吴老师，骆学长和孙学长好，以下是我的作业。

上传课后作业

–教材

P. 185 题 5(编程题，学生成绩管理系统)

要求：1) 结构体成员包括姓名、学号和至少 3 门自定课程；

2) 能够录入学生信息；

3) 求取某一门指定课程的平均分和某一个学生的总分；

4) 按学号或姓名查找学生的信息并显示；

5) 按指定的若干门课程由高到低显示；

6) 按总分由高到低显示学生信息。

测试样例：信息具体内容自定，作业报告中的截图需完成反映以上各功能，并进行简要的文字对应说明，遗漏或错误不得分。

P.186 题 6(编程题，扑克牌游戏)

要求：1) 按题设要求，建立扑克牌的结构类型描述；

2) 随机洗牌函数

3) n 人玩牌的发牌程序$n = 3, n = 4$。

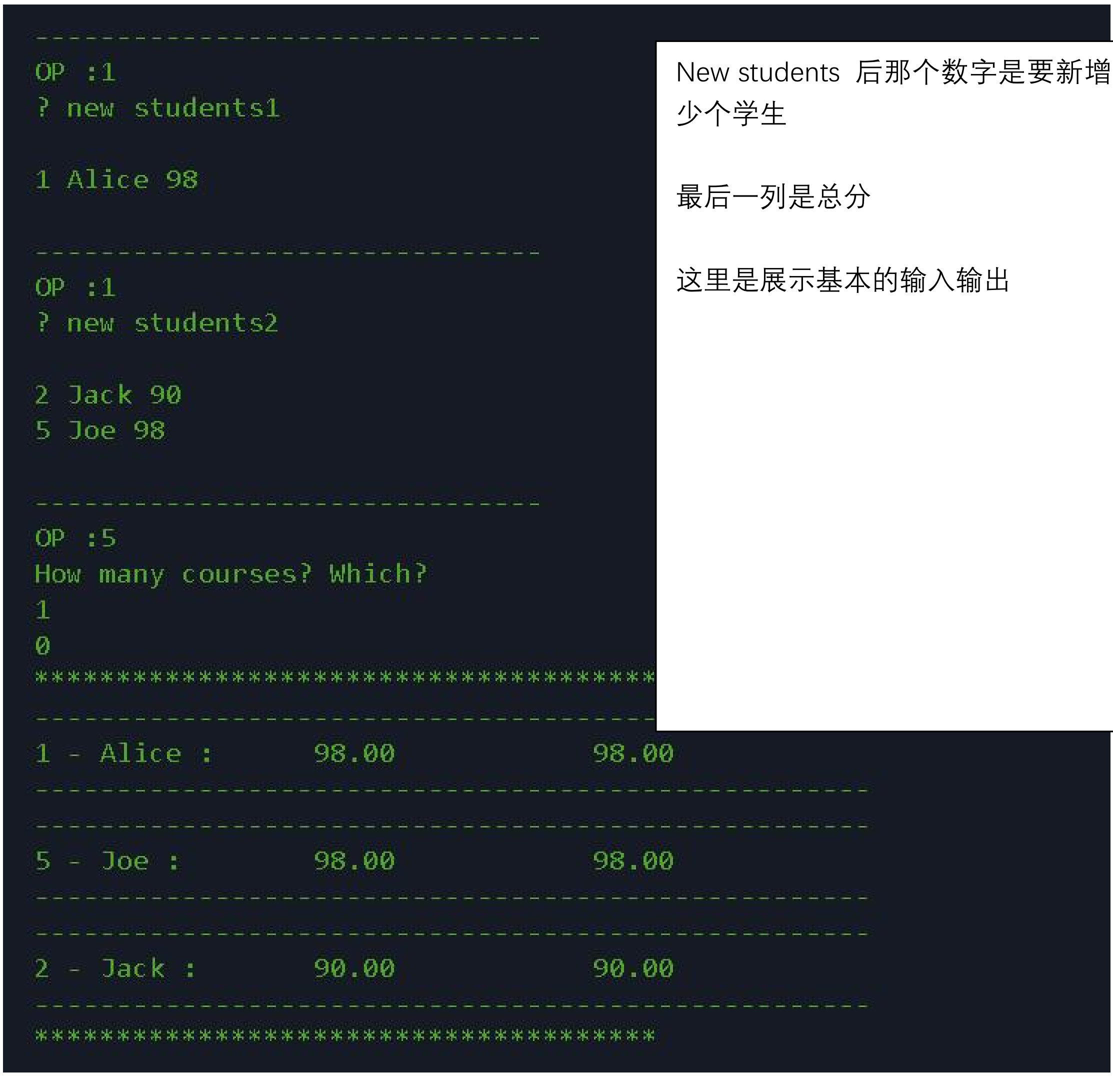
测试样例：作业报告中的截图需完成反映以上各功能。

Chi-Shan0707/Homework-in-CS10004-Programming-by-yhchi

代码仓库↑

1.

先给了一些小的有针对性的输出测试（课程数<3,仅作为 部分测试（也只有这种小测试可以截图）完整的输出内容在最后

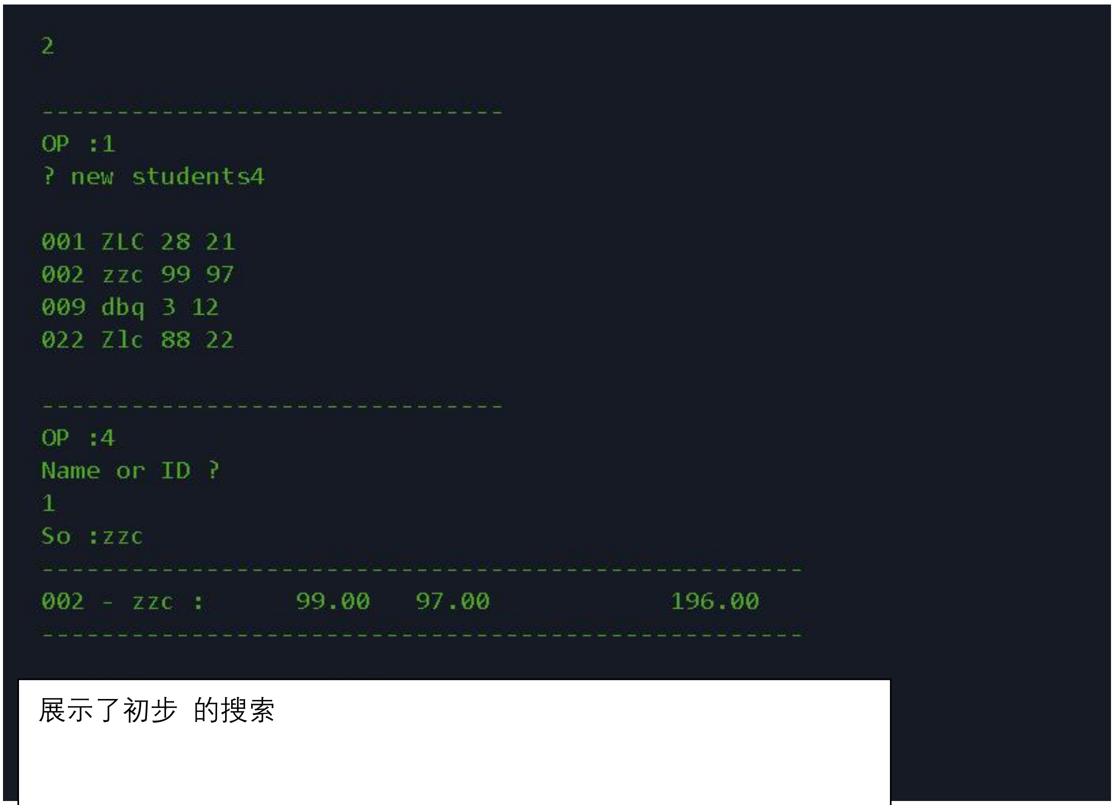


```
-------------------------------
OP :1
? new students1

1 Alice 98

-------------------------------
OP :1
? new students2

2 Jack 90
5 Joe 98

-------------------------------
OP :5
How many courses? Which?
1
0
**************************************
---------------------------------------------------
1 - Alice :      98.00            98.00
---------------------------------------------------
---------------------------------------------------
5 - Joe :        98.00            98.00
---------------------------------------------------
---------------------------------------------------
2 - Jack :       90.00            90.00
---------------------------------------------------
**************************************
```

