构，在编程时还是遇到了一些难以解决的问题，需要设计一些我以前从未用到的算法，才能更好的实现函数功能。在解决问题的过程中，我学到了许多书本上未提及的知识。

这次课程设计，巩固了我的c语言知识，锻炼了我的动手能力，让我在这一方面有了进一步的提升。

其实，让我感触最深的是，学号c语言，不仅仅在于肯光书本上的知识，最重要的是实践，同时一个人的C语言知识并不能说明他的编程能力，对于学C语言的人来说，动手编程能力比知识更重要。

# 八、参考文献

1. 《C语言与程序设计》 曹计昌，卢萍，李开编著，电子工业出版社

2.《C语言实验与课程设计》 曹计昌，卢萍，李开编著，科学出版社

# 九、指导教师评语

# 附录一

主要功能函数

#include<stdio.h>

#include<string.h>

#include<stdlib.h>

//学生信息结构

struct Student{

char SNo[10];//学号

char name[10];//学生姓名

char sex;//性别 （1：男；2：女）

char native\_place[10];//籍贯

char year[10];//出生年月

char telephone[20];//联系电话

float score;//入学成绩

int graduate;//是否毕业（0：没毕业； 1：毕业）

char graduate\_place[10];//毕业去向

struct Student \*next;

};

//班级信息结构

struct Class{

char CNo[8];//班级编号

char major[15];//专业名称

int num\_enter;//入学人数

float ave\_of\_age;//入学平均年龄

int num\_graduate;//毕业人数

char monitor[10];//班长姓名

char telephone[20];//班长电话

char teacher[10];//班主任姓名

char telephone\_teacher[20];//班主任电话

struct Student \*student;//学生信息

struct Class \*next;

};

//年级信息结构

struct Grade{

char CSNo[10];//年级编号

int time;//入学时间

int num\_enter;//入学人数

int num\_graduate;//毕业人数

char name[10];//年级辅导员姓名

char telephone[20];//辅导员电话

char name\_chairman[10];//学生会主席

struct Class \*classes; //班级信息

struct Grade \*next;

};

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

函数名称：freetext

函数功能：模糊查询

函数参数：参数s2为查询内容，参数s1为需与s2比较的内容

返回值：若 s2为s1子串，返回1，否则返回0

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

int freetext(char s1[],char s2[])

{

int flag=0;

int m=strlen(s1),n=strlen(s2);

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

{

if(s1[i]==s2[0])

for(int j=1;j<n;j++)

if(s1[i+j]=='\0'||s1[i+j]!=s2[j]) break;

else flag=1;

}

if(flag==0) return 0;

else return 1;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

函数名称：creat\_grade

函数功能：创建三级链表年级链表

函数参数：年级链表二级头指针head

返回值：无

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void creat\_grade(struct Grade \*\*head)

{

struct Grade \*hp=NULL,\*p;

char ch;

loop:

p=(struct Grade \*)malloc(sizeof(struct Grade));

printf("请输入年级编号，入学时间，入学人数，毕业人数，年级辅导员姓名，辅导员电话，年级学生会主席\n");

scanf("%s %d %d %d %s %s %s",p->CSNo,&p->time,&p->num\_enter,&p->num\_graduate,p->name,p->telephone,p->name\_chairman);

getchar();

p->classes=NULL;

p->next=hp;

hp=p;

printf("是否继续创建年级信息表？\n");

ch=getchar();getchar();

if(ch=='y'||ch=='Y') goto loop;//创建下一个班级信息表

(\*head)=hp;//head指向新创建的班级信息表

printf("是否创建班级信息表？\n");

ch=getchar();getchar();

if(ch=='y'||ch=='Y')

{

p=(\*head); //p指向班级信息表链链头

while(p!=NULL)

{

printf("%s届的班级信息表:\n",p->CSNo);

struct Class \* pcrs;

creat\_class(&pcrs);//创建学生信息表

p->classes=pcrs;

p=p->next;

}

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

函数名称：load\_list

函数功能：录入功能函数，将文件中内容读入三级十字交叉链表

函数参数：无

返回值：年级链表头指针

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

Grade \* load\_list(void) //读取功能函数

{

FILE \*in1, \*in2, \*in3;

Grade \*hp = NULL, \*p1,\*p2;

Class \*pc1,\*pc2;

Student \*ps1,\*ps2;

if ((in1 = fopen("D://c课设//grade.dat", "rb")) == NULL)

{

printf("文件D://c课设//grade.dat打开失败！\n");

