

**课 程 实 验 报 告**

**课程名称： C语言程序设计实验**

**专业班级： 网络空间安全2105班**

**学 号： U202112149**

**姓 名： 李翔**

**指导教师： 王同洋**

**报告日期： 2021.12.24**

**网络空间安全学院**

# 实验7 结构与联合实验

**一、实验目的**

1．通过实验，熟悉和掌握结构的说明和引用、结构的指针、结构数组、以及函数中使用结构的方法。

2．通过实验，掌握动态储存分配函数的用法，掌握自引用结构，单向链表的创建、遍历、结点的增删、查找等操作。

3．了解字段结构和联合的用法。

**二、实验题目及要求**

### 7.1.1．表达式求值的程序验证题

设有说明：

char u[]="UVWXYZ";

char v[]="xyz";

struct T{

int x;

char c;

char \*t;

}a[]={{11,ˊAˊ,u},{100, ˊBˊ,v}},\*p=a;

请先自己计算下面表达式的值，然后通过编程计算来加以验证。(各表达式相互无关)

|  |  |  |  |
| --- | --- | --- | --- |
| **序号** | **表达式** | **计算值** | **验证值** |
| 1 | (++p)->x | 100 | 100 |
| 2 | p++,p->c | ‘B’ | ‘B’ |
| 3 | \*p++->t,\*p->t | ‘x’ | ‘x’ |
| 4 | \*(++p)->t | ‘x’ | ‘x’ |
| 5 | \*++p->t | ‘V’ | ‘V’ |
| 6 | ++\*p->t | ‘V’ | ‘V’ |

### 7.2.2．源程序修改替换题

给定一批整数，以0作为结束标志且不作为结点，将其建成一个先进先出的链表，先进先出链表的指头指针始终指向最先创建的结点（链头），先建结点指向后建结点，后建结点始终是尾结点。

1. 源程序中存在什么样的错误（先观察执行结果）？对程序进行修改、调试，使之能够正确完成指定任务。

源程序如下：

#include "stdio.h"

#include "stdlib.h"

struct s\_list{

int data; /\* 数据域 \*/

struct s\_list \*next; /\* 指针域 \*/

} ;

void create\_list (struct s\_list \*headp,int \*p);

void main(void)

{

struct s\_list \*head=NULL,\*p;

int s[]={1,2,3,4,5,6,7,8,0}; /\* 0为结束标记 \*/

create\_list(head,s); /\* 创建新链表 \*/

p=head; /\*遍历指针p指向链头 \*/

while(p){

printf("%d\t",p->data); /\* 输出数据域的值 \*/

p=p->next; /\*遍历指针p指向下一结点 \*/

}

printf("\n");

}

void create\_list(struct s\_list \*headp,int \*p)

{

struct s\_list \* loc\_head=NULL,\*tail;

if(p[0]==0) /\* 相当于\*p==0 \*/

;

else { /\* loc\_head指向动态分配的第一个结点 \*/

loc\_head=(struct s\_list \*)malloc(sizeof(struct s\_list));

loc\_head->data=\*p++; /\* 对数据域赋值 \*/

tail=loc\_head; /\* tail指向第一个结点 \*/

while(\*p){ /\* tail所指结点的指针域指向动态创建的结点 \*/

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

tail=tail->next; /\* tail指向新创建的结点 \*/

tail->data=\*p++; /\* 向新创建的结点的数据域赋值 \*/

}

tail->next=NULL; /\* 对指针域赋NULL值 \*/

}

headp=loc\_head; /\* 使头指针headp指向新创建的链表 \*/

}

错误：函数create\_list参数不对，应该是个双重指针，如果只是一个一重指针，传入的是head所指向的数据的地址，并非head本身地址，无法起到对head指向链表头部的作用所以在函数声明中将\*headp换成\*\*headp，调用函数的时候head改为&head，函数体内的headp改成\*headp即可。

1. 程序源代码：

#include "stdio.h"

#include "stdlib.h"

struct s\_list

{

int data; /\* 数据域 \*/

struct s\_list \*next; /\* 指针域 \*/

} ;

void create\_list (struct s\_list \*\*headp,int \*p);

int main(void)

{

struct s\_list \*head=NULL,\*p;

int s[]={1,2,3,4,5,6,7,8,0}; /\* 0为结束标记 \*/

create\_list(&head,s); /\* 创建新链表 \*/

p=head; /\*遍历指针p指向链头 \*/

while(p)

{

printf("%d\t",p->data); /\* 输出数据域的值 \*/

p=p->next; /\*遍历指针p指向下一结点 \*/

}

printf("\n");