New students 后那个数字是要新增多少个学生

最后一列是总分

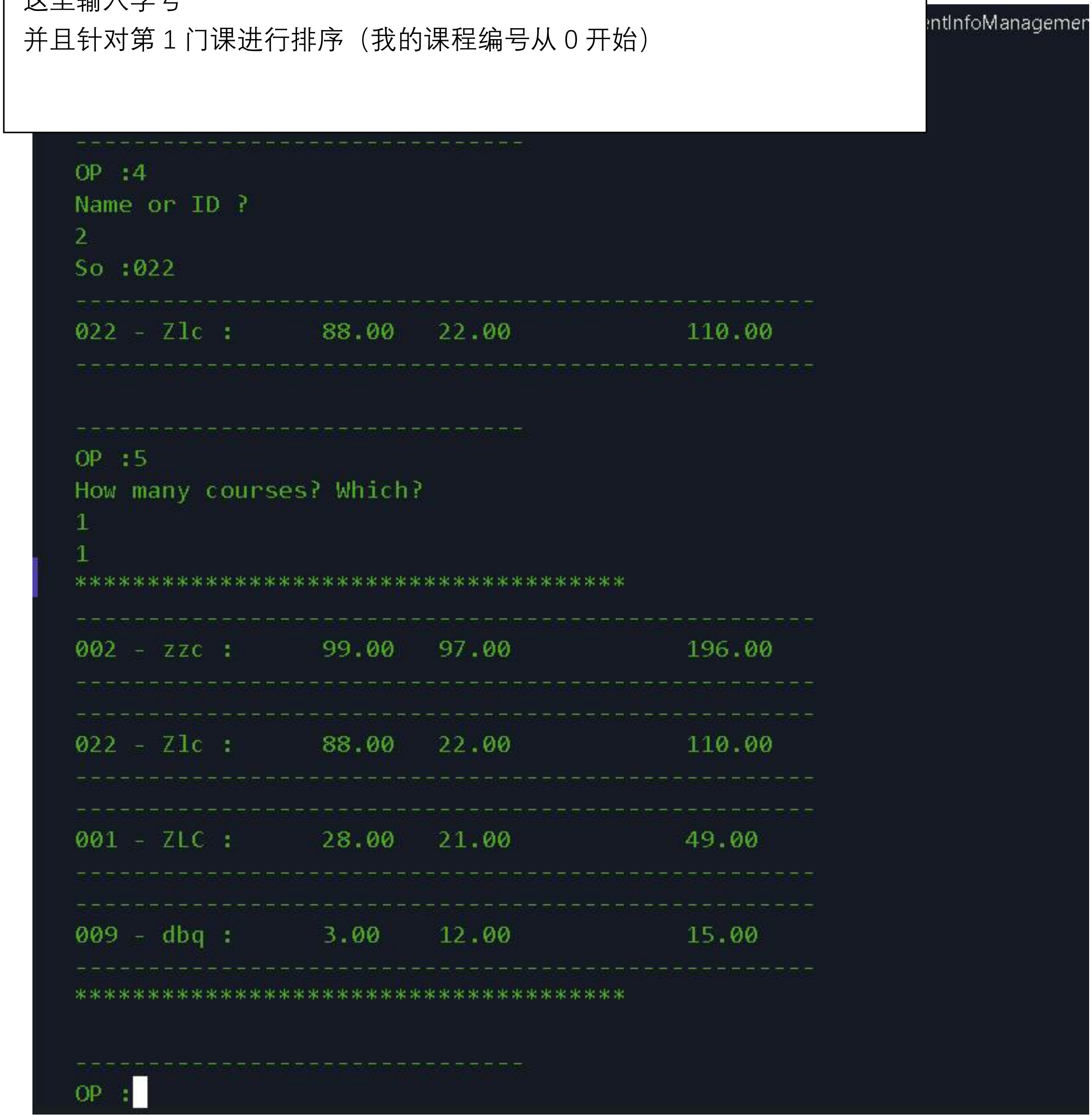
这里是展示基本的输入输出



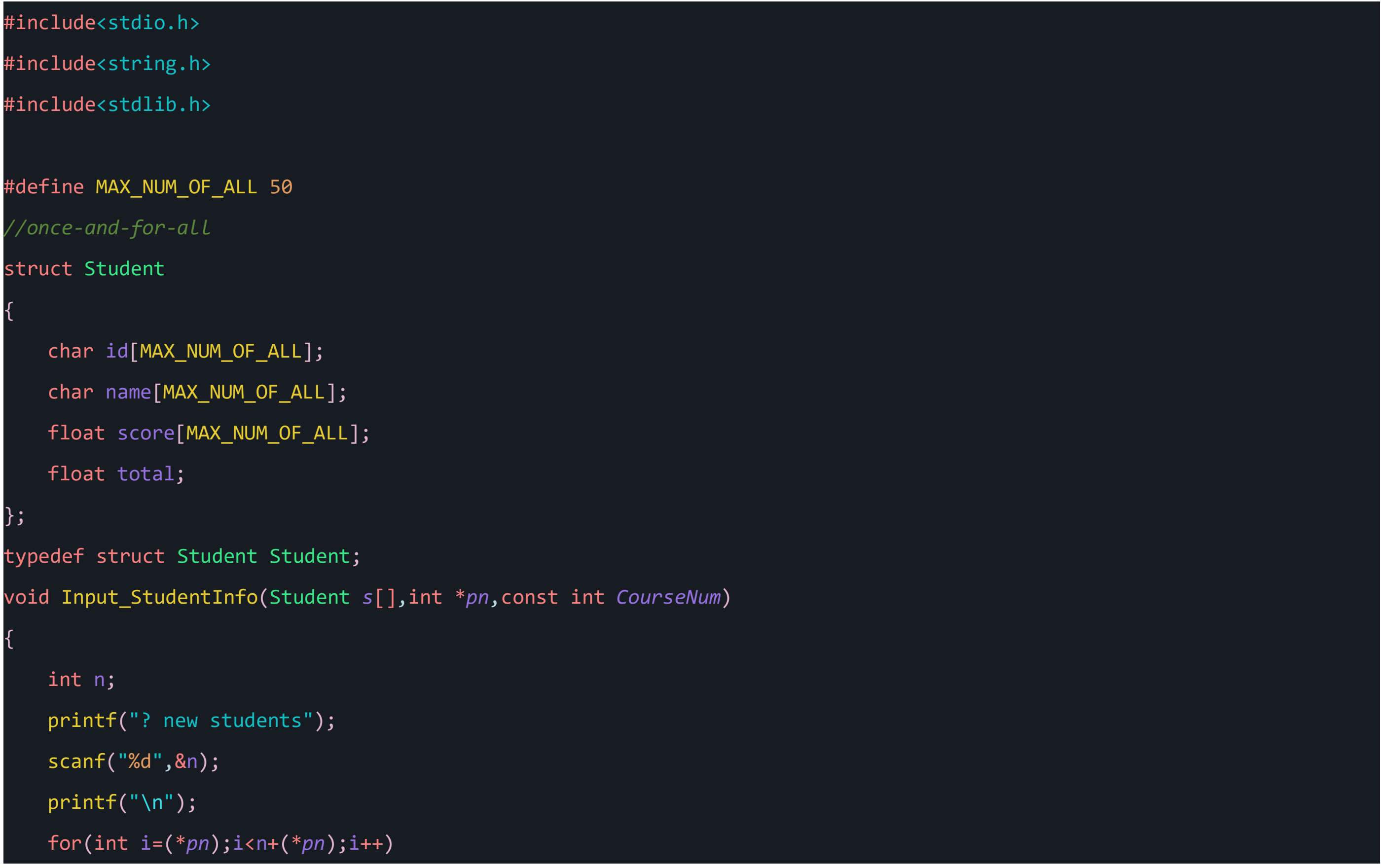
```
2

-------------------------------
OP :1
? new students4

001 ZLC 28 21
002 zzc 99 97
009 dbq 3 12
022 Zlc 88 22

-------------------------------
OP :4
Name or ID ?
1
So :zzc
---------------------------------------------------
002 - zzc :     99.00   97.00           196.00
---------------------------------------------------
```

展示了初步 的搜索

之前是输入姓名
这里输入学号
并且针对第 1 门课进行排序（我的课程编号从 0 开始）



```
-------------------------------
OP :4
Name or ID ?
2
So :022
---------------------------------------------------
022 - Zlc :      88.00   22.00           110.00
---------------------------------------------------


-------------------------------
OP :5
How many courses? Which?
1
1
**************************************
---------------------------------------------------
002 - zzc :      99.00   97.00           196.00
---------------------------------------------------
---------------------------------------------------
022 - Zlc :      88.00   22.00           110.00
---------------------------------------------------
---------------------------------------------------
001 - ZLC :      28.00   21.00           49.00
---------------------------------------------------
---------------------------------------------------
009 - dbq :      3.00    12.00           15.00
---------------------------------------------------
**************************************

-------------------------------
OP :
```



```c
#include<stdio.h>
#include<string.h>
#include<stdlib.h>

#define MAX_NUM_OF_ALL 50
//once-and-for-all
struct Student
{
    char id[MAX_NUM_OF_ALL];
    char name[MAX_NUM_OF_ALL];
    float score[MAX_NUM_OF_ALL];
    float total;
};
typedef struct Student Student;
void Input_StudentInfo(Student s[],int *pn,const int CourseNum)
{
    int n;
    printf("? new students");
    scanf("%d",&n);
    printf("\n");
    for(int i=(*pn);i<n+(*pn);i++)
    {
        s[i].total=0.0f;
        scanf("%s",s[i].id);
        scanf("%s",s[i].name);
        for(int j=0;j<CourseNum;++j)
        {
            scanf("%f",&s[i].score[j]);
            s[i].total+=s[i].score[j];
        }
    }
    //传入指针 进行更改
```

```
    (*pn)+=n;
}
```

```
void print(const Student *p,const int CourseNum)
{
    printf("---------------------------------------------------\n");
    printf("%s - %s :",p->id,p->name);
    for(int j=0;j<CourseNum;j++) printf("\t %.2f",p->score[j]);
    printf("        %.2f\n",p->total);
    printf("---------------------------------------------------\n");
}
```

```
void Query_CourseInfo(const Student s[],const int n,const int CourseNum)
{
    int c;
    printf("Course ?\n");
    scanf("%d",&c);
    float sum=0.0f;
    for(int i=0;i<n;i++)sum+=s[i].score[c];
    printf("sum=%.2f avg=%.2f\n",sum,sum/n);
}
```

```
void Query_StudentInfo(const Student s[],const int n,const int CourseNum)
{
   printf("The kth input?\n");
   int k;scanf("%d",&k);
   --k;
    print(&s[k],CourseNum);



}
```

```
void Find_Student(const Student s[],int n,int CourseNum)
{
```

```
    int mode;
    char key[MAX_NUM_OF_ALL];
```

