案例分析报告

课程名称：学生信息管理系统

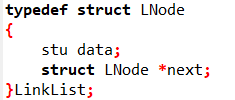
学生姓名：马铭阳，胡斌斌，彭祥，方强

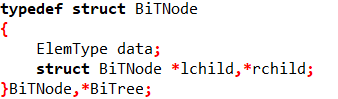
课题背景：

建立强大的学生成绩管理功能，方便系统管理员对学生成绩等信息的添加、修改，打印等操作，同时方便学生对自己各科成绩查询，学习的交流。采用计算机进行以上各类信息的管理，必将大大提高工作效率，对学校各类信息的维护及了解起到积极的作用。另外随着计算机网络的持续发展，无论是工作、学习还是生活，将会越来越多地融入到流动的比特中，人们的工作节奏在不断加快，需要快捷的信息来源，并对此作出反应。因此，无论是从目前的现状来看，还是面向未来，一个功能完备的学校管理系统的开发都非常有必要。

【案例分析】

1. 结构体定义

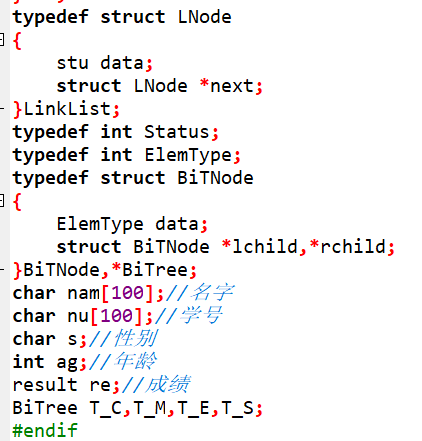
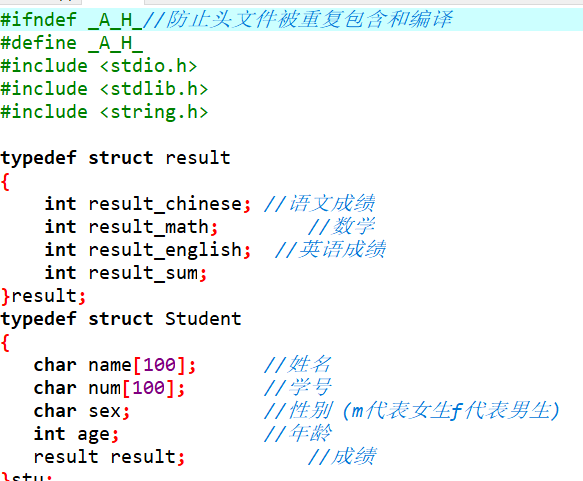




代码中运用了链表和二叉树两种数据结构

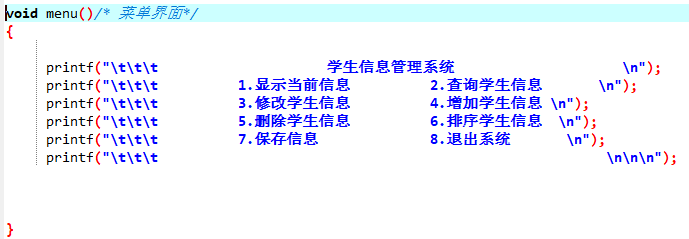
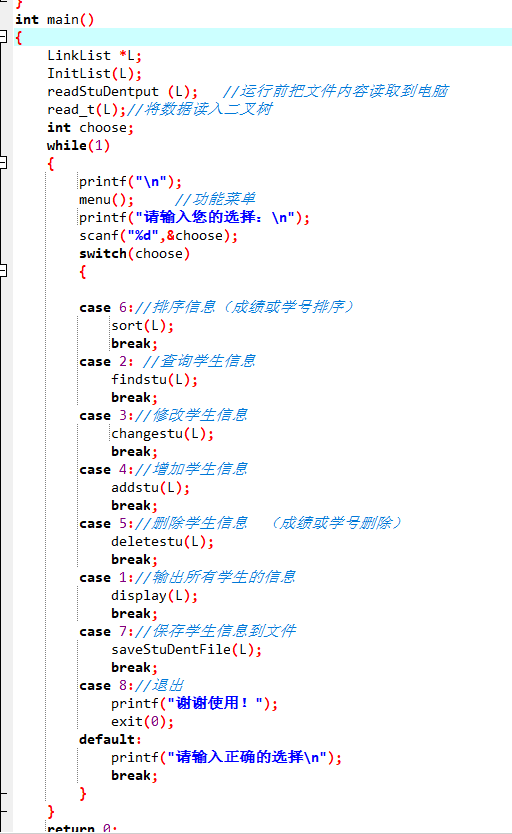
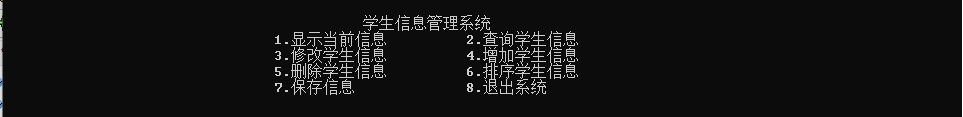
二.各部分的代码：