}

if ((in2 = fopen("D://c课设//class.dat", "rb")) == NULL)

{

printf("文件D://c课设//class.dat打开失败！\n");

exit(1);

}

if ((in3 = fopen("D://c课设//student.dat", "rb")) == NULL)

{

printf("文件D://c课设//student打开失败！\n");

exit(1);

}

while (!feof(in1))

{

p1 = (Grade \*)malloc(sizeof(Grade));

fread(p1, sizeof(Grade), 1, in1);

p1->classes = NULL;

if (hp == NULL)

{

hp = p1;

p2 = p1;

}

else

{

p2->next = p1;

p2 = p1;

}

}

p2->next = NULL;

while (!feof(in2))

{

pc1 = (Class \*)malloc(sizeof(Class));

fread(pc1, sizeof(Class), 1, in2);

pc1->student = NULL;

p2 = hp;

while (p2 != NULL)

{

if (char\_to\_int(p2->CSNo)==char\_to\_int(pc1->CNo))

{

if (p2->classes == NULL)

{

p2->classes = pc1;

pc2 = pc1;

}

else

{

pc2->next = pc1;

pc2 = pc1;

}

break;

}

else

p2 = p2->next;

}

}

while (!feof(in3))

{

ps1 = (Student \*)malloc(sizeof(Student));

fread(ps1, sizeof(Student), 1, in3);

p1 = hp;

while (p1 != NULL)

{

pc1 = p1->classes;

while (pc1)

{

if (char\_to\_int(pc1->CNo)==char\_to\_int(ps1->SNo))

{

if (pc1->student== NULL)

{

pc1->student= ps1;

ps2 = ps1;

}

else

{

ps2->next = ps1;

ps2 = ps1;

}

ps1 = (Student \*)malloc(sizeof(Student));

fread(ps1, sizeof(Student), 1, in3);

}

else

pc1 = pc1->next;

}

p1 = p1->next;

}

}

fclose(in1);

fclose(in2);

fclose(in3);

return hp;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

函数名称：save

函数功能：将创建的或修改过后的三级十字交叉链表存储进对应的三个文件

函数参数：年级链表头指针

返回值：无

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void save(struct Grade \*head)

{

FILE \*out1,\*out2,\*out3;

Grade \*p=head;

out1=fopen("D://c课设//grade.dat","w+");

out2=fopen("D://c课设//class.dat","w+");

out3=fopen("D://c课设//student.dat","w+");

while(p!=NULL){

Class \*pc=p->classes;

fwrite(p,sizeof(Grade),1,out1);

while(pc!=NULL){

Student \*ps=pc->student;

fwrite(pc,sizeof(Class),1,out2);

while(ps!=NULL){

fwrite(ps,sizeof(Student),1,out3);

ps=ps->next;

}

pc=pc->next;

}

p=p->next;

}

fclose(out1);

fclose(out2);

fclose(out3);

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

函数名称：seek\_grade1

函数功能：根据年级编号查询年级信息，若查询到输出查询结果，否则输出“无该年级信息”

可进行模糊查询

函数参数：数组s为需查询的编号内容，head为年级链表头指针

返回值：无

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void seek\_grade1(char s[],struct Grade \*head)//年级编号查询

{

struct Grade \*p=head;

int flag=0;

for(;p!=NULL;p=p->next)

if(strstr(p->CSNo,s)!=0)

{

printf("%-11s %-10d %-4d %-4d %-11s %-21s %-11s\n",p->CSNo,p->time,p->num\_enter,p->num\_graduate,p->name,p->telephone,p->name\_chairman);

printf("1.修改 2.删除 0.取消\n");

int n;scanf("%d",&n);

switch(n){

case 1:

revise\_grade(p);

break;

case 2:

delete\_grade(head,p);

break;

case 0:

break;

}

flag=1;

}

if(flag==0) printf("无该年级信息\n");

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

函数名称：seek\_class1

函数功能：根据班级编号查询班级信息，若查询到输出查询结果，

否则输出“无该班级信息”，可进行模糊查询

函数参数：s为 班级编号内容，head为年级链表头指针

返回值：无

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void seek\_class1(char s[],struct Grade \*head)//班级编号查询

{

struct Grade \*p=head;

int flag;

for(;p!=NULL;p=p->next)

{

struct Class \*q=p->classes;

for(;q!=NULL;q=q->next)

{

if(strstr(q->CNo,s)!=0)