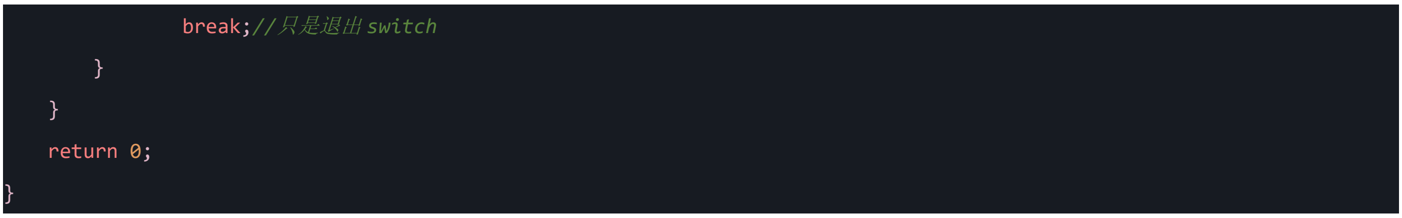
```
     printf("Name or ID ?\n");
    scanf("%d",&mode);
    printf("So :");
    scanf("%s",key);
//查学号or 查姓名
    for(int i=0;i<n;i++)
    {
        if((mode==2&&strcmp(s[i].id,key)==0)||
           (mode==1&&strcmp(s[i].name,key)==0))
        {
            //字符串比较函数
            print(&s[i], CourseNum);
            return;
        }
    }
    printf("not found\n");
}
```

```
void swap_student(Student *a,Student *b)
{
    Student t=*a;
    *a=*b;
    *b=t;
}
```

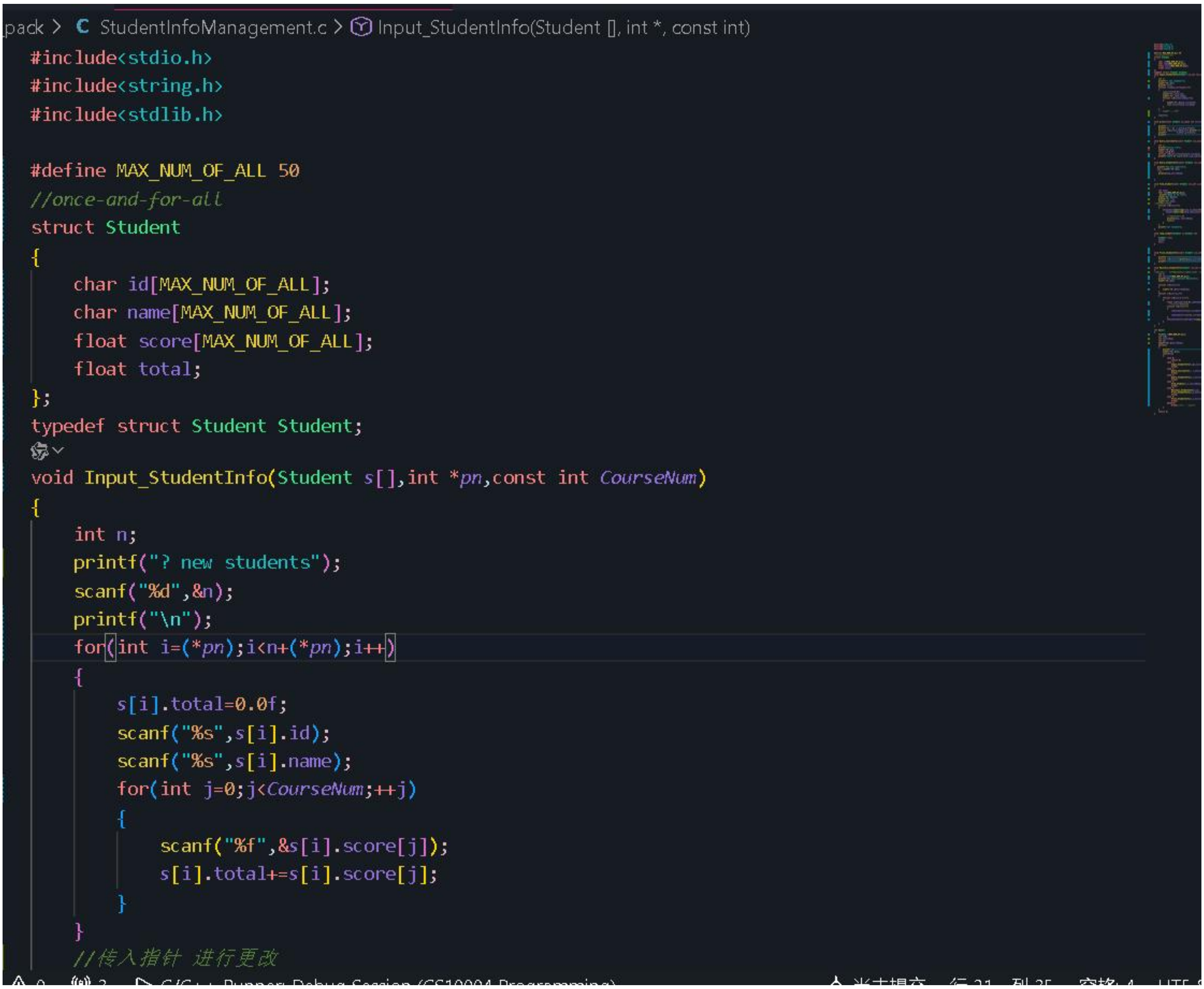
```
void Print_StudentInfo(const Student s[],int n,int CourseNum)
{
    printf("***************************************\n");
    for(int i=0;i<n;i++)print(&s[i],CourseNum);
    printf("***************************************\n");
}
```

```
void Maintain_StudentInfo(Student s[],int n)
{
//如果要按照所有学科排名的话，就手动输入所有学科叭
    int k;
    int CourseId[MAX_NUM_OF_ALL];
    printf("How many courses? Which?\n");
    scanf("%d",&k);

    for(int i=0;i<k;i++)
    {
        scanf("%d",&CourseId[i]);
    }
    for(int i=0;i<n-1;i++)
    {
        for(int j=0;j<n-1-i;j++)
        {
            float sumStudentI=0.0f,sumStudentJ=0.0f;
              //暴力查看交换条件
            for(int t=0;t<k;t++)
            {
                sumStudentI+=s[j].score[CourseId[t]];

                sumStudentJ+=s[j+1].score[CourseId[t]];
            }
            if(sumStudentI<sumStudentJ)swap_student(&s[j],&s[j+1]);
        }
    }
}
```

```
int main()
{
    Student s[MAX_NUM_OF_ALL];
    int n=0;
    int CourseNum;
    int op;
    scanf("%d",&CourseNum);
    while(1)
    {
        printf("\n-----------------------------------\nOP :");
        scanf("%d",&op);
        switch(op)
        {
            case 0:
                return 0;
            case 1:
                Input_StudentInfo(s,&n,CourseNum);
                break;
            case 2:
                Query_CourseInfo(s, n,CourseNum);
                break;
            case 3:
                Query_StudentInfo(s,n,CourseNum);
                break;
            case 4:
                Find_Student(s,n,CourseNum);
                break;
            case 5:
                Maintain_StudentInfo(s,n);
                Print_StudentInfo(s,n,CourseNum);
                break;
            case 6:
                Print_StudentInfo(s,n,CourseNum);
                break;
            default:
```

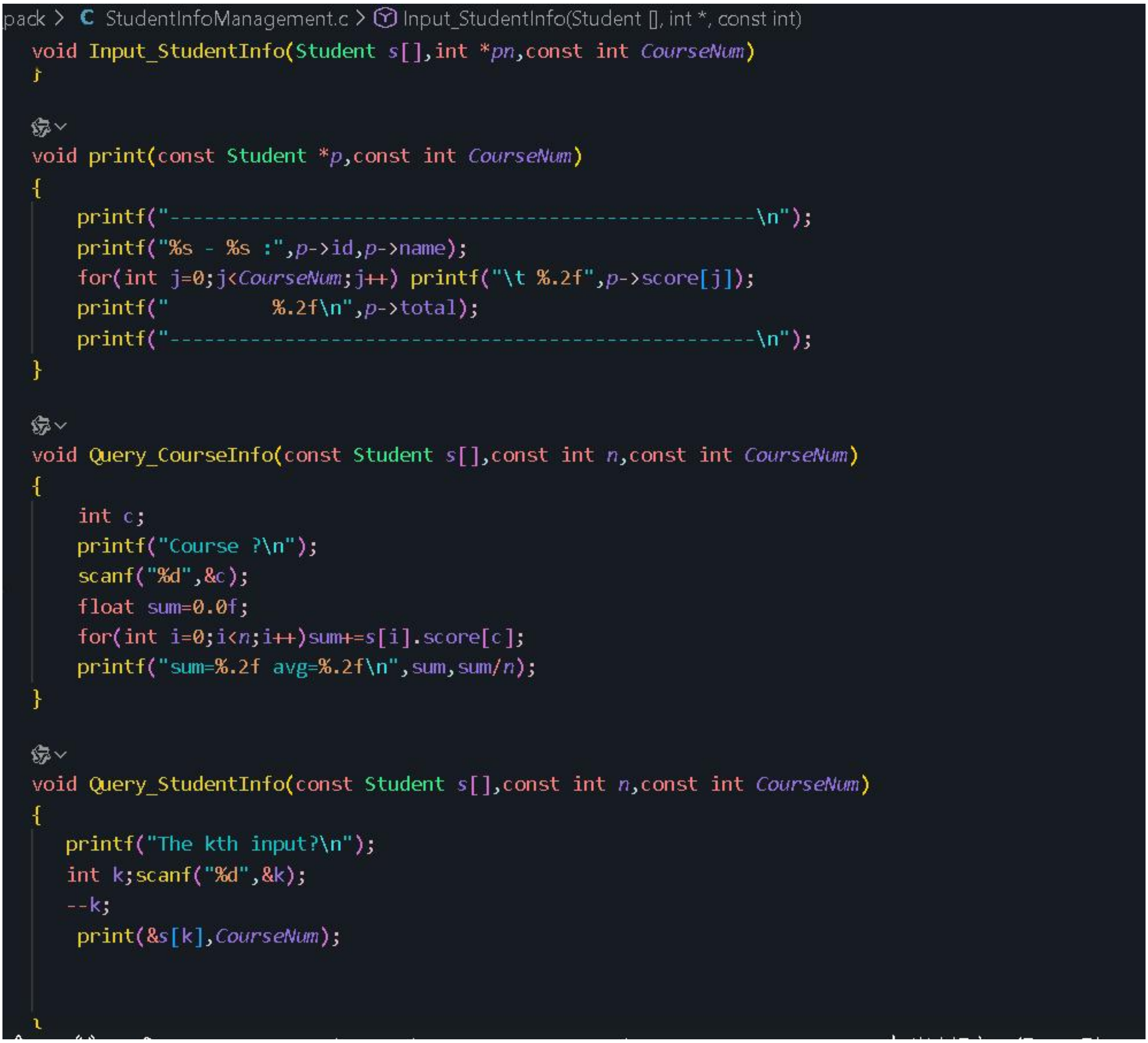


```
                break;//只是退出 switch
            }
        }
        return 0;
}
```