头文件



三.代码部分：

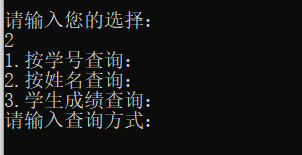
菜单栏



1. 当输入1时；会将student.txt文件内容读入系统内存等。



2.当输入2时；会有两种选项。分别为按学号和按姓名。

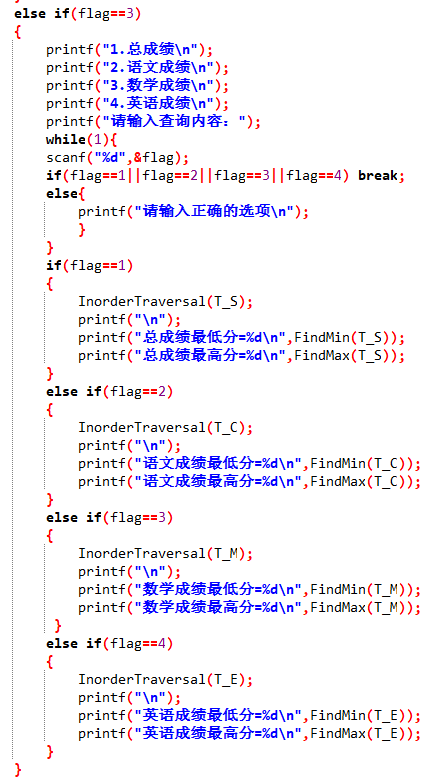


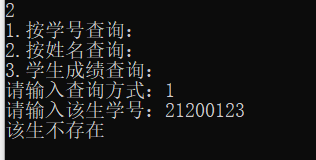
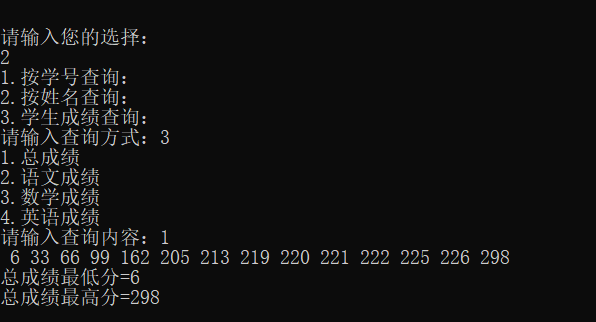
这个是按学号姓名查询部分代码



这里我们以按学生成绩查询为例（这里使用了二叉排序树实现，中序遍历）:

输入正确的学号或姓名即可得到该学生的所有信息。

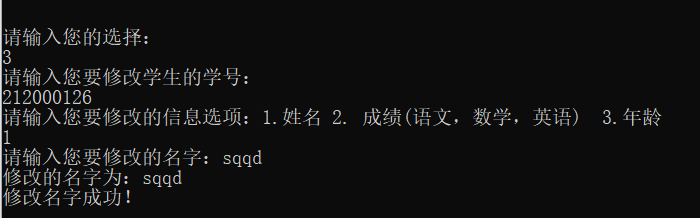
  
如果输出的学号或姓名有误，则会提示该学生不存在。



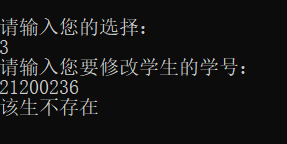
3.当输入3时;会要求你输入修改学生的学号

输入正确的学号后会有选择修改的选项（姓名，成绩和年龄）分别为按键1，2，3。如若修改成功，则会回到初始页面。



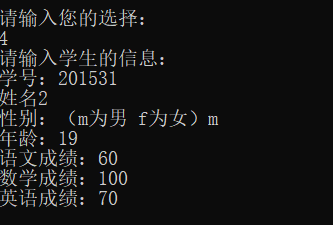


若输入错误的学号，则会提示该学生不存在。

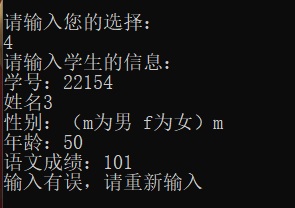


4.当输入4时，系统会要求你输入想要增添学生的学号

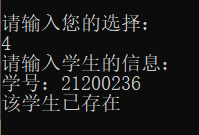
输入想要添加的学号，姓名，性别，年龄，语文数学和英语成绩即为添加成功。



成绩的范围为0~100，当超出100时，系统会显示输入有误

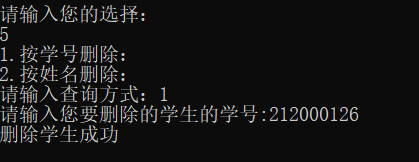


当再次添加成员时，如果出现重复成员，系统会显示该学生已存在。

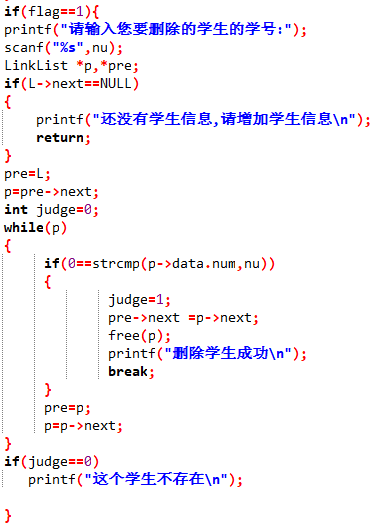


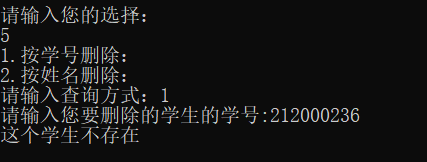
5.当输入5时，会有两种删除方式，一种是按学号删除，一种是按姓名删除。

这里我们按学号删除为例，输入学生学号或姓名即可删除成功。



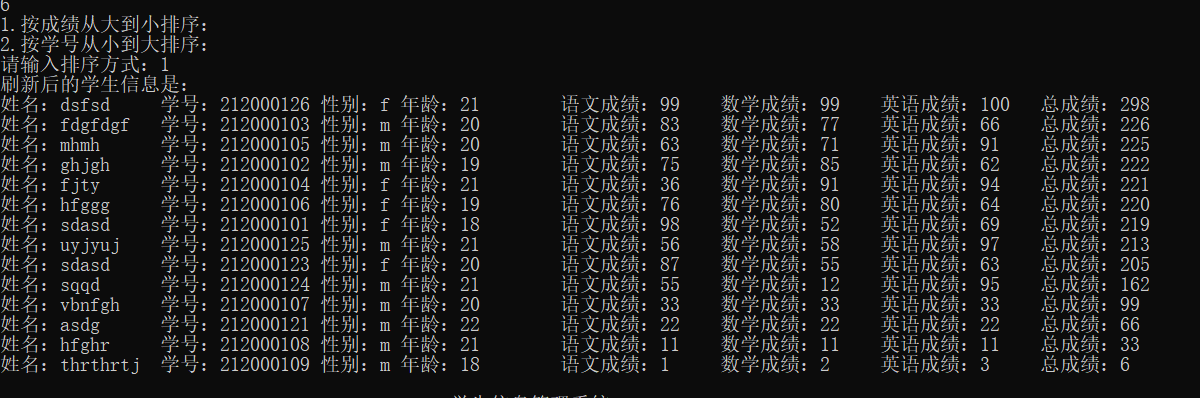
如果输入的学号与已有的学生不匹配，将会提示该学生不存在。





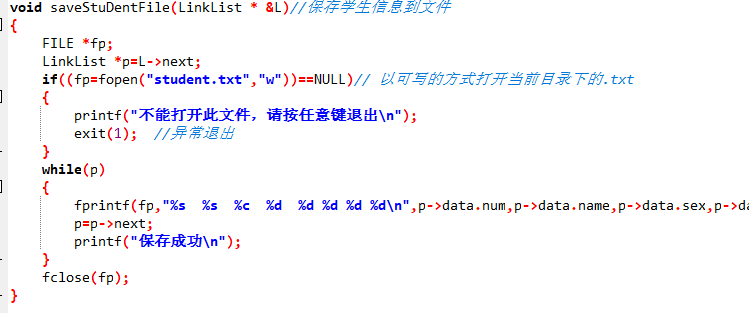
6.当输入6时，会有两种排序方式。分别大是按成绩从到小排序和按学号从小到大排序。（直接交换排序 时间复杂度O（n²））

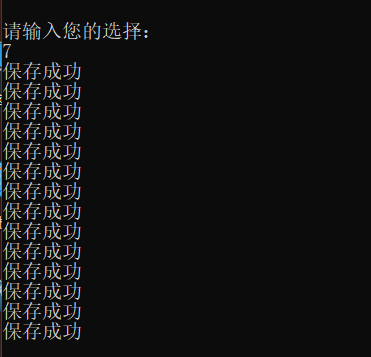