{

printf("%-9s %-16s %-4d %-3f %-4d %-11s %-21s %-10s %-20s\n",q->CNo,q->major,q->num\_enter,q->ave\_of\_age,\

q->num\_graduate,q->monitor,q->telephone,q->teacher,q->telephone\_teacher);

printf("1.修改 2.删除 0.取消\n");

int n;scanf("%d",&n);

switch(n){

case 1:

revise\_class(q);

break;

case 2:

delete\_class(head,q);

break;

case 0:

break;

}

flag=1;

}

}

}

if(flag==0) printf("无该班级信息\n");

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

函数名称：seek\_student1

函数功能：根据学生姓名查询学生信息，若查询到输出该学生信息，

否则输出“无该学生信息” ，可进行模糊查询

函数参数：s为学生姓名，head为班级链表头链表

返回值：无

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void seek\_student1(char s[],struct Grade \*head)//学生姓名查询

{

struct Grade \*p=head;

int flag=0;

for(;p!=NULL;p=p->next)

{

struct Class \*q=p->classes;

for(;q!=NULL;q=q->next)

{

struct Student \*t=q->student;

for(;t!=NULL;t=t->next)

{

if(strstr(t->name,s)!=0)

{

printf("%-11s %-11s %-1c %-11s %-11s %-21s %-4.1f %-2d %-11s\n",t->SNo,t->name,t->sex,t->native\_place,t->year,t->telephone,t->score,t->graduate,t->graduate\_place);

printf("1.修改 2.删除 0.取消\n");

int n;scanf("%d",&n);

switch(n){

case 1:

revise\_student(t);

break;

case 2:

delete\_student(head,t);

break;

case 0:

break;

}

flag=1;

}

}

}

}

if(flag==0) printf("无该学生信息");

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

函数名称：insert\_grade

函数功能：插入一条年级信息到年级链表里

函数参数：年级链表头指针head

返回值：新的年级链表头指针

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

Grade \* insert\_grade(Grade \* head) //年级插入

{

Grade \*p;

p= (Grade \*)malloc(sizeof(Grade));

printf("请输入年级编号，入学时间，入学人数，毕业人数，年级辅导员姓名，辅导员电话，年级学生会主席\n");

scanf("%s%d%d%d%s%s%s",p->CSNo,&p->time,&p->num\_enter,&p->num\_graduate,p->name,p->telephone,p->name\_chairman);

p->next=head;

return p;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

函数名称：insert\_class

函数功能：插入一条班级信息到指定年级中

函数参数：指向指定的年级的指针head

返回值：该年级下的班级链表的头指针

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

Class \* insert\_class(struct Grade \*head)//班级插入

{

struct Class \*p;

p=(Class \*)malloc(sizeof(Class));

printf("请输入班级编号，专业名称，入学人数,入学平均年龄，毕业人数,班长姓名，班长联系电话，班主任姓名，班主任联系电话\n");

scanf("%s %s %d %f %d %s %s %s %s",p->CNo,p->major,&p->num\_enter,&p->ave\_of\_age,&p->num\_graduate,p->monitor,p->telephone,p->teacher,p->telephone\_teacher);

p->next=head->classes;

head->classes=p;

return p;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

函数名称：insert\_student

函数功能：插入一条学生信息到指定班级中

函数参数：指向指定班级的的指针

返回值：该班级下的学生链表头指针

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

Student \* insert\_student(struct Class \*head)//学生插入

{

struct Student \*tail;

tail=(struct Student \*)malloc(sizeof(struct Student));

printf("请输入学号，姓名，性别，籍贯，出生年月，联系电话，入学分数，是否毕业，毕业去向\n");

scanf("%s %s %c %s %s %s %f %d %s",tail->SNo,tail->name,&tail->sex,tail->native\_place,tail->year,tail->telephone,&tail->score,&tail->graduate,tail->graduate\_place);

tail->next=head->student;

head->student=tail;

return tail;

}

//年级统计

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

函数名称：count1

函数功能：指定期限的年级统计

函数参数：参数x为前面的时间，y为后面的时间，head为年级链表头指针

返回值：无

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void count1(int x,int y,struct Grade \*head)

{

int count=0;

int flag=0;

struct Grade \*p=head;

for(;p!=NULL;p=p->next)

{

if(p->time>=x&&p->time<=y)

{

flag=1;

struct Class \*q=p->classes;

for(;q!=NULL;q=q->next) count++;

printf("年级编号\t班数\t入学人数\t毕业人数\t未毕业人数\n");

printf("%s\t\t%d\t%d\t\t%d\t\t%d\n",p->CSNo,count,p->num\_enter,\

p->num\_graduate,p->num\_enter-p->num\_graduate);

count=0;