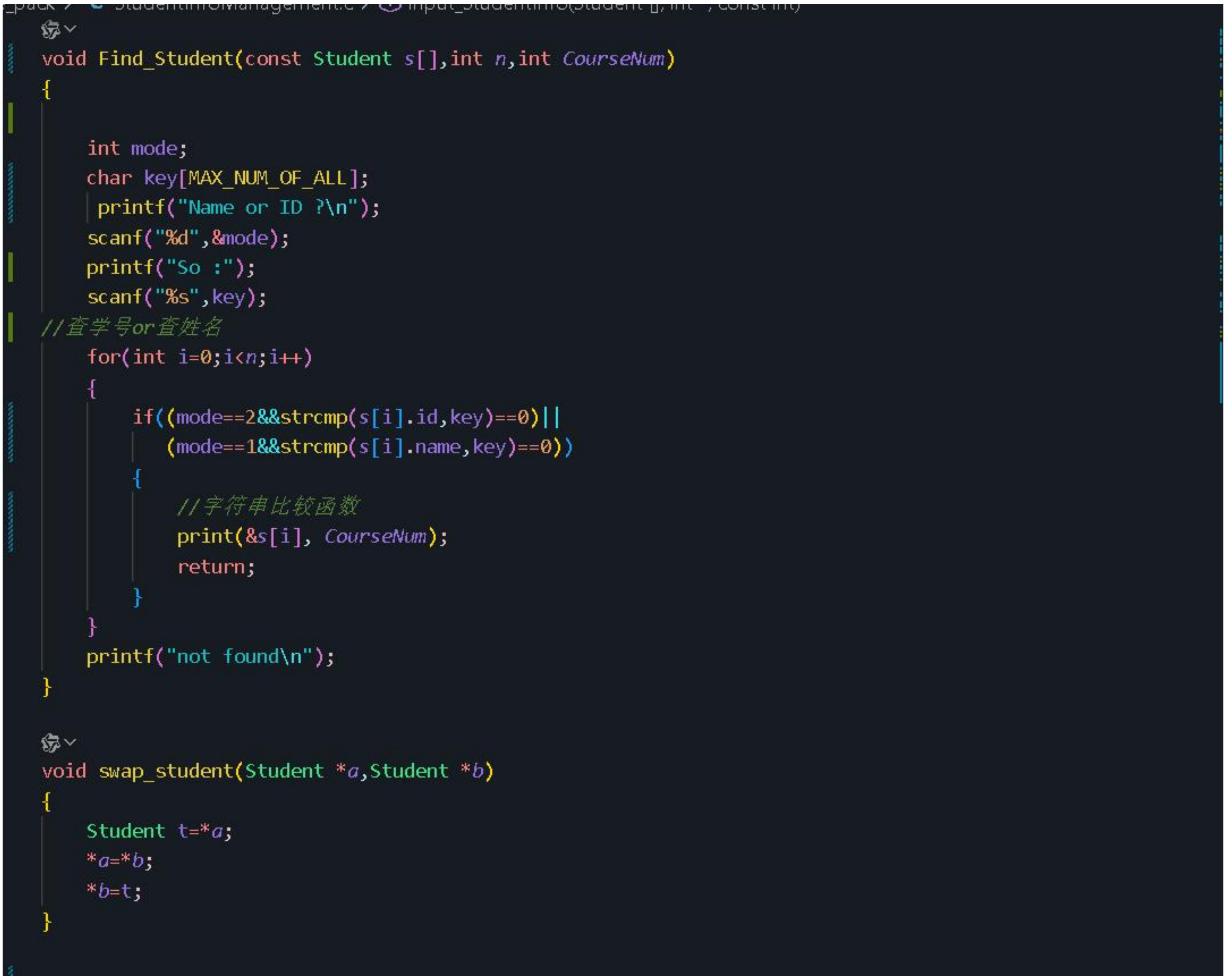


pack > C StudentInfoManagement.c > Input_StudentInfo(Student [], int *, const int)

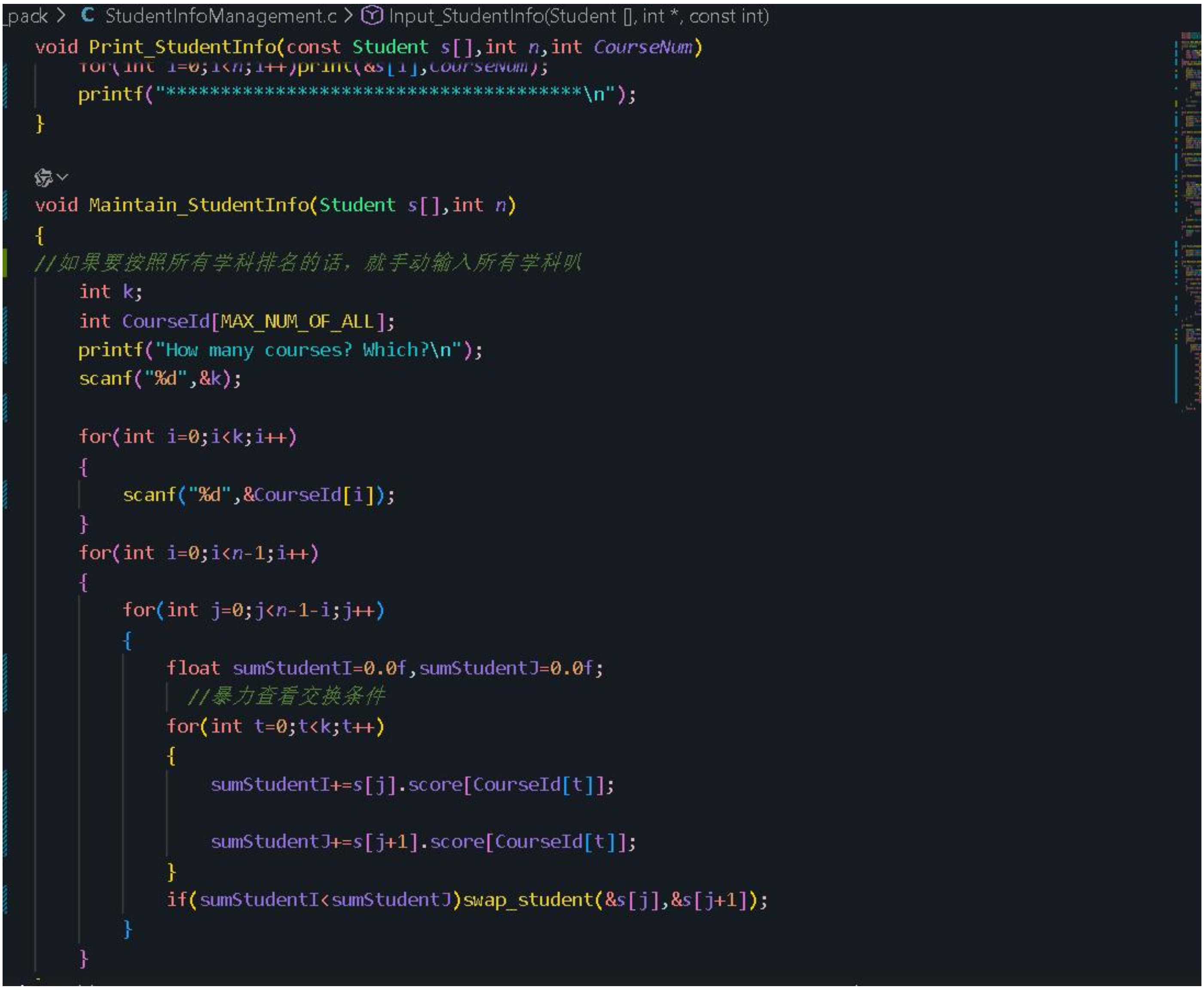
```
#include<stdio.h>
#include<string.h>
#include<stdlib.h>

#define MAX_NUM_OF_ALL 50
//once-and-for-all
struct Student
{
    char id[MAX_NUM_OF_ALL];
    char name[MAX_NUM_OF_ALL];
    float score[MAX_NUM_OF_ALL];
    float total;
};
typedef struct Student Student;

void Input_StudentInfo(Student s[],int *pn,const int CourseNum)
{
    int n;
    printf("? new students");
    scanf("%d",&n);
    printf("\n");
    for(int i=(*pn);i<n+(*pn);i++)
    {
        s[i].total=0.0f;
        scanf("%s",s[i].id);
        scanf("%s",s[i].name);
        for(int j=0;j<CourseNum;++j)
        {
            scanf("%f",&s[i].score[j]);
            s[i].total+=s[i].score[j];
        }
    }
    //传入指针 进行更改
```


pack > C StudentInfoManagement.c > Input_StudentInfo(Student [], int *, const int)

```
void Input_StudentInfo(Student s[],int *pn,const int CourseNum)
}

void print(const Student *p,const int CourseNum)
{
    printf("--------------------------------------------------\n");
    printf("%s - %s :",p->id,p->name);
    for(int j=0;j<CourseNum;j++) printf("\t %.2f",p->score[j]);
    printf("         %.2f\n",p->total);
    printf("--------------------------------------------------\n");
}

void Query_CourseInfo(const Student s[],const int n,const int CourseNum)
{
    int c;
    printf("Course ?\n");
    scanf("%d",&c);
    float sum=0.0f;
    for(int i=0;i<n;i++)sum+=s[i].score[c];
    printf("sum=%.2f avg=%.2f\n",sum,sum/n);
}

void Query_StudentInfo(const Student s[],const int n,const int CourseNum)
{
   printf("The kth input?\n");
   int k;scanf("%d",&k);
   --k;
    print(&s[k],CourseNum);


}
```