选项其中一种按成绩从大到小排序来作为案例分析。



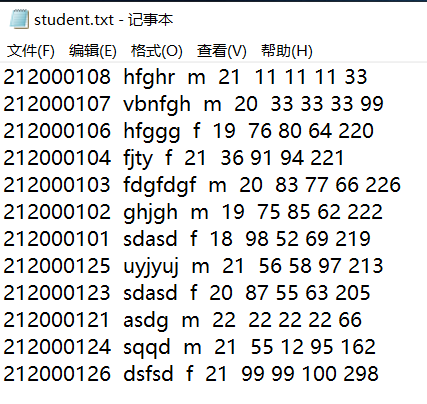


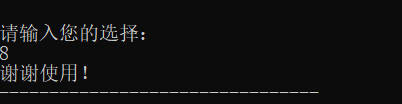
7.当输入7时，会保存以上更改删除添加等操作,会将修改后的内容修改到student.txt文件中。





8.当输入8时，程序结束。





源代码

#include "a.h"

Status Insertree(BiTree &T,ElemType e){

if(T==NULL)

{

T=(BiTree)malloc(sizeof(struct BiTNode));

T->data=e;

T->lchild=NULL;

T->rchild=NULL;

return 1;

}

if(T->data>e) Insertree(T->lchild,e);

else if(T->data<e) Insertree(T->rchild,e);

return 1;

}

int FindMin( BiTree T)//时间复杂度O(log2n)

{

if(T==NULL) return NULL;

BiTree p=T;

while(p&&p->lchild) p=p->lchild;

return p->data;

}

int FindMax( BiTree T)//时间复杂度O(log2n)

{

if(T==NULL) return NULL;

BiTree p=T;

while(p&&p->rchild) p=p->rchild;

return p->data;

}

int InorderTraversal( BiTree T )

{

if(T==NULL) return 0;

InorderTraversal(T->lchild);

printf(" %d",T->data);

InorderTraversal(T->rchild);