}

void create\_list(struct s\_list \*\*headp,int \*p){

struct s\_list \* loc\_head=NULL,\*tail;

if(p[0]==0);/\* 相当于\*p==0 \*/

else{

loc\_head=(struct s\_list \*)malloc(sizeof(struct s\_list));/\* loc\_head指向动态分配的第一个结点 \*/

loc\_head->data=\*p++; /\* 对数据域赋值 \*/

tail=loc\_head; /\* tail指向第一个结点 \*/

while(\*p){ /\* tail所指结点的指针域指向动态创建的结点 \*/

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

tail=tail->next; /\* tail指向新创建的结点 \*/

tail->data=\*p++; /\* 向新创建的结点的数据域赋值 \*/

}

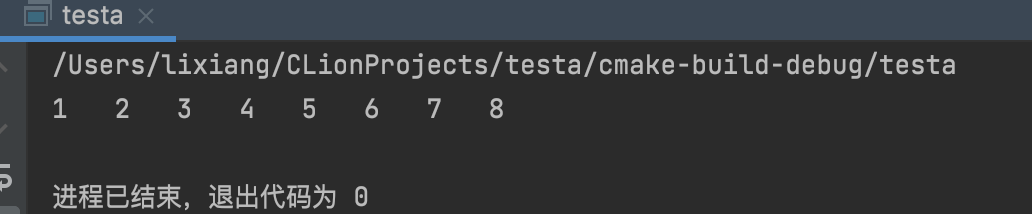
tail->next=NULL; /\* 对指针域赋NULL值 \*/

}

\*headp=loc\_head; /\* 使头指针headp指向新创建的链表 \*/

}

3.程序运行结果：



1. 修改替换create\_list函数，将其建成一个后进先出的链表，后进先出链表的头指针始终指向最后创建的结点（链头），后建结点指向先建结点，先建结点始终是尾结点。

程序源代码：

#include "stdio.h"

#include "stdlib.h"

struct s\_list

{

int data; /\* 数据域 \*/

struct s\_list \*next; /\* 指针域 \*/

} ;

void create\_list (struct s\_list \*\*headp,int \*p);

int main(void)

{

struct s\_list \*head=NULL,\*p;

int s[]={1,2,3,4,5,6,7,8,0}; /\* 0为结束标记 \*/

create\_list(&head,s); /\* 创建新链表 \*/

p=head; /\*遍历指针p指向链头 \*/

while(p)

{

printf("%d\t",p->data); /\* 输出数据域的值 \*/

p=p->next; /\*遍历指针p指向下一结点 \*/

}

printf("\n");

}

void create\_list(struct s\_list \*\*headp,int \*p){

struct s\_list \* loc\_head=NULL,\*tail,\*temp;

if(p[0]==0);/\* 相当于\*p==0 \*/

else{

tail=(struct s\_list \*)malloc(sizeof(struct s\_list));

tail->data=\*p++; /\* 对数据域赋值 \*/

loc\_head=tail;/\*使头指针和尾指针相等\*/

tail->next=NULL;/\* 对指针域赋NULL值 \*/

while(\*p)

{

temp=loc\_head;/\*选择一个中间指针记录未申请结点的loc\_head的值\*/

loc\_head=(struct s\_list \*)malloc(sizeof(struct s\_list));/\*申请新结点\*/

loc\_head->data=\*p++; /\* 向新创建的结点的数据域赋值 \*/

loc\_head->next=temp;/\*使loc\_head的下一个等于中间指针，相当于反向指回来\*/

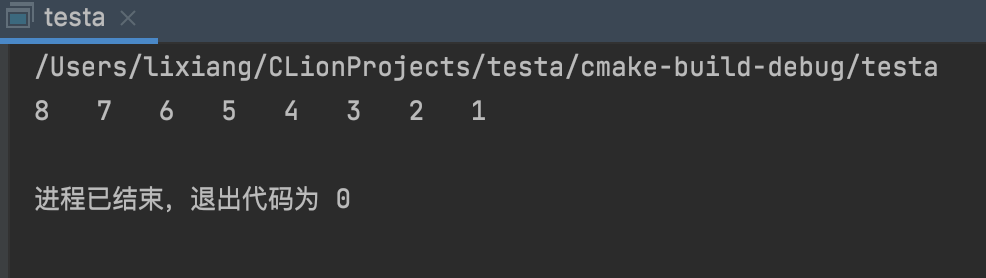
}

}

\*headp=loc\_head; /\* 使头指针headp指向新创建的链表 \*/

}

程序运行截图：



### 7.2.3头歌实验

实验7-1：

本关任务：用单向链表建立一张班级成绩单，包括每个学生的学号、姓名、英语、高等数学、普通物理、C语言程序设计四门课程的成绩。用菜单实现下列功能：

① 输入每个学生的各项信息。

② 输出每个学生的各项信息。

③ 修改指定学生的指定数据项的内容。

④ 统计每个同学的平均成绩（保留2位小数）。

⑤ 输出各位同学的学号、姓名、四门课程的总成绩和平均成绩。序源代码：

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

struct marklist

{

char name[10];

int studynumber,english,math,physics,cprogramming;

struct marklist\*next;