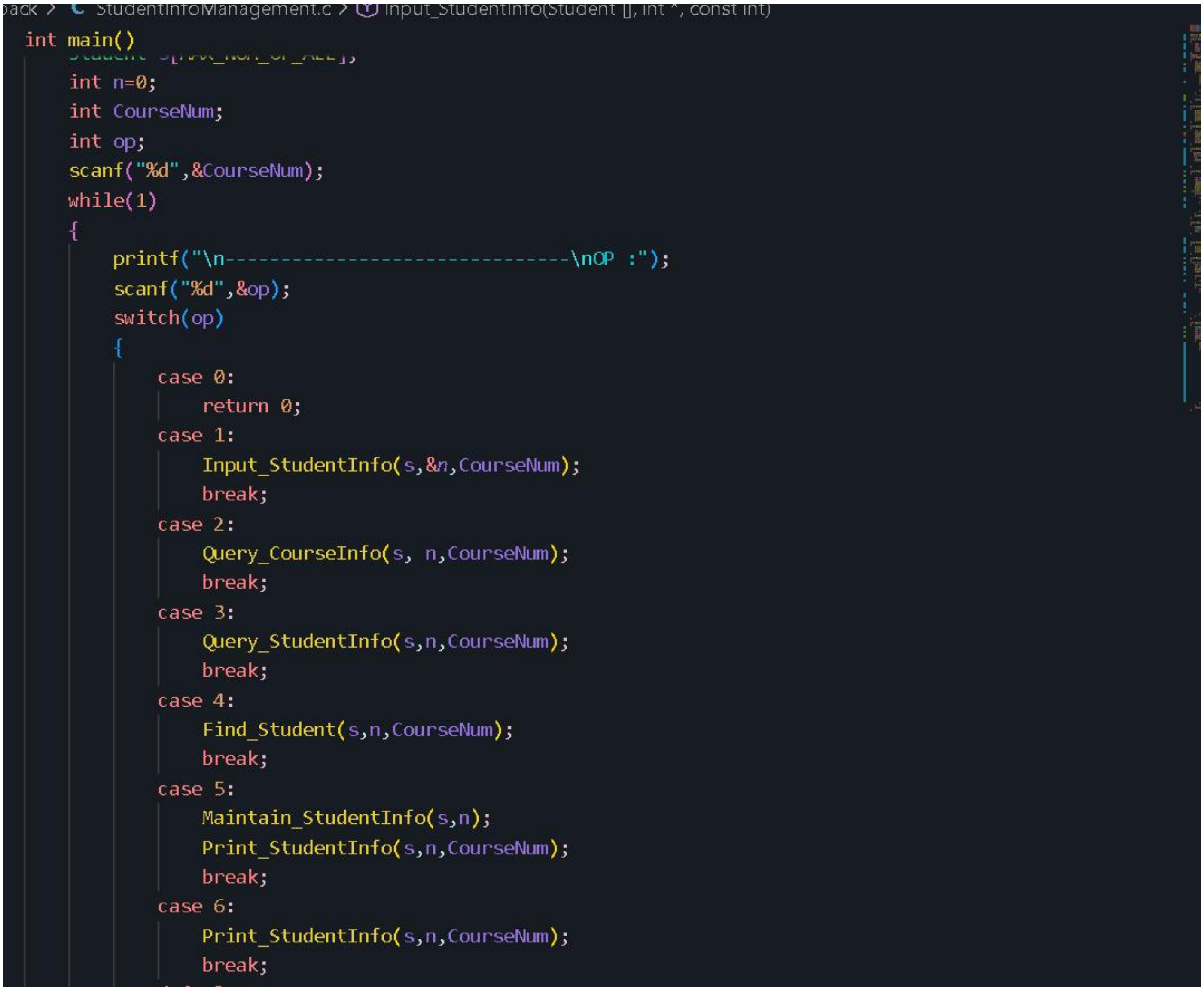

_pack > C StudentInfoManagement.c > input_StudentInfo(Student [], int *, const int)

```c
void Find_Student(const Student s[],int n,int CourseNum)
{

    int mode;
    char key[MAX_NUM_OF_ALL];
     printf("Name or ID ?\n");
    scanf("%d",&mode);
    printf("So :");
    scanf("%s",key);
//查学号or查姓名
    for(int i=0;i<n;i++)
    {
        if((mode==2&&strcmp(s[i].id,key)==0)||
           (mode==1&&strcmp(s[i].name,key)==0))
        {
            //字符串比较函数
            print(&s[i], CourseNum);
            return;
        }
    }
    printf("not found\n");
}

void swap_student(Student *a,Student *b)
{
    Student t=*a;
    *a=*b;
    *b=t;
}
```



pack > StudentInfoManagement.c > Input_StudentInfo(Student [], int *, const int)

```c
void Print_StudentInfo(const Student s[],int n,int CourseNum)
    for(int i=0;i<n;i++)print(&s[i],CourseNum);
    printf("****************************************\n");
}

void Maintain_StudentInfo(Student s[],int n)
{
//如果要按照所有学科排名的话，就手动输入所有学科叭
    int k;
    int CourseId[MAX_NUM_OF_ALL];
    printf("How many courses? Which?\n");
    scanf("%d",&k);

    for(int i=0;i<k;i++)
    {
        scanf("%d",&CourseId[i]);
    }
    for(int i=0;i<n-1;i++)
    {
        for(int j=0;j<n-1-i;j++)
        {
            float sumStudentI=0.0f,sumStudentJ=0.0f;
             //暴力查看交换条件
            for(int t=0;t<k;t++)
            {
                sumStudentI+=s[j].score[CourseId[t]];

                sumStudentJ+=s[j+1].score[CourseId[t]];
            }
            if(sumStudentI<sumStudentJ)swap_student(&s[j],&s[j+1]);
        }
    }
```



pack > C StudentInfoManagement.c > Input_StudentInfo(Student [], int *, const int)

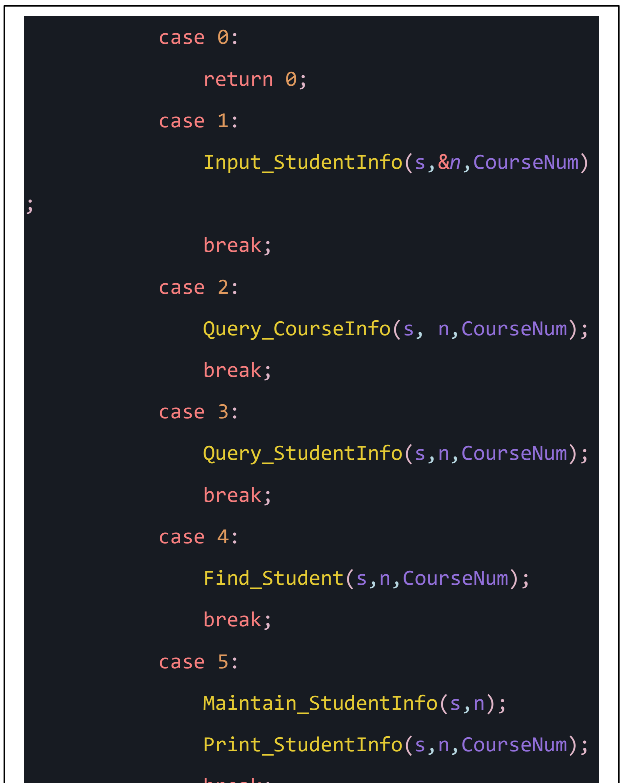
```
int main()
    Student s[MAX_NUM_OF_ALL];
    int n=0;
    int CourseNum;
    int op;
    scanf("%d",&CourseNum);
    while(1)
    {
        printf("\n-----------------------------\nOP :");
        scanf("%d",&op);
        switch(op)
        {
            case 0:
                return 0;
            case 1:
                Input_StudentInfo(s,&n,CourseNum);
                break;
            case 2:
                Query_CourseInfo(s, n,CourseNum);
                break;
            case 3:
                Query_StudentInfo(s,n,CourseNum);
                break;
            case 4:
                Find_Student(s,n,CourseNum);
                break;
            case 5:
                Maintain_StudentInfo(s,n);
                Print_StudentInfo(s,n,CourseNum);
                break;
            case 6:
                Print_StudentInfo(s,n,CourseNum);
                break;
```

一个完整且系统的例子

注

3 表示课程数量

Newstudent 后输入了 6 表示输入学生数



```
            case 0:
                return 0;
            case 1:
                Input_StudentInfo(s,&n,CourseNum)
;
                break;
            case 2:
                Query_CourseInfo(s, n,CourseNum);
                break;
            case 3:
                Query_StudentInfo(s,n,CourseNum);
                break;
            case 4:
                Find_Student(s,n,CourseNum);
                break;
            case 5:
                Maintain_StudentInfo(s,n);
                Print_StudentInfo(s,n,CourseNum);
```