}

void menu()/\* 菜单界面\*/

{

printf("\t\t\t 学生信息管理系统 \n");

printf("\t\t\t 1.显示当前信息 2.查询学生信息 \n");

printf("\t\t\t 3.修改学生信息 4.增加学生信息 \n");

printf("\t\t\t 5.删除学生信息 6.排序学生信息 \n");

printf("\t\t\t 7.保存信息 8.退出系统 \n");

printf("\t\t\t \n\n\n");

}

void InitList(LinkList \*&L)//初始化链表

{

L=(LinkList \*)malloc(sizeof(LinkList));

L->next=NULL;

}

void ListInsert(LinkList \*&L,LinkList \*p)//带头节点逆序法

{

LinkList \*q=NULL;

q=L;

p->next=q->next;

q->next=p;

}

void addstu(LinkList \*&L)//增加新的学生

{

printf("请输入学生的信息：\n");

printf("学号：");

scanf("%s",nu);

LinkList \*q=L->next;

while(q!=NULL ) //用于判断学生信息是否已经存在

{

if(0==strcmp(q->data.num,nu))

{

printf("该学生已存在\n");

break;

}

q=q->next;

}

if(q==NULL)

{

LinkList \*p;

InitList(p);

strcpy(p->data.num,nu);

printf("姓名") ;

scanf("%s",nam) ;

strcpy(p->data.name,nam);

printf("性别：（m为男 f为女）");

scanf(" %c",&s);

p->data.sex=s;

printf("年龄：");

scanf("%d",&ag);

p->data.age=ag;

printf("语文成绩：");

scanf("%d",&re.result\_chinese);

while(re.result\_chinese<0||re.result\_chinese>100)

{

printf("输入有误，请重新输入\n");

scanf("%d",&re.result\_chinese);

}

p->data.result.result\_chinese=re.result\_chinese;

printf("数学成绩：");

scanf("%d",&re.result\_math);

while(re.result\_math<0||re.result\_math>100)

{

printf("输入有误，请重新输入\n");

scanf("%d",&re.result\_math);

}

p->data.result.result\_math=re.result\_math;

printf("英语成绩：");

scanf("%d",&re.result\_english);

while(re.result\_english<0||re.result\_english>100)

{

printf("输入有误，请重新输入\n");

scanf("%d",&re.result\_english);

}

p->data.result.result\_english=re.result\_english;

p->data.result.result\_sum=re.result\_chinese+re.result\_math+re.result\_english;

ListInsert(L,p);

}

}

void deletestu(LinkList \*L)//删除学生

{

int flag;

printf("1.按学号删除：\n");

printf("2.按姓名删除：\n");

printf("请输入查询方式：");

while(1){

scanf("%d",&flag);

if(flag==1||flag==2) break;

else{

printf("请输入正确的选项\n");

}

}

if(flag==1){

printf("请输入您要删除的学生的学号:");

scanf("%s",nu);

LinkList \*p,\*pre;

if(L->next==NULL)

{

printf("还没有学生信息,请增加学生信息\n");

return;

}

pre=L;

p=pre->next;

int judge=0;

while(p)

{

if(0==strcmp(p->data.num,nu))

{

judge=1;

pre->next =p->next;

free(p);

printf("删除学生成功\n");

break;

}

pre=p;

p=p->next;

}

if(judge==0)

printf("这个学生不存在\n");

}

else if(flag==2){

printf("请输入您要删除的学生的姓名:");

scanf("%s",nam);

LinkList \*p,\*pre;

if(L->next==NULL)

{

printf("还没有学生信息,请增加学生信息\n");

return;

}

pre=L;

p=pre->next;

int judge=0;

while(p)

{

if(0==strcmp(p->data.name,nam))

{

judge=1;

pre->next =p->next;

free(p);

printf("删除学生成功\n");

break;

}

pre=p;

p=p->next;

}

if(judge==0)

printf("这个学生不存在\n");

}

}

void changestu(LinkList \*L)//修改学生信息

{

int flag=0;

printf("请输入您要修改学生的学号：\n");

scanf("%s",nu);

LinkList \*q=L->next;

while(q!=NULL )

{

if(0==strcmp(q->data.num,nu))

{

flag=1;

printf("请输入您要修改的信息选项：1.姓名 2. 成绩(语文，数学，英语) 3.年龄 \n");

int choose;

scanf("%d",&choose);

switch(choose)