};

void input(struct marklist \*\*headp,int n)/\*创建链表读入数据的函数,此处使用双重指针目的是直接改变head指针所指向地址\*/

{

struct marklist \*createhead,\*tail;

createhead=(struct marklist\*) malloc(sizeof (struct marklist));

scanf("%d%s%d%d%d%d",&createhead->studynumber,createhead->name,&createhead->english,&createhead->math,&createhead->physics,&createhead->cprogramming);

tail=createhead;

for(int i=0;i<n-1;i++)

{

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

tail=tail->next;

scanf("%d%s%d%d%d%d",&tail->studynumber,tail->name,&tail->english,&tail->math,&tail->physics,&tail->cprogramming);

}

tail->next=NULL;

\*headp=createhead;

}

void output(struct marklist \*headp)

{

struct marklist \*p=headp;

while(p)

{

printf("%d %s %d %d %d %d\n",p->studynumber,p->name,p->english,p->math,p->physics,p->cprogramming);

p=p->next;

}

}

void revise(struct marklist \*headp)

{

struct marklist \*p1=headp;

int studynumber1,i;

scanf("%d",&studynumber1);

while(p1)

{

if(studynumber1==p1->studynumber)

{

scanf("%d",&i);

if(i==0)

{

char afterrevise[10];

scanf("%s",afterrevise);

strcpy(p1->name,afterrevise);

}

else if(i==1)

{

scanf("%d",&p1->english);

}

else if(i==2)

{

scanf("%d",&p1->math);

}

else if(i==3)

{

scanf("%d",&p1->physics);

}

else if(i==4)

{

scanf("%d",&p1->cprogramming);

}

printf("%d %s %d %d %d %d\n",p1->studynumber,p1->name,p1->english,p1->math,p1->physics,p1->cprogramming);

}

p1=p1->next;

}

}

void average(struct marklist \*headp)

{

float average;

int sum;

struct marklist \*p2=headp;

while(p2)

{

sum=0;

sum+=p2->english;

sum+=p2->math;

sum+=p2->physics;

sum+=p2->cprogramming;

average=(float )sum/4;

printf("%d %s %.2f\n",p2->studynumber,p2->name,average);

p2=p2->next;

}

}

void outputall(struct marklist \*headp)

{

float average;

int sum;

struct marklist \*p2=headp;

while(p2)

{

sum=0;

sum+=p2->english;

sum+=p2->math;

sum+=p2->physics;

sum+=p2->cprogramming;

average=(float )sum/4;

printf("%d %s %d %.2f\n",p2->studynumber,p2->name,sum,average);

p2=p2->next;

}

}

int main()

{

struct marklist \*head=NULL,\*tail;

int choice,n;

while(1)

{

do

{

scanf("%d",&choice);

}while(choice<0||choice>5);

switch (choice)

{

case 1:

scanf("%d",&n);

input(&head,n);

printf("完成了%d位同学的成绩输入。\n",n);

break;

case 2:

output(head);

break;

case 3:

revise(head);

break;

case 4:

average(head);

break;

case 5:

outputall(head);

break;

default: goto down;