}

}

if(flag==0) printf("该期间无年级信息！");

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

函数名称：count2

函数功能：无指定期限的年级统计

函数参数：head为年级链表头指针

返回值：无

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void count2(struct Grade \*head)

{

int count=0;

int flag=0;

struct Grade \*p=head;

for(;p!=NULL;p=p->next)

{

flag=1;

struct Class \*q=p->classes;

for(;q!=NULL;q=q->next) count++;

printf("年级编号\t班数\t入学人数\t毕业人数\t未毕业人数\n");

printf("%s\t\t%d\t%d\t\t%d\t\t%d\n",p->CSNo,count,p->num\_enter,\

p->num\_graduate,p->num\_enter-p->num\_graduate);

count=0;

}

if(flag==0) printf("该期间无年级信息！");

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

函数名称：count\_class1

函数功能：统计指定期限各班级未毕业人数

函数参数：参数x为前面的时间，y为后面的时间

返回值：无

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void count\_class1(int x,int y,struct Grade \*head)

{

struct Grade \*p=head;

struct Count \*headp=NULL,\*tail=NULL;

for(;p!=NULL;p=p->next)//将第一个年级存入链表

{

if(p->time>=x&&p->time<=y&&p->classes!=NULL)//在指定期限内且该年级中有班级信息时

{

struct Class \*q;

q=p->classes;

headp=(struct Count\*)malloc(sizeof(struct Count));

headp->num=q->num\_enter-q->num\_graduate;strcpy(headp->CNO,q->CNo);//将班级编号与未毕业人数存储在新的链表中

tail=headp;//tail指向头指针

for(q=q->next;q!=NULL;q=q->next)//继续建立链表

{

tail->next=(struct Count\*)malloc(sizeof(struct Count));

tail=tail->next;

tail->num=q->num\_enter-q->num\_graduate;strcpy(tail->CNO,q->CNo);

}

tail->next=NULL;

p=p->next;

break;

}

}

for(;p!=NULL;p=p->next)//建立后续链表

{

if(p->time>=x&&p->time<=y&&p->classes!=NULL)//在指定期限内且该年级中有班级信息时

{

struct Class \*q;

q=p->classes;

for(q=q->next;q!=NULL;q=q->next)

{

tail->next=(struct Count\*)malloc(sizeof(struct Count));

tail=tail->next;

tail->num=q->num\_enter-q->num\_graduate;strcpy(tail->CNO,q->CNo);

}

tail->next=NULL;

}

}

tail=headp;

sort1(tail);

tail=headp;

for(;tail!=NULL;tail=tail->next)

printf("班级%s 未毕业人数：%d\n",tail->CNO,tail->num);

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

函数名称：count\_class1

函数功能：统计无指定期限各班级未毕业人数

函数参数：无

返回值：无

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

//统计各班级未毕业人数

void count\_class2(struct Grade \*head)

{

struct Grade \*p=head;

struct Count \*headp=NULL,\*tail=NULL;

for(;p!=NULL;p=p->next)//将第一个年级存入链表

{

if(p->classes!=NULL)//该年级中有班级信息时

{

struct Class \*q;

q=p->classes;

headp=(struct Count\*)malloc(sizeof(struct Count));

headp->num=q->num\_enter-q->num\_graduate;strcpy(headp->CNO,q->CNo);//将班级编号与未毕业人数存储在新的链表中

tail=headp;//tail指向头指针

for(q=q->next;q!=NULL;q=q->next)//继续建立链表

{

tail->next=(struct Count\*)malloc(sizeof(struct Count));

tail=tail->next;

tail->num=q->num\_enter-q->num\_graduate;strcpy(tail->CNO,q->CNo);

}

tail->next=NULL;

p=p->next;

break;

}

}

for(;p!=NULL;p=p->next)//建立后续链表

{

if(p->classes!=NULL)//该年级中有班级信息时

{

struct Class \*q;

q=p->classes;

for(q=q->next;q!=NULL;q=q->next)

{

tail->next=(struct Count\*)malloc(sizeof(struct Count));

tail=tail->next;

tail->num=q->num\_enter-q->num\_graduate;strcpy(tail->CNO,q->CNo);

}

tail->next=NULL;

}

}

tail=headp;

sort1(tail);

tail=headp;

for(;tail!=NULL;tail=tail->next)

printf("班级%s 未毕业人数：%d",tail->CNO,tail->num);

