**数据结构实验考试试卷**

**姓 名：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 学 号:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**班 级：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 成 绩:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**实验项目名称：链表的实现**

**实验内容**

**利用链表的的基本操作实现简单程序（学生成绩管理程序）。**

**①定义链表的存储结构、**

**②设计链表的相关算法（建立、插入、删除、取数**

**据元素、倒置等）**

**③设计程序并测试。**

**实验代码:**

#include<stdio.h>

#include<stdlib.h>

#include<Windows.h>

typedef struct link

{

char num[20];

float grade[5];

int ID;//元素位置

link \*front;

link \*next;

};

int flag=0;

link \*head;//定义链表头

char project[10][20]={("计算机原理"),("数据结构"),("数字电路"),("英语"),("jsp")};

link \*crate(int mark)

{

link \*data=NULL;//初始化链表头

data=(link \*)malloc(sizeof(link));//强制转换为link类型并开辟空间

if(mark)

{

data->next=NULL;

data->front=NULL;

}

else

{

data->next=head;

data->front=head->front;

head->front=data;//前驱置为data执行将data放到前面

}

data->ID=-1;//初始化学生位置

return data;

}

link \*add(link \*p,int mark)

{

link \*data=NULL;//初始化节点

data=(link \*)malloc(sizeof(link));

if(mark)

//data->next=head;//收尾相接循环链表

data->next=NULL;

else data->next=p->next;//初始化节点接口

data->front=p;//双向链表

data->ID=p->ID;//结构体引用成员可以用->而结构体指针不行，必须->

p->next=data;//->表示直接从指针指向的地址中取出成员，不需要\*转义

return data;

}

void Add\_Message(link \*data,int mark)

{

int i=0,k=0,choose=0,code=0;//初始化

char check=0;

printf("请输入学号");

while((check=getchar())!=EOF)//检测是否输入完毕

{

i=0;//初始化录入字符

if(flag==0)

{

data=crate(mark);//head记录链表头地址并判断是否插入

head=data;//记录头结点

flag++;

}

else

data=add(data,mark);//增加节点并判断是否插入

data->num[i]=check;//录入学号

for(i=1;i<20;)

{

if((data->num[i]=getchar())=='\n')

{

data->num[i]='\0';

break;

}

++i;

if(i>20)

{

printf("学号大小超出限制，重新输入学号");//检测是否超出限制

i=0;

}

}

for(i=0;i<5;i++)

{

printf("%s:",project[i]);

scanf("%f",&data->grade[i]);//录入成绩

}

if(mark)

{

getchar();//防止录入回车影响后续数据录入

++data->ID;

printf("请输入第%d个学生学号(EOF结束输入):",data->ID+2);

}

else

break;

}

}

int Insert\_List(int insert)

{

link \*data;

data=head;

while(data!=NULL)

{

if(insert==1)//检测是否插入链表头

{

flag=0;

Add\_Message(data,0);

data=head;

while(data!=NULL)

{

data->ID++;//先执行操作再链接下一个节点

data=data->next;

}

return 1;

}

else if((data->ID+2)==insert)

{

Add\_Message(data,0);//第二个参数为0代表插入操作

while(data->next!=NULL)

{

data=data->next;//先链接节点再进行操作

data->ID++;//注意该语句放置在操作前和操作后的区别

}

return 1;

}

data=data->next;

}

return 0;

}

int Delete\_List(int del)

{

link \*data;

link \*Del;

data=head;

while(data!=NULL)

{

if(del==1)

{

head=data->next;//等价于head=head->next

head->front=data->front;

Del=data;

while(data!=NULL)

{

data->ID--;

data=data->next;

}

delete Del;

return 1;

}

if((data->ID-2)==del)

{

Del=data->next;

data->next=(data->next)->next;

(data->next)->front=data;

while(data->next!=NULL)

{

data=data->next;

data->ID--;

}

delete Del;//从内存删除数据

return 1;

}

data=data->next;

}

return 0;

}

void Check()

{

link \*data;

data=head;

while(data!=NULL)

{

printf("%s\n",data->num);

for(int j=0;j<5;j++)

printf("%f\n",data->grade[j]);

printf("%d\n",data->ID);

data=data->next;

}

}

void Back\_List(){

}

main()

{

int i=0,k=0,choose=-1,code=0;

link \*data;//初始化

Add\_Message(data,1);//第二个参数为1代表输入链表数据操作

system("cls");

while(choose!=0)

{

printf("输入您想进行的操作:0.退出菜单1.插入2.删除3.逆序\n");

scanf("%d",&choose);

switch(choose)

{

case 1: printf("请输入插入位置:");

scanf("%d",&k);

code=Insert\_List(k);

system("cls");

if(code==0)

printf("插入位置有误\n");

else if(code==1)

printf("插入成功\n");

else printf("未知错误\n");

getchar();

break;

case 2: printf("输入需要删除信息的学生:");

scanf("%d",&k);

code=Delete\_List(k);

system("cls");

if(code=0)

printf("不存在该学生\n");

else if(code=1)

printf("删除成功\n");

else printf("未知错误\n");

break;

case 3: //code=Back\_List(k);

system("cls");

if(code==-1)

printf("逆序失败\n");

else printf("逆序成功\n",code);

break;

//case 4:break;

//case 5:break;

}

}

Check();

}