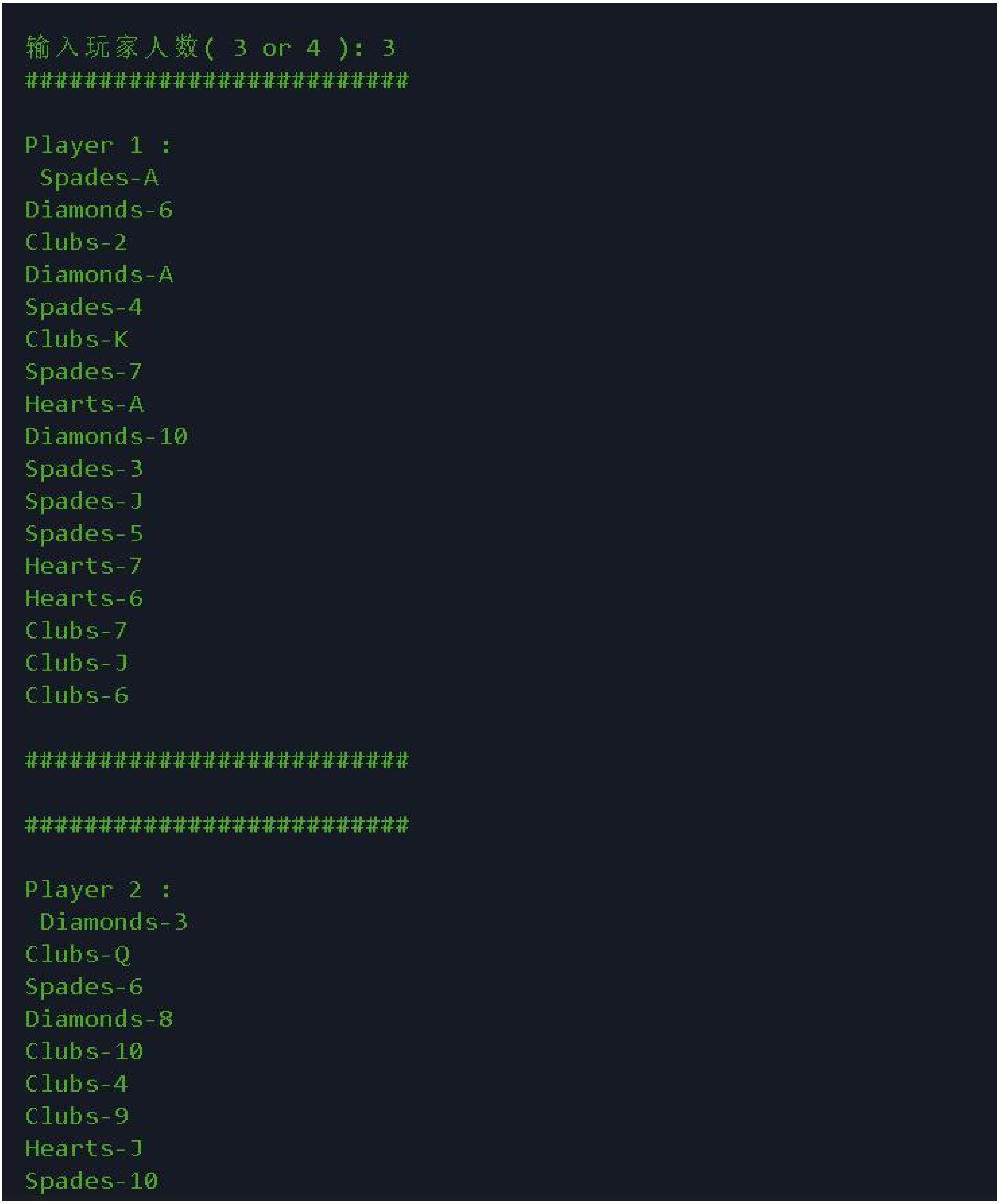
```
3

-------------------------------
OP :1
? new students6

003 jack 100 99 97
023 Pete 88 87 100
034 Sun 90 90 90
100 wiki 100 22 9
101 jason 100 100 100
102 ice 96 78 34

-------------------------------
OP :2
Course ?
2
sum=430.00 avg=71.67

-------------------------------
OP :3
The kth input?
5
---------------------------------------------------
101 - jason :      100.00    100.00    100.00                 300.00
---------------------------------------------------

-------------------------------
OP :4
Name or ID ?
1
So :jason
---------------------------------------------------
101 - jason :      100.00    100.00    100.00                 300.00
---------------------------------------------------

-------------------------------
OP :4
Name or ID ?
2
So :023
---------------------------------------------------
023 - Pete :        88.00     87.00     100.00                 275.00
---------------------------------------------------

-------------------------------
OP :5
How many courses? Which?
3
```

```
0 1 2
*************************************
-------------------------------------------------
101 - jason :      100.00    100.00    100.00                    300.00
-------------------------------------------------
-------------------------------------------------
003 - jack :       100.00    99.00     97.00                    296.00
-------------------------------------------------
-------------------------------------------------
023 - Pete :        88.00     87.00     100.00                    275.00
-------------------------------------------------
-------------------------------------------------
034 - Sun :          90.00     90.00     90.00                    270.00
-------------------------------------------------
-------------------------------------------------
102 - ice :          96.00     78.00     34.00                    208.00
-------------------------------------------------
-------------------------------------------------
100 - wiki :       100.00    22.00     9.00                    131.00
-------------------------------------------------
*************************************

------------------------------
OP :5
How many courses? Which?
2
2 1
*************************************
-------------------------------------------------
101 - jason :      100.00    100.00    100.00                    300.00
-------------------------------------------------
-------------------------------------------------
003 - jack :       100.00    99.00     97.00                    296.00
-------------------------------------------------
-------------------------------------------------
023 - Pete :        88.00     87.00     100.00                    275.00
-------------------------------------------------
-------------------------------------------------
034 - Sun :          90.00     90.00     90.00                    270.00
-------------------------------------------------
-------------------------------------------------
102 - ice :          96.00     78.00     34.00                    208.00
-------------------------------------------------
-------------------------------------------------
100 - wiki :       100.00    22.00     9.00                    131.00
-------------------------------------------------
*************************************

------------------------------
```

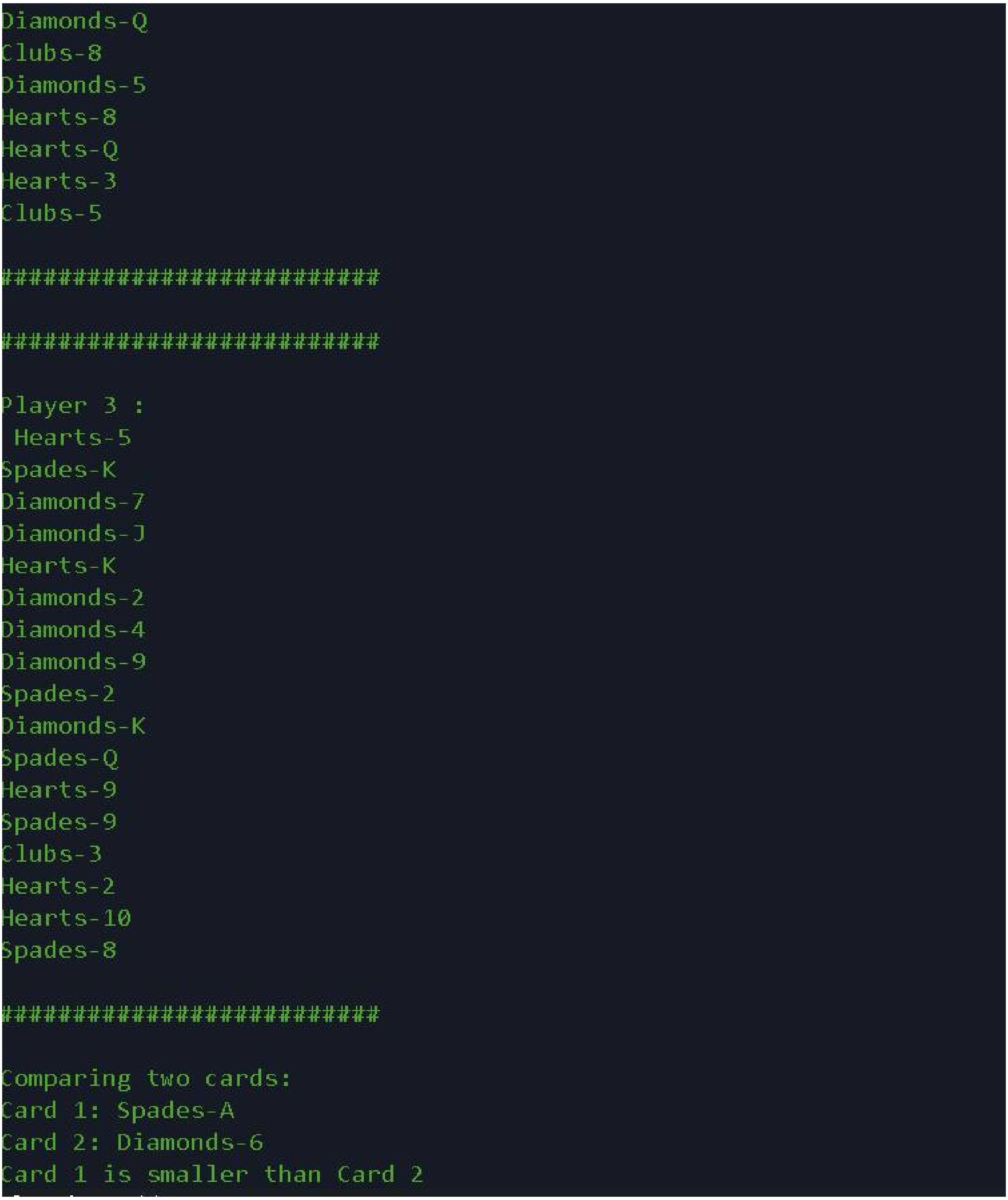
```
OP :5
How many courses? Which?
2
0 1
**************************************
--------------------------------------------------
101 - jason :      100.00   100.00   100.00              300.00
--------------------------------------------------
--------------------------------------------------
003 - jack :       100.00   99.00    97.00               296.00
--------------------------------------------------
--------------------------------------------------
034 - Sun :         90.00    90.00    90.00              270.00
--------------------------------------------------
--------------------------------------------------
023 - Pete :        88.00    87.00    100.00             275.00
--------------------------------------------------
--------------------------------------------------
102 - ice :         96.00    78.00    34.00              208.00
--------------------------------------------------
--------------------------------------------------
100 - wiki :       100.00   22.00    9.00               131.00
--------------------------------------------------
**************************************

------------------------------
OP :6
**************************************
--------------------------------------------------
101 - jason :      100.00   100.00   100.00              300.00
--------------------------------------------------
--------------------------------------------------
003 - jack :       100.00   99.00    97.00               296.00
--------------------------------------------------
--------------------------------------------------
034 - Sun :         90.00    90.00    90.00              270.00
--------------------------------------------------
--------------------------------------------------
023 - Pete :        88.00    87.00    100.00             275.00
--------------------------------------------------
--------------------------------------------------
102 - ice :         96.00    78.00    34.00              208.00
--------------------------------------------------
--------------------------------------------------
100 - wiki :       100.00   22.00    9.00               131.00
--------------------------------------------------
**************************************
```

