

**课 程 实 验 报 告**

**课程名称： C语言程序设计实验**

**专业班级： 软件工程2003班**

**学 号： U202010783**

**姓 名： 刘铭宸**

**指导教师： 唐赫**

**报告日期： 2020.12.21**

**软件工程**

**目 录**

[**7 结构与联合实验**](#_Toc404837943) **1**

[7.1 实验目的 1](#_Toc404837921)

[7.2 实验内容 1](#_Toc404837922)

[7.3 实验小结](#_Toc404837923) 24

# 7 结构与联合试验

**7.1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。

## 7.2 实验内容

**7.2.1 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 |  |  |
| 2 | p++,p->c |  |  |
| 3 | \*p++->t,\*p->t |  |  |
| 4 | \*(++p)->t |  |  |
| 5 | \*++p->t |  |  |
| 6 | ++\*p->t |  |  |

**解答：**

|  |  |  |  |
| --- | --- | --- | --- |
| **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 |

(1)进行++p运算后，p指向{100,’B’,v}，此时p->x为100。

(2)首先进行p++运算，p指向{100,’B’,v}，此时p->c为B。

(3)首先进行\*p++->t后，p指向{100,’B’,v}，p->t指向字符数组v的首地址，此时\*p->t为v的首元素x。

(4)首先进行++p运算，p指向{100,’B’,v}，此时\*p->t为x。

(5)首先进行p->t，p->t指向u数组的首地址，++p->t后指向u中第二个元素的地址，故\*++p->t为V。

(6)p->t指向u数组的首地址，故\*p->t为U，进行++运算后结果为字符U的ASCII码加一，得到字符V。

**7.2.2 源程序修改替换**

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 \*/

}

**解答：**

在函数creat\_list中，由于传入的参数是在main函数中创建的头指针，所以仅仅对形参进行了创建链表的操作，并不能对实参进行改变，所以应该传入头指针的地址，使用指针传参。

#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 \*/

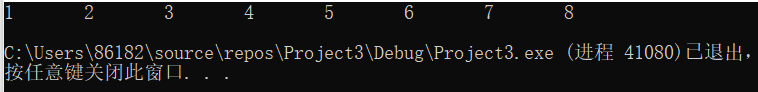
}

**测试：**

（a） 测试数据：

1,2,3,4,5,6,7,8,0

（b） 对应测试数据的运行结果截图



（2）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。

**解答：**

首先建立尾结点tail，将第一个元素作为尾结点，并让头指针指向tail，之后的元素进行表头插入。每收到一个非0元素，建立新的结点，使其next域指向头指针，之后再更新头指针，使其指向该结点。

#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 {

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

tail->data = \*p++;

\*headp = tail;

tail->next = NULL;

while (\*p) {

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

loc\_head->next = \*headp;

loc\_head->data = \*p++;

\*headp = loc\_head;

}

}

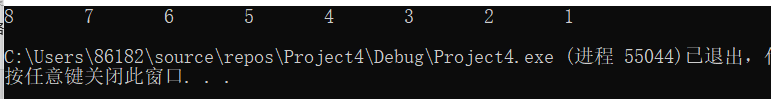
}

**测试：**

（a） 测试数据：

1,2,3,4,5,6,7,8,0

（b） 对应测试数据的运行结果截图



**7.2.3 程序设计**

（1）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。

**解答：**

1）解题思路

将每个学生定义成一个结点，num、name、score数组分别存放学生的学号、姓名和四门课程的成绩，avescore存放学生的平均成绩。按照题目要求，分别设计了CreatNode函数，根据输入的学生信息建立单链表并计算出平均成绩赋值给avescore；PrintNode函数，用来输出每个学生的各项信息；ChangeNode函数，根据输入的学号查找该学生对应的结点，对指定数据项进行修改；AveNode函数，输出每个学生的平均成绩；SumNode函数，计算每个学生的总成绩。

2）源程序清单

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

typedef struct Node {

    char num[100], name[100];

    int score[4];

    double avescore;

    struct Node\* next;

}Node, \* CNode;

void CreatNode(CNode \*head) {

    int qua, i, j;

    scanf("%d", &qua);

    CNode p , tail;

    p = (CNode)malloc(sizeof(Node));