}

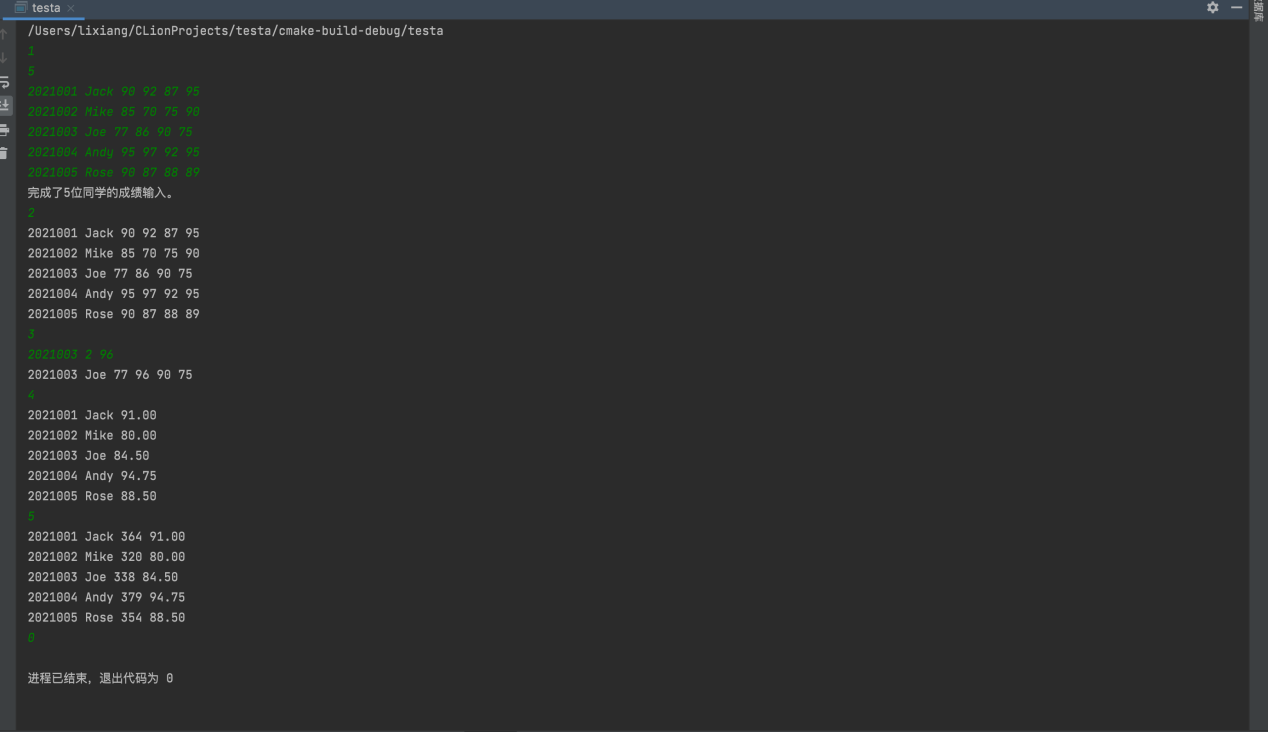
}

down:

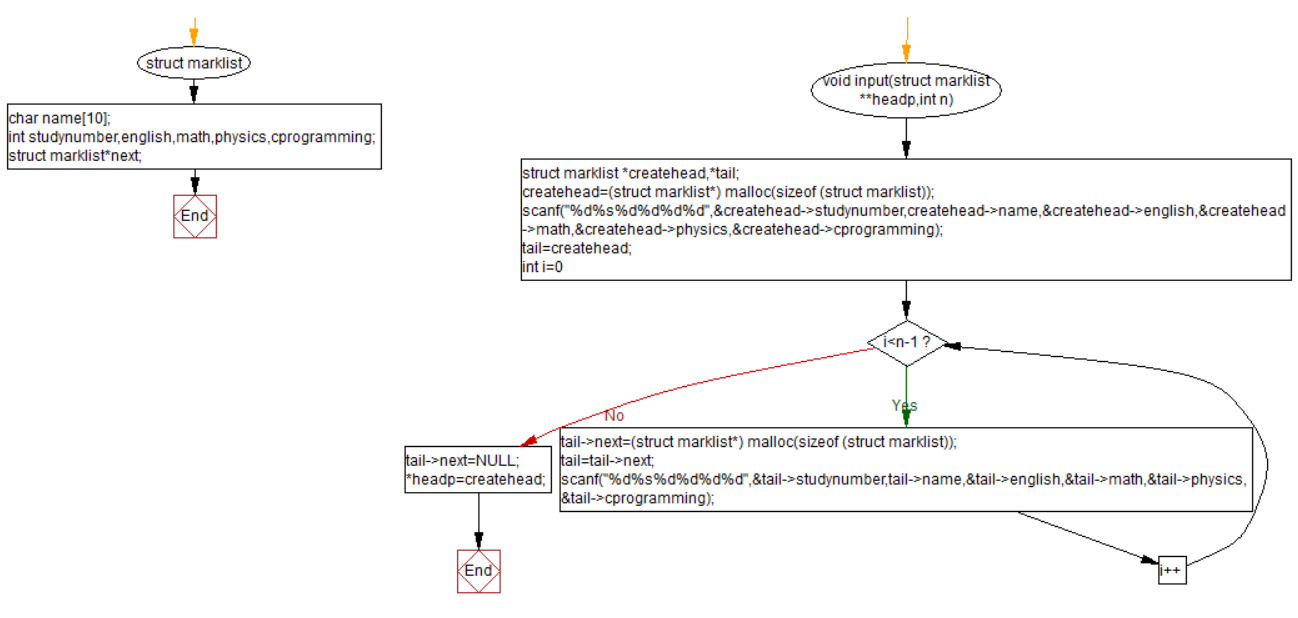
return 0;