{

case 1:

printf("请输入您要修改的名字：");

scanf("%s",nam);

printf("修改的名字为：%s\n",nam);

strcpy(q->data.name,nam);

printf("修改名字成功！\n");

break;

case 2:

printf("请输入您要修改的成绩：语文 数学 英语");

scanf("%d%d%d",&re.result\_chinese,&re.result\_math,&re.result\_english);

printf("修改的语文成绩为：%d 修改的数学成绩为：%d 修改的英语成绩为：%d\n",re.result\_chinese,re.result\_math,re.result\_english);

re.result\_sum=re.result\_chinese+re.result\_english+re.result\_math;

q->data.result=re;

printf("修改成绩成功！\n");

break;

case 3:

printf("请输入您要修改的年龄：");

scanf("%d",&ag);

printf("修改的年龄为：%d\n",ag);

q->data.age=ag;

printf("修改年龄成功！\n");

break;

default :

printf("请输入正确的选项\n");

break;

}

}

q=q->next;

}

if(flag==0)

{

printf("该生不存在\n");

}

}

void findstu(LinkList \*L)//按学号或者姓名查找学生并输出该生信息

{

int flag;

printf("1.按学号查询：\n");

printf("2.按姓名查询：\n");

printf("3.学生成绩查询：\n");

printf("请输入查询方式：");

while(1){

scanf("%d",&flag);

if(flag==1||flag==2||flag==3) break;

else{

printf("请输入正确的选项\n");

}

}

if(flag==1){

printf("请输入该生学号：");

scanf("%s",nu);

//判断

LinkList \*q=L->next;

while(q!=NULL )

{

if(strcmp(q->data.num,nu)==0)

{

printf("姓名：%s\n",q->data.name);

printf("学号：%s\n",q->data.num);

printf("性别：%c\n",q->data.sex);

printf("年龄：%d\n",q->data.age);

printf("语文成绩：%d\n",q->data.result.result\_chinese);

printf("数学成绩：%d\n",q->data.result.result\_math);

printf("英语成绩：%d\n",q->data.result.result\_english);

printf("总成绩：%d\n",q->data.result.result\_sum);

break;

}

q=q->next;

}

if(q==NULL)

printf("该生不存在\n");

}else if(flag==2){

printf("请输入该生姓名：");

scanf("%s",nam);

LinkList \*q=L->next;

while(q!=NULL)

{

if(strcmp(q->data.name,nam)==0)

{

printf("姓名：%s\n",q->data.name);

printf("学号：%s\n",q->data.num);

printf("性别：%c\n",q->data.sex);

printf("年龄：%d\n",q->data.age);

printf("语文成绩：%d\n",q->data.result.result\_chinese);

printf("数学成绩：%d\n",q->data.result.result\_math);

printf("英语成绩：%d\n",q->data.result.result\_english);

printf("总成绩：%d\n",q->data.result.result\_sum);

break;

}

q=q->next;

}

if(q==NULL)

printf("该生不存在\n");

}

else if(flag==3)

{

printf("1.总成绩\n");

printf("2.语文成绩\n");

printf("3.数学成绩\n");

printf("4.英语成绩\n");

printf("请输入查询内容：");

while(1){

scanf("%d",&flag);

if(flag==1||flag==2||flag==3||flag==4) break;

else{

printf("请输入正确的选项\n");

}

}

if(flag==1)

{

InorderTraversal(T\_S);

printf("\n");

printf("总成绩最低分=%d\n",FindMin(T\_S));

printf("总成绩最高分=%d\n",FindMax(T\_S));

}

else if(flag==2)

{

InorderTraversal(T\_C);

printf("\n");

printf("语文成绩最低分=%d\n",FindMin(T\_C));

printf("语文成绩最高分=%d\n",FindMax(T\_C));

}

else if(flag==3)

{

InorderTraversal(T\_M);

printf("\n");

printf("数学成绩最低分=%d\n",FindMin(T\_M));

printf("数学成绩最高分=%d\n",FindMax(T\_M));