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

函数名称：brith

函数功能：统计指定期限内各年出生人数

函数参数：s1为前面的时间，s2为后面的时间 ，head为年级链表头指针

返回值：无

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

int brith(int s1,int s2,struct Grade \*head)

{

struct Grade \*p=head;

for(int n=s1;n<=s2;n++)

{

int count=0;

for(;p!=NULL;p=p->next)

{

if(p->classes!=NULL)

{

struct Class \*q=p->classes;

for(;q!=NULL;q=q->next)

{

struct Student \*t=q->student;

for(;t!=NULL;t=t->next)

{

if(n==char\_to\_int(t->year)) count++;

}

}

}

}

printf("%d年出生的学生数：%d\n",n,count);

p=head;

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

函数名称：count\_student1

函数功能：统计指定期限毕业后到某企业的人数，输出企业名称与

对应人数，企业名称可模糊查询

函数参数：参数x为前面的时间，y为后面的时间，head为年级链表头

指针，s为统计的公司

返回值：无

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void count\_student1(int x,int y,struct Grade \*head,char s[])

{

struct Grade \*p=head;

struct Count2 \*headp=NULL,\*tail;

for(;p!=NULL;p=p->next)

{

if(p->time>=x&&p->time<=y&&p->classes!=NULL)//建立第一个年级的链表

{

struct Class \*q=p->classes;

headp=(struct Count2\*)malloc(sizeof(struct Count2));//建立头指针

strcpy(headp->graduate\_place,q->student->graduate\_place);

headp->num=1;

tail=headp;//tail指向头指针

for(;q!=NULL&&q->student!=NULL;q=q->next)//遍历班级

{

struct Student \*t=q->student;

for(t=t->next;t!=NULL;t=t->next)//遍历学生

{

tail->next=(struct Count2\*)malloc(sizeof(struct Count2));

tail=tail->next;

strcpy(tail->graduate\_place,t->graduate\_place);

tail->num=1;

}

}

tail->next=NULL;

p=p->next;

break;

}

}

for(;p!=NULL;p=p->next)

{

if(p->time>=x&&p->time<=y&&p->classes!=NULL)

{

struct Class \*q=p->classes;

for(;q!=NULL&&q->student!=NULL;q=q->next)

{

struct Student \*t=q->student;

for(;t!=NULL;t=t->next)

{

tail->next=(struct Count2\*)malloc(sizeof(struct Count2));

tail=tail->next;

strcpy(tail->graduate\_place,t->graduate\_place);

tail->num=1;

}

}

}

tail->next=NULL;

}

tail=headp;

arrange(tail);

tail=headp;

sort2(tail);

tail=headp;

int flag=0;

for(;tail!=NULL;tail=tail->next)

if(strstr(tail->graduate\_place,s)!=0)

{

printf("%s: %d\n",tail->graduate\_place,tail->num);flag=1;

}

if(flag==0) printf("该期间无毕业学生去往该公司\n");

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

函数名称：count\_student2

函数功能：统计无指定期限毕业后到某企业的人数

函数参数：head为年级链表头指针，s为企业名

返回值：无

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void count\_student2(struct Grade \*head,char s[])

{

struct Grade \*p=head;

struct Count2 \*headp=NULL,\*tail;

for(;p!=NULL;p=p->next)

{

if(p->classes!=NULL)//建立第一个年级的链表

{

struct Class \*q=p->classes;

for(;q->student==NULL;q=q->next);//使q指向第一个有学生信息的班级链表

headp=(struct Count2\*)malloc(sizeof(struct Count2));//建立头指针

strcpy(headp->graduate\_place,q->student->graduate\_place);

headp->num=1;

tail=headp;//tail指向头指针

for(;q!=NULL;q=q->next)//遍历班级

{

struct Student \*t=q->student;

for(t=t->next;t!=NULL;t=t->next)//遍历学生

{

tail->next=(struct Count2\*)malloc(sizeof(struct Count2));

tail=tail->next;

strcpy(tail->graduate\_place,t->graduate\_place);

tail->num=1;

}

}

tail->next=NULL;

p=p->next;

break;

}

}

for(;p!=NULL;p=p->next)

{

if(p->classes!=NULL)

{

struct Class \*q=p->classes;

for(;q!=NULL;q=q->next)

{

struct Student \*t=q->student;

for(;t!=NULL;t=t->next)

{

tail->next=(struct Count2\*)malloc(sizeof(struct Count2));

tail=tail->next;

strcpy(tail->graduate\_place,t->graduate\_place);

tail->num=1;

}

}

}

tail->next=NULL;