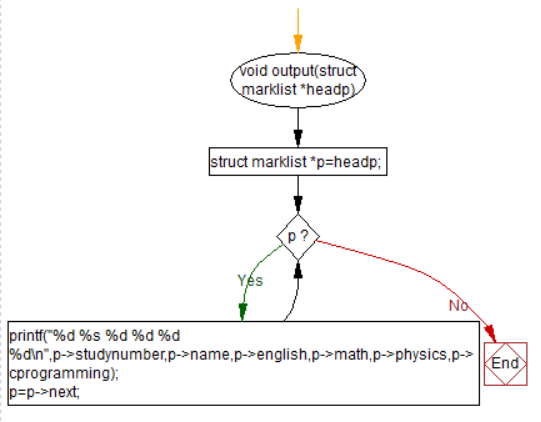
}

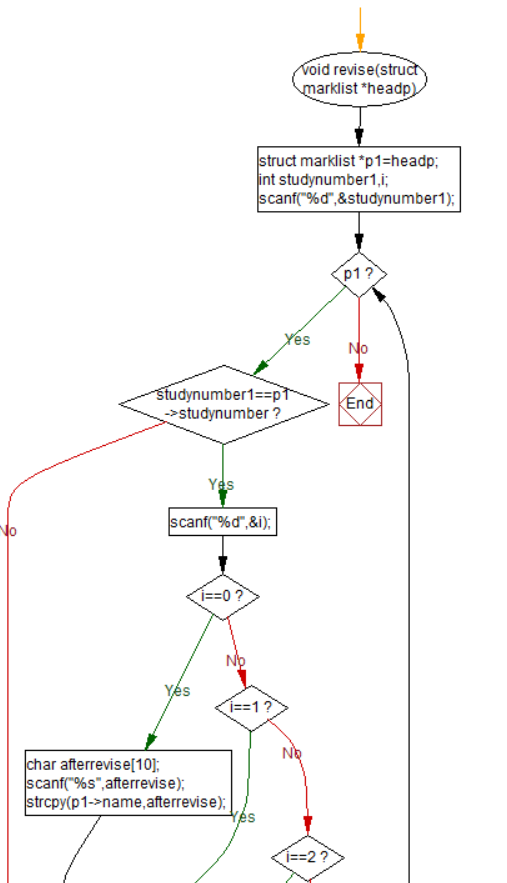
程序运行截图：

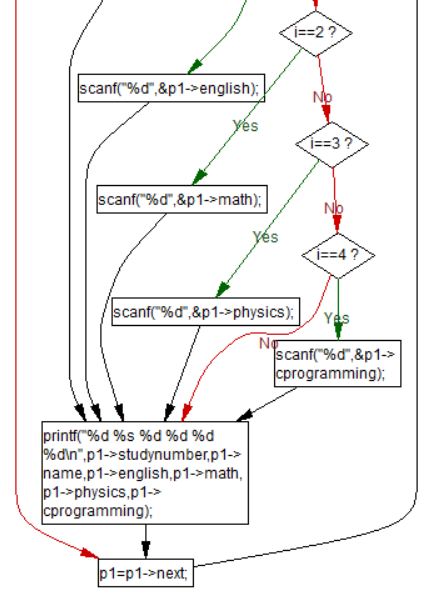


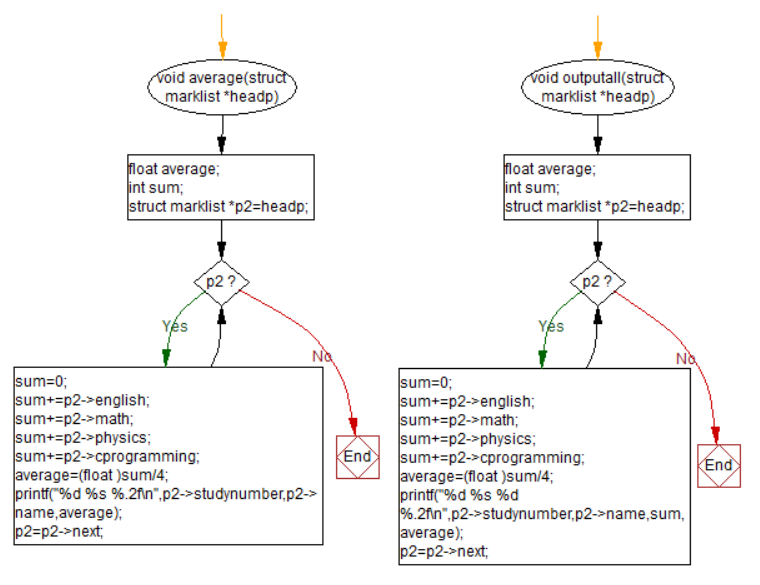
程序流程图：

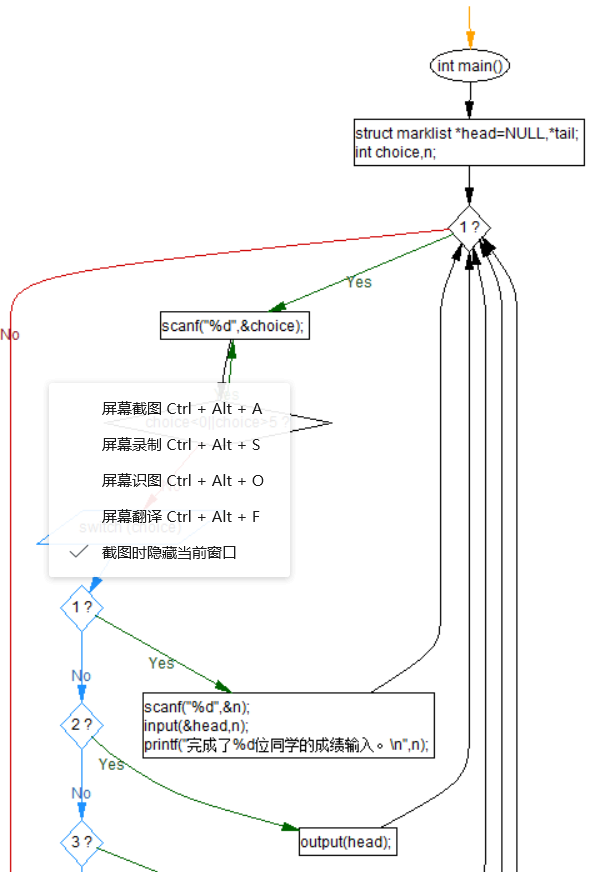


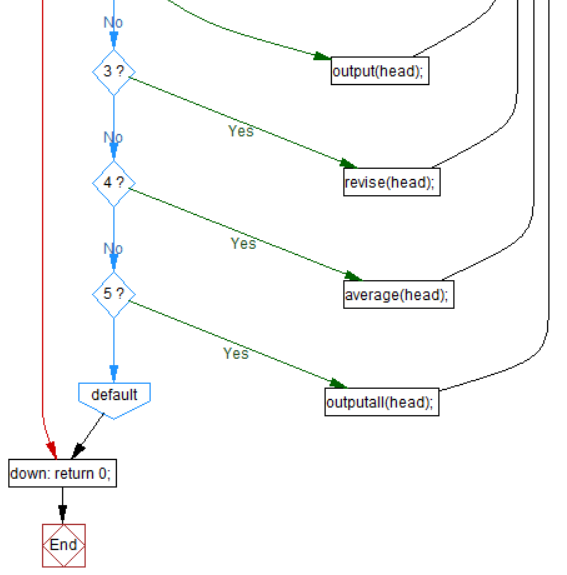












实验7-2

本关任务：对程序设计题第（1）题的程序，⑥增加按照平均成绩进行升序排序的函数，写出用交换结点数据域的方法升序排序的函数，排序可用选择法或冒泡法。

程序源代码（此处由于是上题的改变所以只给出函数）:

void averagesort(struct marklist \*head)

{

struct marklist \*p1=head,\*p2;

int len=0,i,j,temp;

float temp1;

char temp2[10];

while (p1){len++;p1=p1->next;}

for(i=0,p1=head;i<len-1;i++,p1=p1->next)

{

for(j=i+1,p2=p1->next;j<len;j++,p2=p2->next)

{

if(p1->average>p2->average)

{

temp=p1->studynumber;p1->studynumber=p2->studynumber;p2->studynumber=temp;

temp=p1->english;p1->english=p2->english;p2->english=temp;

temp=p1->math;p1->math=p2->math;p2->math=temp;

temp=p1->physics;p1->physics=p2->physics;p2->physics=temp;

temp=p1->cprogramming;p1->cprogramming=p2->cprogramming;p2->cprogramming=temp;

temp1=p1->average;p1->average=p2->average;p2->average=temp1;

temp=p1->sum;p1->sum=p2->sum;p2->sum=temp;

strcpy(temp2,p1->name);strcpy(p1->name,p2->name);strcpy(p2->name,temp2);