    \*head = p;

    for (i = 0; i < qua; i++) {

        tail = (CNode)malloc(sizeof(Node));

        scanf("%s", &tail->num);

        scanf("%s", &tail->name);

        for (j = 0; j < 4; j++)

            scanf("%d", &tail->score[j]);

        tail->avescore = (tail->score[0] + tail->score[1] + tail->score[2] + tail->score[3]) / 4.0;

        p->next = tail;

        p = tail;

    }

    tail->next = NULL;

    printf("完成了%d位同学的成绩输入。\n",qua);

}

void PrintNode(CNode\* head) {

    CNode p = \*head;

    while (p->next) {

        printf("%s %s ", p->next->num, p->next->name);

        for (int i = 0; i < 4; i++)

            printf("%d ", p->next->score[i]);

        p = p->next;

        printf("\n");

    }

}

void ChangeNode(CNode\* head) {

    CNode p = \*head;

    char num[100];

    int k = -1;

    scanf("%s", &num);

    while (p->next) {

        if (!strcmp(p->next->num, num)) {

            scanf("%d", &k);

            break;

        }

        p = p->next;

    }

    if (k == 0)

        scanf("%s", &p->next->name);

    if (k > 0) {

        scanf("%d", &p->next->score[k - 1]);

        p->next->avescore = (p->next->score[0] + p->next->score[1] + p->next->score[2] + p->next->score[3]) / 4.0;

    }

    printf("%s %s ", p->next->num, p->next->name);

    for (int i = 0; i < 4; i++)

        printf("%d ", p->next->score[i]);

    printf("\n");

}

void AveNode(CNode\* head) {

    CNode p = \*head;

    while(p->next) {

        printf("%s %s %.2lf\n", p->next->num, p->next->name, p->next->avescore);

        p = p->next;

    }

}

void SumNode(CNode\* head) {

    CNode p = \*head;

    while (p->next) {

        printf("%s %s ", p->next->num, p->next->name);

        double sum = 0;

        for (int i = 0; i < 4; i++)

            sum += p->next->score[i];

        printf("%.0lf %.2lf\n", sum, p->next->avescore);

        p = p->next;

    }

}

int main(void) {

    int n;

    CNode head = NULL;

    while ((scanf("%d", &n) != EOF)) {

        if (n == 0)break;

        switch (n) {

        case 1:CreatNode(&head); break;

        case 2:PrintNode(&head); break;

        case 3:ChangeNode(&head); break;

        case 4:AveNode(&head); break;

        case 5:SumNode(&head); break;