}

tail=headp;

arrange(tail);

tail=headp;

sort2(tail);

tail=headp;

int flag=0;

for(;tail!=NULL;tail=tail->next)

if(freetext(tail->graduate\_place,s)==1)

{

printf("%s: %d\n",tail->graduate\_place,tail->num);flag=1;

}

if(flag==0) printf("该期间无毕业学生去往该公司");

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

函数名称：\_add

函数功能：插入功能总函数

函数参数：年级链表二级头指针head，年级系统返回主菜单的标识flag

返回值：无

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void \_add(int \*flag,struct Grade \*\*head)

{

add:

printf("\n\t\t\t------数据输入系统------\n\n");

printf("1.插入年级信息 2.插入班级信息 3.插入学生信息 0.取消\n");

int y;scanf("%d",&y);

switch(y){

case 1:

(\*head)=insert\_grade(\*head);

sort\_grade(\*head);

printf("是否继续插入信息？(0:是 1:否)：\n");

int n1;scanf("%d",&n1);

if(n1==0) goto add;

else \*flag=1;

break;

case 2:

printf("请输入要插入班级的年级编号：");

char s1[10];scanf("%s",s1);

struct Grade \*q;

q=\*head;

for(;q!=NULL;q=q->next)

{

if(strcmp(q->CSNo,s1)==0)

{

q->classes=insert\_class(q);

break;

}

}

sort\_class(q->classes);

printf("是否继续插入信息？(0:是 1:否)：\n");

int n2;scanf("%d",&n2);

if(n2==0) goto add;

else \*flag=1;

break;

case 3:

printf("请输入要插入学生的班级编号：");

char s2[8];scanf("%s",s2);

struct Grade \*p;

p=\*head;

for(;p!=NULL;p=p->next)

{

struct Class \*q=p->classes;

for(;q!=NULL;q=q->next)

{

if(strcmp(q->CNo,s2)==0)

{

q->student=insert\_student(q);

sort\_student(q->student);

break;

}

}

}

printf("是否继续插入信息？(0:是 1:否)：\n");

int n3;scanf("%d",&n3);

if(n3==0) goto add;

else \*flag=1;

break;

case 0:

\*flag=1;break;

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

函数名称：\_seek

函数功能：查询功能主函数

函数参数：head为年级链表头指针，flag为返回主函数的标识

返回值：无

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void \_seek(int \*flag,Grade \*head)

{

seek:

printf("\n\t\t\t------数据查询系统------\n\n");

printf("1.年级查询 2.班级查询 3.学生查询 0.取消\n");

int x;scanf("%d",&x);

switch(x){

case 1:

seek1:

printf("查询方式：\n");

printf("1.年级编号 2.入学时间 3.入学人数 4.组合查询 0.返回\n");

int x;scanf("%d",&x);

switch(x){

case 1:

printf("请输入年级编号：");

char s[10];

scanf("%s",s);

seek\_grade1(s,head);

printf("1.继续查询 0.返回\n");

int i1;scanf("%d",&i1);

switch(i1){

case 1:goto seek1;

case 0:goto seek;

}

break;

case 2:

printf("请输入入学时间区间：");

int n1,n2;

scanf("%d %d",&n1,&n2);

seek\_grade2(n1,n2,head);

printf("1.继续查询 0.返回\n");

int i2;scanf("%d",&i2);

switch(i1){

case 1:goto seek1;

case 0:goto seek;

}

break;

case 3:

printf("请输入入学人数区间：");

int n3,n4;

scanf("%d %d",&n3,&n4);

seek\_grade3(n3,n4,head);

printf("1.继续查询 0.返回\n");

int i3;scanf("%d",&i3);

switch(i3){

case 1:goto seek1;

case 0:goto seek;

}

break;

case 4:

printf("请按顺序输入年级编号、入学时间区间、入学人数区间：\n");

char t[10];

int m1,m2,m3,m4;

scanf("%s %d %d %d %d",t,&m1,&m2,&m3,&m4);

seek\_grade4(t,m1,m2,m3,m4,head);

printf("1.继续查询 0.返回\n");

int i4;scanf("%d",&i4);

switch(i4){

case 1:goto seek1;

case 0:goto seek;

}

break;

case 0:

goto seek;

}

break;

case 2:

seek2:

printf("查询方式：\n");

printf("1.班级编号 2.专业 3.人数\

\n4.班主任姓名 5.组合查询 0.返回\n");

int y;scanf("%d",&y);

switch(y){

case 1:

printf("请输入班级编号：");

char s[8];scanf("%s",s);

seek\_class1(s,head);

printf("1.继续查询 0.返回\n");

int i1;scanf("%d",&i1);

switch(i1){

case 1:goto seek2;

case 0:goto seek;