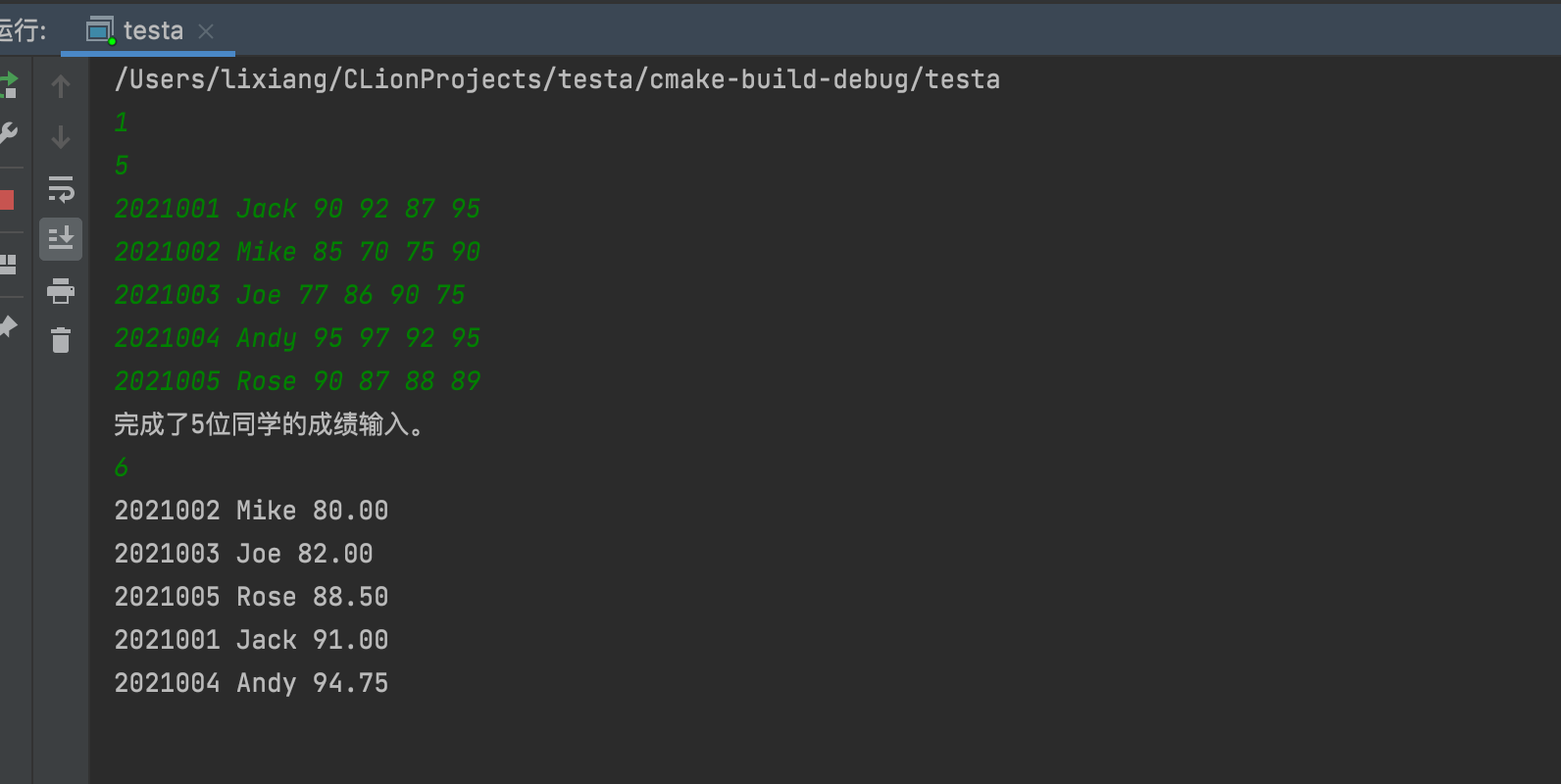
}

}

}

}

程序运行截图:



实验7-3

本关任务：对程序设计题第（2）题，进一步写出用交换结点指针域的方法升序排序的函数。

程序源代码（同样只给出函数）：

void averagesort()

{

struct marklist \*prior1,\*prior2,\*p1,\*p2,\*t,\*p3;

p1=head;

for(p2=p1->next,prior2=p1;p2!=NULL;prior2=p2,p2=p2->next)

if(p1->average>p2->average)

{

prior2->next=p1; t=p1->next;

p1->next=p2->next; head=p2;

p2->next=t; p1=p2;

}

for(prior1=head,p1=prior1->next;p1->next!=NULL;prior1=p1,p1=p1->next)