        }

    }

}

3）测试

(a) 测试数据：

**测试输入1：**

1

5

2021001 Jack 90 92 87 95

2021002 Mike 85 70 75 90

2021003 Joe 77 86 90 75

2021004 Andy 95 97 92 95

2021005 Rose 90 87 88 89

**预期输出1：**

完成了5位同学的成绩输入。

**测试输入2：**

1

5

2021001 Jack 90 92 87 95

2021002 Mike 85 70 75 90

2021003 Joe 77 86 90 75

2021004 Andy 95 97 92 95

2021005 Rose 90 87 88 89

3

2021003 0 Joan

**预期输出2：**

完成了5位同学的成绩输入。

2021003 Joan 77 86 90 75

**测试输入3：**

1

5

2021001 Jack 90 92 87 95

2021002 Mike 85 70 75 90

2021003 Joe 77 86 90 75

2021004 Andy 95 97 92 95

2021005 Rose 90 87 88 89

3

2021003 2 96

**预期输出3：**

完成了5位同学的成绩输入。

2021003 Joan 77 96 90 75

**测试输入4：**

1

5

2021001 Jack 90 92 87 95

2021002 Mike 85 70 75 90

2021003 Joe 77 86 90 75

2021004 Andy 95 97 92 95

2021005 Rose 90 87 88 89

4

**预期输出4：**

完成了5位同学的成绩输入。

2021001 Jack 91.00

2021002 Mike 80.00

2021003 Joan 84.50

2021004 Andy 94.75

2021005 Rose 88.50

**测试输入5：**

1

5

2021001 Jack 90 92 87 95

2021002 Mike 85 70 75 90

2021003 Joe 77 86 90 75

2021004 Andy 95 97 92 95

2021005 Rose 90 87 88 89

5

**预期输出5：**

2021001 Jack 364 91.00

2021002 Mike 320 80.00

2021003 Joan 338 84.50

2021004 Andy 379 94.75

2021005 Rose 354 88.50

完成了5位同学的成绩输入。

**测试输入6：**

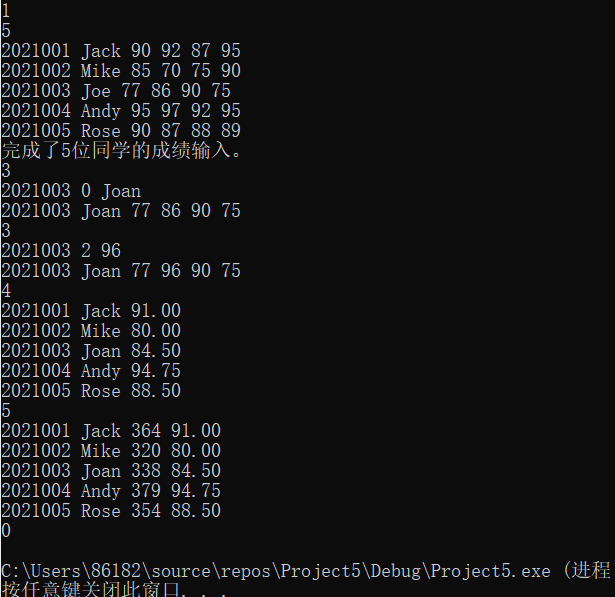
0

**预期输出6：**

（退出）

注意：以上用例需完整的进行，整体前后次序不得颠倒。

(b) 对应测试数据的运行结果截图



（2）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。

**解答：**

1. 解题思路：

增加一个SortNode函数，使用冒泡排序，定义一指针p指向头结点，如果p->next->avescore大于p->next->next->avescore，则对这两个结点的所有数据域进行交换。

2）程序清单

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

typedef struct Node {

    char num[100], name[100];

    int score[4];

    double avescore;

    struct Node\* next;

}Node, \* CNode;

void CreatNode(CNode \*head) {

    int qua, i, j;

    scanf("%d", &qua);

    CNode p , tail;

    p = (CNode)malloc(sizeof(Node));

    \*head = p;

    for (i = 0; i < qua; i++) {

        tail = (CNode)malloc(sizeof(Node));

        scanf("%s", &tail->num);

        scanf("%s", &tail->name);

        for (j = 0; j < 4; j++)

            scanf("%d", &tail->score[j]);

        tail->avescore = (tail->score[0] + tail->score[1] + tail->score[2] + tail->score[3]) / 4.0;

        p->next = tail;

        p = tail;

    }

    tail->next = NULL;

    printf("完成了%d位同学的成绩输入。\n",qua);

}

void PrintNode(CNode\* head) {

    CNode p = \*head;

    while (p->next) {

        printf("%s %s ", p->next->num, p->next->name);

        for (int i = 0; i < 4; i++)

            printf("%d ", p->next->score[i]);

        p = p->next;

        printf("\n");

    }

}

void ChangeNode(CNode\* head) {

    CNode p = \*head;

    char num[100];

    int k = -1;

    scanf("%s", &num);

    while (p->next) {

        if (!strcmp(p->next->num, num)) {

            scanf("%d", &k);

            break;

        }

        p = p->next;

    }

    if (k == 0)

        scanf("%s", &p->next->name);

    if (k > 0) {

        scanf("%d", &p->next->score[k - 1]);

        p->next->avescore = (p->next->score[0] + p->next->score[1] + p->next->score[2] + p->next->score[3]) / 4.0;

    }

    printf("%s %s ", p->next->num, p->next->name);

    for (int i = 0; i < 4; i++)

        printf("%d ", p->next->score[i]);

    printf("\n");

}

void AveNode(CNode\* head) {

    CNode p = \*head;

    while(p->next) {

        printf("%s %s %.2lf\n", p->next->num, p->next->name, p->next->avescore);

        p = p->next;

    }

}

void SumNode(CNode\* head) {

    CNode p = \*head;

    while (p->next) {

        printf("%s %s ", p->next->num, p->next->name);

        double sum = 0;

        for (int i = 0; i < 4; i++)

            sum += p->next->score[i];

        printf("%.0lf %.2lf\n", sum, p->next->avescore);

        p = p->next;