}

break;

case 2:

printf("请输入专业名称：");

char t[15];scanf("%s",t);

seek\_class2(t,head);

printf("1.继续查询 0.返回\n");

int i2;scanf("%d",&i2);

switch(i2){

case 1:goto seek2;

case 0:goto seek;

}

break;

case 3:

printf("请输入班级人数：");

int n1,n2;scanf("%d%d",&n1,&n2);

seek\_class4(n1,n2,head);

printf("1.继续查询 0.返回\n");

int i3;scanf("%d",&i3);

switch(i3){

case 1:goto seek2;

case 0:goto seek;

}

break;

case 4:

printf("请输入班主任姓名：");

char r[10];scanf("%s",r);

seek\_class3(r,head);

printf("1.继续查询 0.返回\n");

int i4;scanf("%d",&i4);

switch(i4){

case 1:goto seek2;

case 0:goto seek;

}

break;

case 5:

printf("请按顺序输入班级编号、专业名称、班级人数、班主任姓名\n");

char v1[8],v2[15],v3[20];

int m1,m2;

scanf("%s %s %d %d %s",v1,v2,&m1,&m2,v3);

seek\_class5(v1,v2,v3,m1,m2,head);

printf("1.继续查询 0.返回\n");

int i5;scanf("%d",&i5);

switch(i5){

case 1:goto seek2;

case 0:goto seek;

}

break;

case 0:

goto seek;

}

break;

case 3:

seek3:

printf("查询方式：\n");

printf("1.姓名 2.专业 3.入学时间 4.年龄\

\n5.是否毕业 6.毕业去向 7.组合查询 0.返回\n");

int z;scanf("%d",&z);

switch(z){

case 1:

printf("请输入学生姓名：");

char s1[10];scanf("%s",s1);

seek\_student1(s1,head);

printf("1.继续查询 0.返回\n");

int i1;scanf("%d",&i1);

switch(i1){

case 1:goto seek3;

case 0:goto seek;

}

break;

case 2:

printf("请输入学生专业：");

char s2[15];scanf("%s",s2);

seek\_student5(s2,head);

printf("1.继续查询 0.返回\n");

int i2;scanf("%d",&i2);

switch(i2){

case 1:goto seek3;

case 0:goto seek;

}

break;

case 3:

printf("请输入学生入学时间：");

int n1,n4;scanf("%d%d",&n1,&n4);

seek\_student4(n1,n4,head);

printf("1.继续查询 0.返回\n");

int i3;scanf("%d",&i3);

switch(i3){

case 1:goto seek3;

case 0:goto seek;

}

break;

case 4:

printf("请输入学生年龄：");

int n2,n5;scanf("%d%d",&n2,&n5);

seek\_student6(n2,n5,head);

printf("1.继续查询 0.返回\n");

int i4;scanf("%d",&i4);

switch(i4){

case 1:goto seek3;

case 0:goto seek;

}

break;

case 5:

printf("请选择输出毕业\未毕业学生（0：未毕业； 1：毕业）：");

int n3;scanf("%d",&n3);

seek\_student3(n3,head);

printf("1.继续查询 0.返回\n");

int i5;scanf("%d",&i5);

switch(i5){

case 1:goto seek3;

case 0:goto seek;

}

break;

case 6:

printf("请输入企业名称：");

char s3[10];scanf("%s",s3);

seek\_student2(s3,head);

printf("1.继续查询 0.返回\n");

int i6;scanf("%d",&i6);

switch(i6){

case 1:goto seek3;

case 0:goto seek;

}

break;

case 7:

printf("请按顺序输入学生姓名、专业、入学时间、年龄、是否毕业、毕业去向：\n");

char t1[10],t2[15],t3[10];

int m1,m2,m3,m4,m5;

scanf("%s %s %d %d %d %d %d %s",s1,s2,&m1,&m2,&m3,&m4,&m5,s3);

seek\_student7(s1,m3,m4,s2,m1,m2,m5,s3,head);

printf("1.继续查询 0.返回\n");

int i7;scanf("%d",&i7);

switch(i7){

case 1:goto seek3;

case 0:goto seek;

}

break;

case 0:

goto seek;

}

break;

case 0:

\*flag=1;

break;