}

else if(flag==4)

{

InorderTraversal(T\_E);

printf("\n");

printf("英语成绩最低分=%d\n",FindMin(T\_E));

printf("英语成绩最高分=%d\n",FindMax(T\_E));

}

}

}

void display(LinkList \*&L)//浏览全部学生信息

{

LinkList \*q=L->next;

if(q==NULL)

{

printf("还没有学生信息，请增加学生信息\n");

return;

}

while(q)

{

printf("姓名：%s\t",q->data.name);

printf("学号：%s\t",q->data.num);

printf("性别：%c\t",q->data.sex);

printf("年龄：%d\t",q->data.age);

printf("语文成绩：%d\t",q->data.result.result\_chinese);

printf("数学成绩：%d\t",q->data.result.result\_math);

printf("英语成绩：%d\t",q->data.result.result\_english);

printf("总成绩：%d\n",q->data.result.result\_sum);

q=q->next;

}

}

//时间复杂度O(n~2) 空间复杂度O(n)

void sort\_1(LinkList \*L)//按学号排序排序 并输出排序后的结果

{

LinkList \*q,\*p,\*r=L->next;

int count=0;

//判断

if(r==NULL)

{

printf("还没有学生信息，请增加学生信息\n");

return;

}

while(r) //两层循环完成排序

{

p=r;

q=r->next;

LinkList \*tmp;//用于排序时暂存节点

InitList(tmp); //借助中间变量 便于交换值

while(q)

{

if(q->data.result.result\_sum > p->data.result.result\_sum) //成绩从大到小

{

//先复制q结点信息到tmp

strcpy(tmp->data.num,q->data.num);

strcpy(tmp->data.name,q->data.name);

tmp->data.sex=q->data.sex;

tmp->data.age=q->data.age;

tmp->data.result=q->data.result;

//再复制p结点信息到q

strcpy(q->data.num,p->data.num);

strcpy(q->data.name,p->data.name);

q->data.sex=p->data.sex;

q->data.age=p->data.age;

q->data.result=p->data.result;

//最后复制exchange结点信息到p

strcpy(p->data.num,tmp->data.num);

strcpy(p->data.name,tmp->data.name);

p->data.sex=tmp->data.sex;

p->data.age=tmp->data.age;

p->data.result=tmp->data.result;

}

q=q->next;

}

r=r->next;

}

printf("刷新后的学生信息是：\n");

display(L);

}

void sort\_2(LinkList \*L)//按学号排序排序 并输出排序后的结果

{

LinkList \*q,\*p,\*t=L->next;

int count=0;

//判断

if(t==NULL)

{

printf("还没有学生信息，请增加学生信息\n");

return;

}

while(t) //两层循环完成排序

{

p=t;

q=t->next;

LinkList \*tmp;//用于排序时暂存节点

InitList(tmp); //借助中间变量 便于交换值

while(q)

{

if(strcmp(q->data.num,p->data.num)<0) //学号从大到小

{

//printf("%d %s %s\n",strcmp(q->data.num,p->data.num),q->data.num,p->data.num);

//先复制q结点信息到tmp

strcpy(tmp->data.num,q->data.num);

strcpy(tmp->data.name,q->data.name);

tmp->data.sex=q->data.sex;

tmp->data.age=q->data.age;

tmp->data.result=q->data.result;

//再复制p结点信息到q

strcpy(q->data.num,p->data.num);

strcpy(q->data.name,p->data.name);

q->data.sex=p->data.sex;

q->data.age=p->data.age;

q->data.result=p->data.result;

//最后复制exchange结点信息到p

strcpy(p->data.num,tmp->data.num);

strcpy(p->data.name,tmp->data.name);

p->data.sex=tmp->data.sex;

p->data.age=tmp->data.age;

p->data.result=tmp->data.result;

}

q=q->next;

}

t=t->next;

}

printf("刷新后的学生信息是：\n");

display(L);

}

void sort(LinkList \*L){

int flag;

printf("1.按成绩从大到小排序：\n");

printf("2.按学号从小到大排序：\n");

printf("请输入排序方式：");

while(1){

scanf("%d",&flag);

if(flag==1||flag==2) break;

else{

printf("请输入正确的选项\n");