注 这里输入2表示关键课程数量
再输入课程编号（从0开始）

3 人每人 17 张



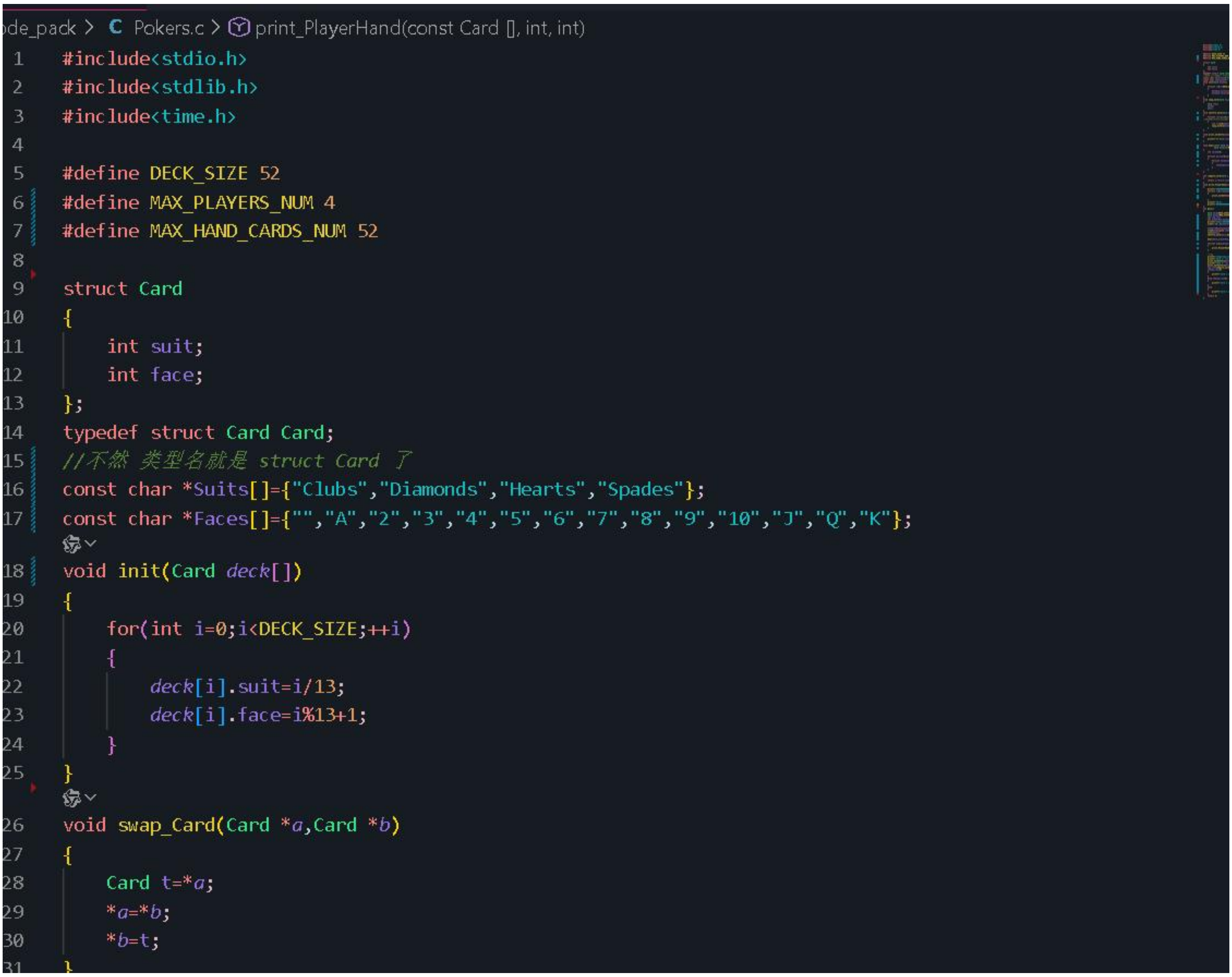
```
输入玩家人数( 3 or 4 ): 3
##########################

Player 1 :
 Spades-A
Diamonds-6
Clubs-2
Diamonds-A
Spades-4
Clubs-K
Spades-7
Hearts-A
Diamonds-10
Spades-3
Spades-J
Spades-5
Hearts-7
Hearts-6
Clubs-7
Clubs-J
Clubs-6

##########################

##########################

Player 2 :
 Diamonds-3
Clubs-Q
Spades-6
Diamonds-8
Clubs-10
Clubs-4
Clubs-9
Hearts-J
Spades-10
```



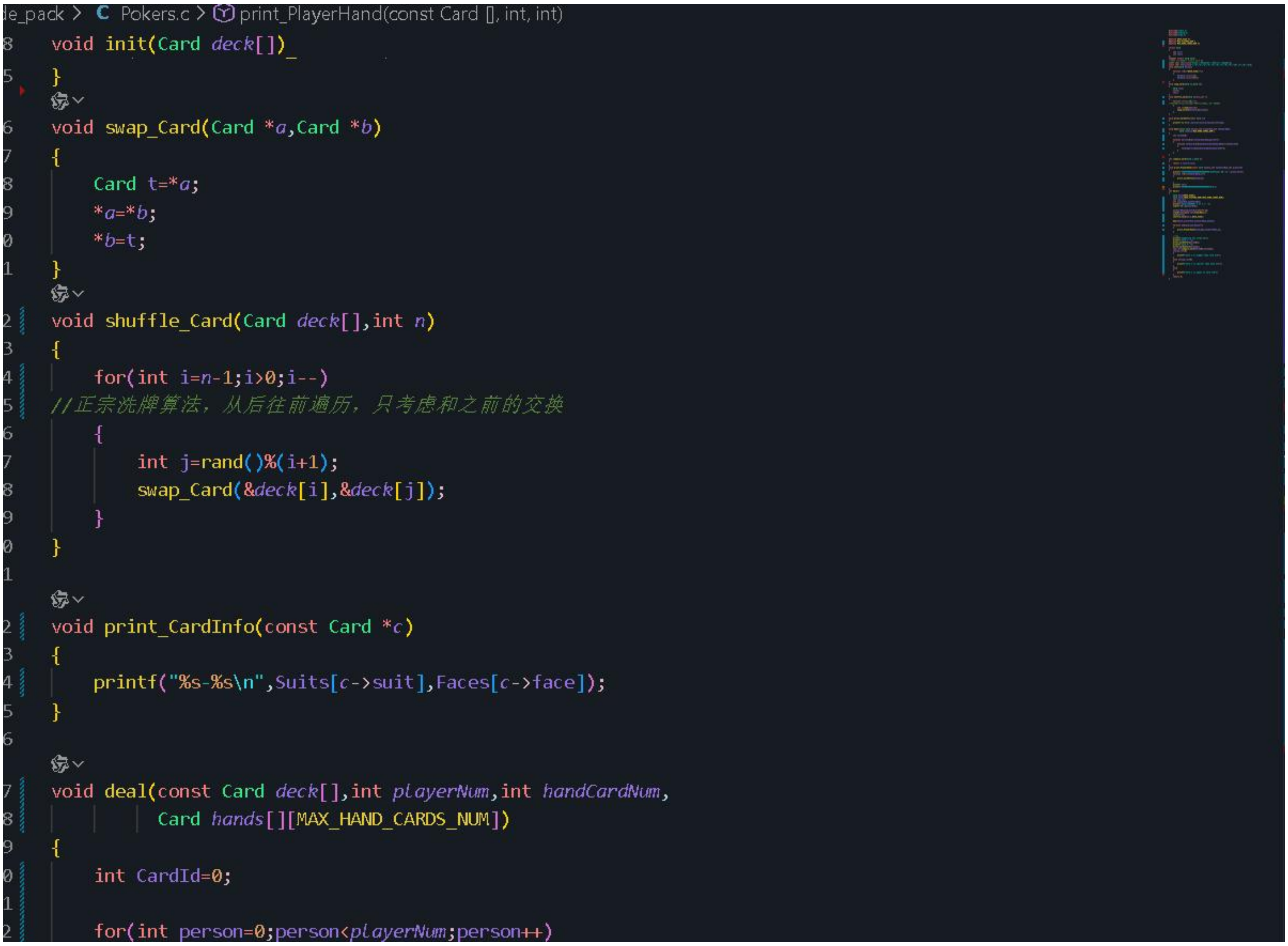
```
Diamonds-Q
Clubs-8
Diamonds-5
Hearts-8
Hearts-Q
Hearts-3
Clubs-5

##########################

##########################

Player 3 :
 Hearts-5
Spades-K
Diamonds-7
Diamonds-J
Hearts-K
Diamonds-2
Diamonds-4
Diamonds-9
Spades-2
Diamonds-K
Spades-Q
Hearts-9
Spades-9
Clubs-3
Hearts-2
Hearts-10
Spades-8

##########################

Comparing two cards:
Card 1: Spades-A
Card 2: Diamonds-6
Card 1 is smaller than Card 2
```

```
输入玩家人数( 3 or 4 ): 4
##########################
Player 1 :
	Hearts-A
Spades-7
Hearts-K
Diamonds-10
Spades-3
Diamonds-6
Diamonds-A
Spades-Q
Hearts-J
Clubs-K
Hearts-5
Clubs-9
Spades-K
##########################
##########################
Player 2 :
	Diamonds-J
Clubs-10
Diamonds-8
Hearts-6
Hearts-3
Clubs-3
Diamonds-Q
Spades-10
Diamonds-9
Hearts-9
Hearts-4
Clubs-2
Clubs-5
##########################
##########################
Player 3 :
	Hearts-Q
Diamonds-3
Hearts-8
Hearts-2
Spades-5
Hearts-7
Clubs-7
Clubs-A
Spades-8
Hearts-10
Clubs-6
Spades-9
Spades-2
```