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

函数名称：\_census

函数功能：统计功能主函数

函数参数：head为年级链表头指针，flag为返回主菜单的标识

返回值：无

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void \_census(int \*flag,struct Grade \*head)

{

census:

printf("\n\t\t\t------数据统计系统------\n\n");

printf("1.年级统计 2.班级统计 3.学生统计 0.取消\n");

int z;scanf("%d",&z);

switch(z){

case 1:

census1:

printf("请输入要统计的数据：\n");

printf("年级统计包括班数、入学人数、毕业人数、未毕业人数\n");

printf("请输入期限：");

int y1,y2;scanf("%d%d",&y1,&y2);

printf("请选择排序方式：\n");

printf("1.年级编号 2.入学人数 3.未毕业人数\n");

int j1;scanf("%d",&j1);

switch(j1){

case 1:

sort\_grade(head);break;

case 2:

sort3(head);break;

case 3:

sort4(head);break;

}

count1(y1,y2,head);

printf("是否继续统计？(0:是 1：否)\n");

int m1;scanf("%d",&m1);

if(m1==0) goto census1;

else goto census;

break;

case 2:

census2:

printf("班级统计只支持统计各班未毕业人数功能\n");

printf("请输入期限：");

int t1,t2;

scanf("%d%d",&t1,&t2);

count\_class1(t1,t2,head);

printf("是否继续统计？(0:是 1：否)\n");

int m2;scanf("%d",&m2);

if(m2==0) goto census2;

else goto census;

break;

case 3:

census3:

printf("请输入要统计的数据：\n");

printf("1.毕业去向 2.出生时间 0.返回\n");

int n3;scanf("%d",&n3);

switch(n3){

case 1:

printf("请输入期限及要查询的公司：\n");

int r1,r2;char s[10];

scanf("%d%d%s",&r1,&r2,s);

count\_student1(r1,r2,head,s);

break;

case 2:

printf("请输入出生时间区间：");

int r3,r4;

scanf("%d%d",&r3,&r4);

brith(r3,r4,head);

break;

case 0:

goto census;

}

printf("是否继续统计？(0:是 1：否)\n");

int m3;scanf("%d",&m3);

if(m3==0) goto census3;

else goto census;

break;

case 0:

\*flag=1; break;

}

}

int main()

{

struct Grade \*head=NULL;

head=load\_list();

loop:

printf("\t\t\t------学生管理系统------\n");

printf("\t\t\t------ 1 数据查询 ------\n");

printf("\t\t\t------ 2 输入新数据-----\n");

printf("\t\t\t------ 3 数据统计 ------\n");

printf("\t\t\t------ 0 结束 ----------\n");

int n;

int flag=0;

scanf("%d",&n);

switch(n){

case 1:

\_seek(&flag,head);//查询系统函数

if(flag==1){

flag=0;goto loop;//返回主菜单

}

break;

case 2:

\_add(&flag,&head);//插入系统函数

if(flag==1){

flag=0;goto loop;//返回主菜单

}

break;

case 3:

\_census(&flag,head);//统计函数主菜单

if(flag==1){

flag=0;goto loop;//返回主菜单

}

break;

case 0:

break;

}

save(head);//将链表存到文件中

return 0;

}

# 附录二

文件中数据，用于测试

年级：

年级编号 入学时间 入学人数 毕业人数 年级辅导员姓名 辅导员电话 年级学生会主席

2015 2015 300 250 张政 13145567890 张三

2016 2016 290 250 王佳文 15023564885 李四

2017 2017 295 253 王乾龙 13425551918 王五

班级：

班级编号 专业名称 入学人数 入学平均年龄 毕业人数 班长姓名 班长联系电话 班主任姓名 班主任联系电话

201501 经济 30 18 26 马志福 13456289702 王成 15026489237

201502 计算机 29 19 27 张丘洋 15246298735 刘志伟 15426358972

201601 新闻 31 18 27 王伟 13456298735 李志伟 15423658925

201602 材料 34 19 30 黄强 15426589350 周大为 133256489570

201701 机械 27 18 25 赵彤 19125463584 孙兴 13456298562

学生：

学号 姓名 性别 籍贯 出生年月 联系电话 入学分数 毕业与否 毕业去向

20150101 马志福 1 宁夏 199811 15123648926 630 1 三星

20150201 张丘洋 1 河南 199901 15326586473 640 0 无

20160101 王伟 1 湖北 199808 18123654982 635 1 360

20160201 黄强 1 上海 199905 18145626534 650 0 无

20170101 赵彤 0 浙江 199711 15326452365 645 1 华为