    }

}

void SortNode(CNode\* head) {

    CNode p = \*head, q = \*head;

    int n = 0;

    char numb[100], nam[100];

    while (p->next) {

        n++;

        p = p->next;

    }

    for (int i = 0; i < n - 1; i++,q=\*head) {

        for (int j = 0; j < n - 1 - i; j++,q=q->next) {

            if (q->next->avescore > q->next->next->avescore) {

                strcpy(numb, q->next->num);

                strcpy(q->next->num, q->next->next->num);

                strcpy(q->next->next->num, numb);

                strcpy(nam, q->next->name);

                strcpy(q->next->name, q->next->next->name);

                strcpy(q->next->next->name, nam);

                for (int k = 0; k < 4; k++) {

                    int tmp = q->next->score[k];

                    q->next->score[k] = q->next->next->score[k];

                    q->next->next->score[k] = tmp;

                }

                double tmp2 = q->next->avescore;

                q->next->avescore = q->next->next->avescore;

                q->next->next->avescore = tmp2;

            }

        }

    }

    p = \*head;

    while (p->next) {

        printf("%s %s %.2lf\n", p->next->num, p->next->name,p->next->avescore);

        p = p->next;

    }

}

int main(void) {

    int n;

    CNode head = NULL;

    while ((scanf("%d", &n) != EOF)) {

        if (n == 0)break;

        switch (n) {

        case 1:CreatNode(&head); break;

        case 2:PrintNode(&head); break;

        case 3:ChangeNode(&head); break;

        case 4:AveNode(&head); break;

        case 5:SumNode(&head); break;

        case 6:SortNode(&head); break;

        }

    }

}

3）测试

（a） 测试数据：

**测试输入1：**

1

5

2021001 Jack 90 92 87 95

2021002 Mike 85 70 75 90

2021003 Joe 77 86 90 75

2021004 Andy 95 97 92 95

2021005 Rose 90 87 88 89

**预期输出1：**

完成了5位同学的成绩输入。

**测试输入2：**

1

5

2021001 Jack 90 92 87 95

2021002 Mike 85 70 75 90

2021003 Joe 77 86 90 75

2021004 Andy 95 97 92 95

2021005 Rose 90 87 88 89

6

**预期输出2：**

完成了5位同学的成绩输入。

2021002 Mike 80.00

2021003 Joe 82.00

2021005 Rose 88.50

2021001 Jack 91.00

2021004 Andy 94.75

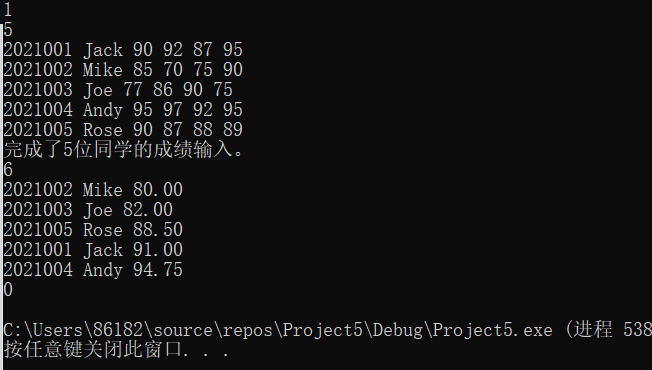
**测试输入3：**

0

**预期输出3：**

（退出）

（b） 对应测试数据的运行结果截图



（3）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。

**解答：**

1. 解题思路：

更改上一题中的SortNode函数，使用冒泡排序，定义一指针p指向头结点，如果p->next->avescore大于p->next->next->avescore，将这两个结点进行交换：找到第一个结点的前驱，使其指向该结点的后继；使该结点的后继指向第二个结点的后继；最后让第二个结点的后继指向第一个节点。

2）程序清单

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

typedef struct Node {

char num[100], name[100];

int score[4];

double avescore;

struct Node\* next;

}Node, \* CNode;

void CreatNode(CNode \*head) {

int qua, i, j;

scanf("%d", &qua);

CNode p , tail;

p = (CNode)malloc(sizeof(Node));

\*head = p;

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

tail = (CNode)malloc(sizeof(Node));

scanf("%s", &tail->num);

scanf("%s", &tail->name);

for (j = 0; j < 4; j++)

scanf("%d", &tail->score[j]);

tail->avescore = (tail->score[0] + tail->score[1] + tail->score[2] + tail->score[3]) / 4.0;

p->next = tail;

p = tail;