{

for(p2=p1->next,prior2=p1;p2!=NULL;prior2=p2,p2=p2->next)

{

if(p1->average>p2->average)

{

t=p2->next; prior1->next=p2;

prior2->next=p1;p2->next=p1->next;

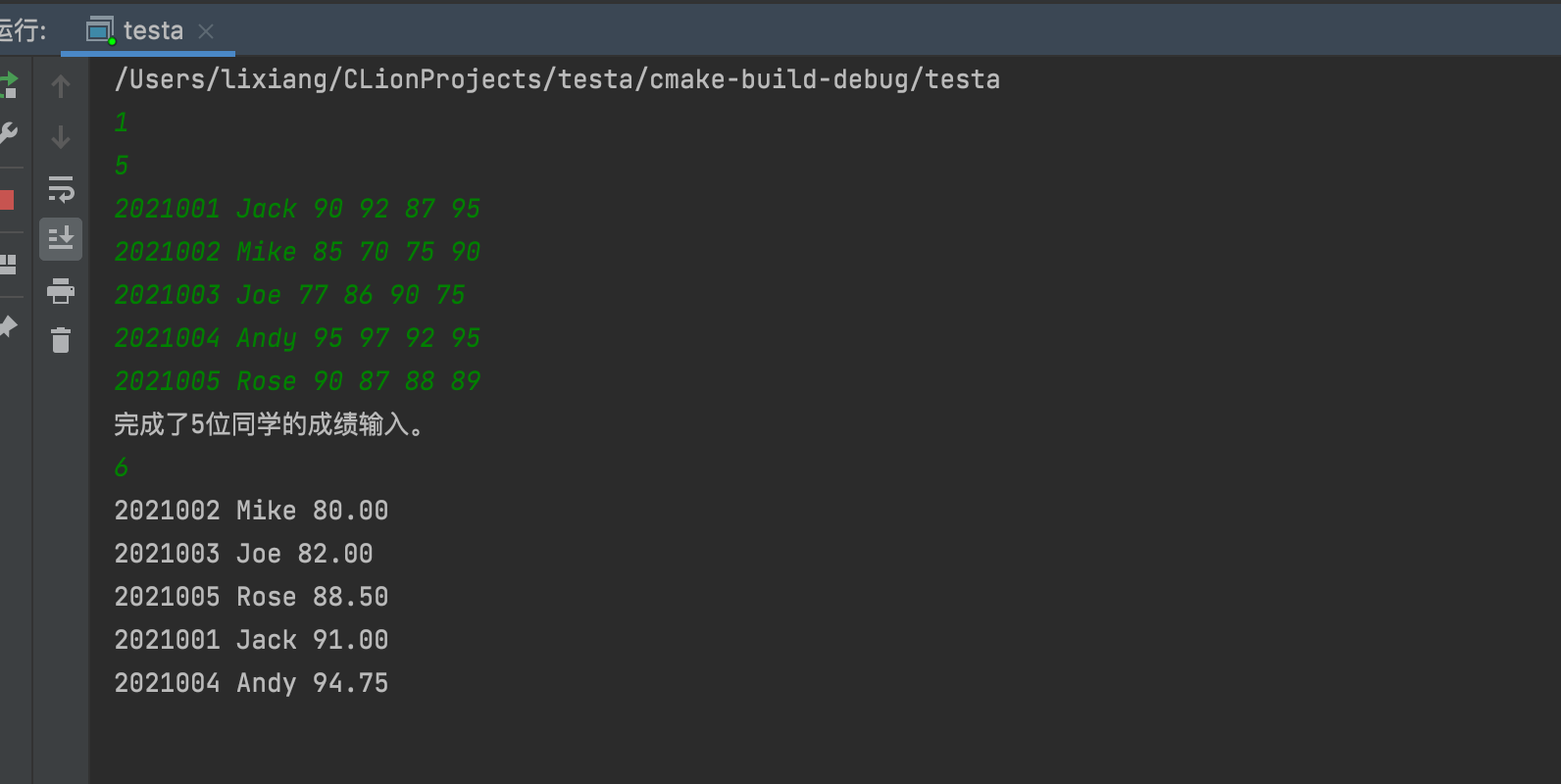
p1->next=t; p1=p2;

}

}

}

程序运行截图：



**7.3实验小结**

通过本次的实验：

我学会了链表的创建和链表的升序降序，可以通过交换指针域和交换数据域来实现，也学会了创建先进后出和先进先出链表，掌握了动态储存分配函数的用法，掌握自引用结构，单向链表的创建、遍历、结点的增删、查找等操作。

还有菜单选项可以用switch-case语句实现，非常的方便。