}

}

//printf("flag=%d\n",flag);

if (flag==1) sort\_1(L);

else if(flag==2) sort\_2(L);

}

void saveStuDentFile(LinkList \* &L)//保存学生信息到文件

{

FILE \*fp;

LinkList \*p=L->next;

if((fp=fopen("student.txt","w"))==NULL)// 以可写的方式打开当前目录下的.txt

{

printf("不能打开此文件，请按任意键退出\n");

exit(1); //异常退出

}

while(p)

{

fprintf(fp,"%s %s %c %d %d %d %d %d\n",p->data.num,p->data.name,p->data.sex,p->data.age,p->data.result.result\_chinese,p->data.result.result\_math,p->data.result.result\_english,p->data.result.result\_sum);

p=p->next;

printf("保存成功\n");

}

fclose(fp);

}

void readStuDentput (LinkList \*&L) //运行前把文件内容读取到电脑内存

{

FILE \*fp;

fp=fopen("student.txt","rb"); //以只读方式打开当前目录下的.txt

if(fp==NULL)

{

printf("文件不存在\n");

// exit(0); //终止程序

}

int i=0;

while(!feof(fp)) //feof(fp)有两个返回值:如果遇到文件结束，函数feof（fp）的值为非零值，否则为0

{

char nu[100];//学号

char nam[100];//名字

char s;//性别

int ag;//年龄

re;//成绩

fscanf(fp," %s %s %c %d %d %d %d %d %d",nu,nam,&s,&ag,&re.result\_chinese,&re.result\_math,&re.result\_english,&re.result\_sum,&re.result\_sum);

i++;

}

fclose(fp);

FILE \*FP;

FP=fopen("student.txt","rb"); //以只读方式打开当前目录下的.txt

if(FP==NULL)

{

printf("无法打开文件\n");

//exit(0); //终止程序

}

int count=i-1;//记录数据个数

int j=1;

while(!feof(FP))//feof(fp)有两个返回值:如果遇到文件结束，函数feof（fp）的值为非零值，否则为0

{

fscanf(FP,"%s %s %c %d %d %d %d %d",nu,nam,&s,&ag,&re.result\_chinese,&re.result\_math,&re.result\_english,&re.result\_sum);

LinkList \*n=(LinkList \*)malloc(sizeof(LinkList));

strcpy(n->data.num,nu);//把后者的内容拷贝到前者中

strcpy(n->data.name,nam);//把后者的内容拷贝到前者中

n->data.sex=s;

n->data.age=ag;

n->data.result=re;

ListInsert(L,n);//将n节点插入L中

n=n->next;

if(j==count)

break;

j++;

}

fclose(FP); //关闭文件

}

void read\_t(LinkList \*L)

{

LinkList \*p = L->next;

while(p)

{

Insertree(T\_C,p->data.result.result\_chinese);

Insertree(T\_M,p->data.result.result\_math);

Insertree(T\_E,p->data.result.result\_english);

Insertree(T\_S,p->data.result.result\_sum);

p=p->next;

}

}

int main()

{

LinkList \*L;

InitList(L);

readStuDentput (L); //运行前把文件内容读取到电脑

read\_t(L);//将数据读入二叉树

int choose;

while(1)

{

printf("\n");

menu(); //功能菜单

printf("请输入您的选择：\n");

scanf("%d",&choose);

switch(choose)

{

case 6://排序信息（成绩或学号排序）

sort(L);

break;

case 2: //查询学生信息

findstu(L);

break;

case 3://修改学生信息

changestu(L);

break;

case 4://增加学生信息

addstu(L);

break;

case 5://删除学生信息 （成绩或学号删除）

deletestu(L);

break;

case 1://输出所有学生的信息

display(L);

break;

case 7://保存学生信息到文件

saveStuDentFile(L);

break;

case 8://退出

printf("谢谢使用！");

exit(0);

default:

printf("请输入正确的选择\n");

break;

}

}

return 0;

}