}

tail->next = NULL;

printf("完成了%d位同学的成绩输入。\n",qua);

}

void PrintNode(CNode\* head) {

CNode p = \*head;

while (p->next) {

printf("%s %s ", p->next->num, p->next->name);

for (int i = 0; i < 4; i++)

printf("%d ", p->next->score[i]);

p = p->next;

printf("\n");

}

}

void ChangeNode(CNode\* head) {

CNode p = \*head;

char num[100];

int k = -1;

scanf("%s", &num);

while (p->next) {

if (!strcmp(p->next->num, num)) {

scanf("%d", &k);

break;

}

p = p->next;

}

if (k == 0)

scanf("%s", &p->next->name);

if (k > 0) {

scanf("%d", &p->next->score[k - 1]);

p->next->avescore = (p->next->score[0] + p->next->score[1] + p->next->score[2] + p->next->score[3]) / 4.0;

}

printf("%s %s ", p->next->num, p->next->name);

for (int i = 0; i < 4; i++)

printf("%d ", p->next->score[i]);

printf("\n");

}

void AveNode(CNode\* head) {

CNode p = \*head;

while(p->next) {

printf("%s %s %.2lf\n", p->next->num, p->next->name, p->next->avescore);

p = p->next;

}

}

void SumNode(CNode\* head) {

CNode p = \*head;

while (p->next) {

printf("%s %s ", p->next->num, p->next->name);

double sum = 0;

for (int i = 0; i < 4; i++)

sum += p->next->score[i];

printf("%.0lf %.2lf\n", sum, p->next->avescore);

p = p->next;

}

}

void SortNode(CNode\* head) {

CNode p = \*head, q = \*head;

int n = 0;

while (p->next) {

n++;

p = p->next;

}

for (int i = 0; i < n - 1; i++, q = \*head) {

for (int j = 0; j < n - 1 - i; j++, q = q->next) {

if (q->next->avescore > q->next->next->avescore) {

CNode k = q->next, m = k->next;

q->next = m;

k->next = m->next;

m->next = k;

}

}

}

p = \*head;

while (p->next) {

printf("%s %s %.2lf\n", p->next->num, p->next->name, p->next->avescore);

p = p->next;

}

}

int main(void) {

int n;

CNode head = NULL;

while ((scanf("%d", &n) != EOF)) {

if (n == 0)break;

switch (n) {

case 1:CreatNode(&head); break;

case 2:PrintNode(&head); break;

case 3:ChangeNode(&head); break;

case 4:AveNode(&head); break;

case 5:SumNode(&head); break;

case 6:SortNode(&head); break;

}

}

}

3）测试

（a）测试数据：

**测试输入1：**

1

5

2021001 Jack 90 92 87 95

2021002 Mike 85 70 75 90

2021003 Joe 77 86 90 75

2021004 Andy 95 97 92 95

2021005 Rose 90 87 88 89

**预期输出1：**

完成了5位同学的成绩输入。

**测试输入2：**

6

**预期输出2：**

2021002 Mike 80.00

2021003 Joe 82.00

2021005 Rose 88.50

2021001 Jack 91.00

2021004 Andy 94.75

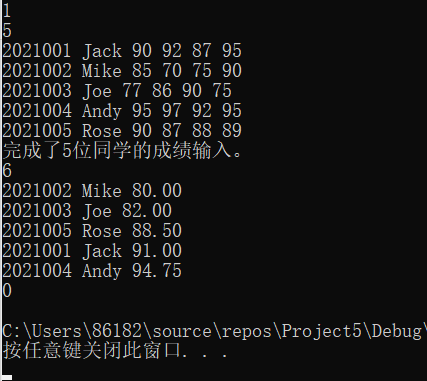
**测试输入3：**

0

**预期输出3：**

（退出）

（b） 对应测试数据的运行结果截图



## 7.3 实验小结

主要叙述实验过程中遇到的问题，如何解决的，通过分析、结果问题后的体会。

在表达式求值的程序验证题中，我熟悉和掌握了结构的说明和引用、结构的指针、结构数组、以及函数中使用结构的方法。在源程序修改替换题中，我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。在程序设计题中，我掌握了单链表的各种操作，熟悉了按数据域对结点排序的两种方法。