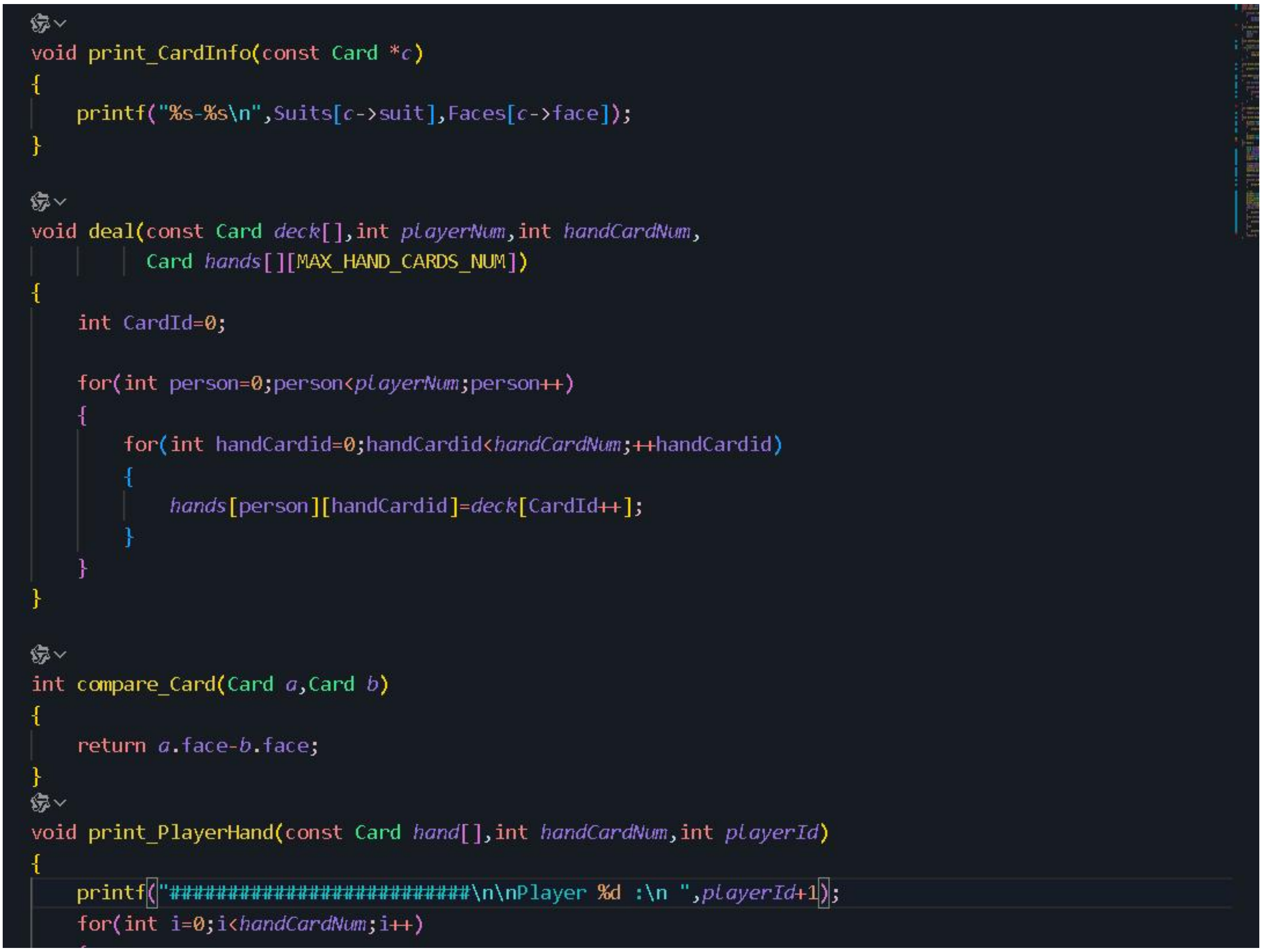
```
###########################
###########################
Player 4 :
        Clubs-J
Spades-6
Clubs-4
Spades-4
Diamonds-K
Spades-J
Diamonds-4
Clubs-8
Diamonds-2
Diamonds-7
Spades-A
Clubs-Q
Diamonds-5
###########################
Comparing two cards:
Card 1: Hearts-A
Card 2: Spades-7
Card 1 is smaller than Card 2
```


de_pack > C Pokers.c > print_PlayerHand(const Card [], int, int)

```c
#include<stdio.h>
#include<stdlib.h>
#include<time.h>

#define DECK_SIZE 52
#define MAX_PLAYERS_NUM 4
#define MAX_HAND_CARDS_NUM 52

struct Card
{
    int suit;
    int face;
};
typedef struct Card Card;
//不然 类型名就是 struct Card 了
const char *Suits[]={"Clubs","Diamonds","Hearts","Spades"};
const char *Faces[]={"","A","2","3","4","5","6","7","8","9","10","J","Q","K"};
void init(Card deck[])
{
    for(int i=0;i<DECK_SIZE;++i)
    {
        deck[i].suit=i/13;
        deck[i].face=i%13+1;
    }
}
void swap_Card(Card *a,Card *b)
{
    Card t=*a;
    *a=*b;
    *b=t;
}
```

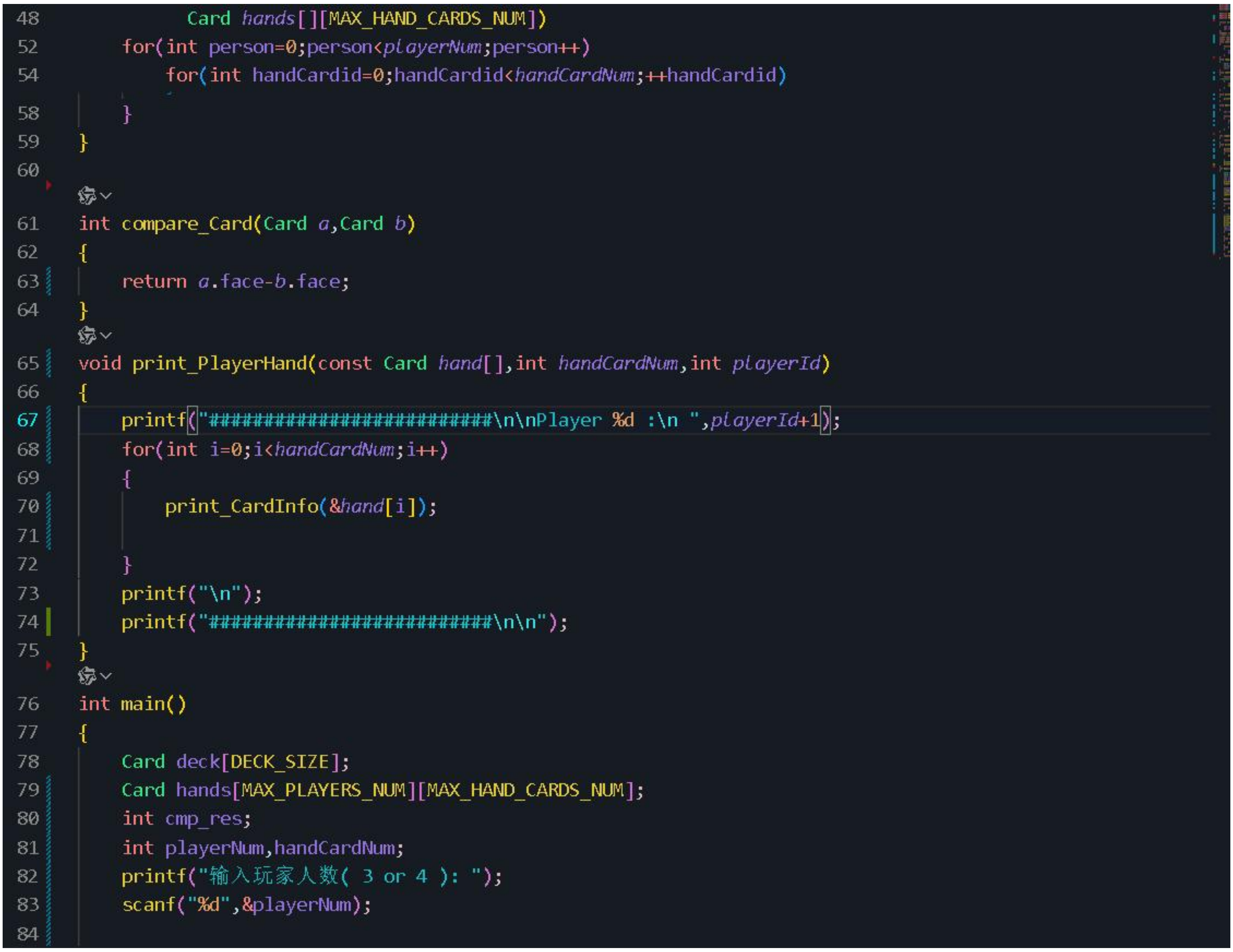

de_pack > Pokers.c > print_PlayerHand(const Card [], int, int)

```
void init(Card deck[])
}
void swap_Card(Card *a,Card *b)
{
    Card t=*a;
    *a=*b;
    *b=t;
}
void shuffle_Card(Card deck[],int n)
{
    for(int i=n-1;i>0;i--)
//正宗洗牌算法，从后往前遍历，只考虑和之前的交换
    {
        int j=rand()%(i+1);
        swap_Card(&deck[i],&deck[j]);
    }
}

void print_CardInfo(const Card *c)
{
    printf("%s-%s\n",Suits[c->suit],Faces[c->face]);
}

void deal(const Card deck[],int playerNum,int handCardNum,
              Card hands[][MAX_HAND_CARDS_NUM])
{
    int CardId=0;

    for(int person=0;person<playerNum;person++)
```

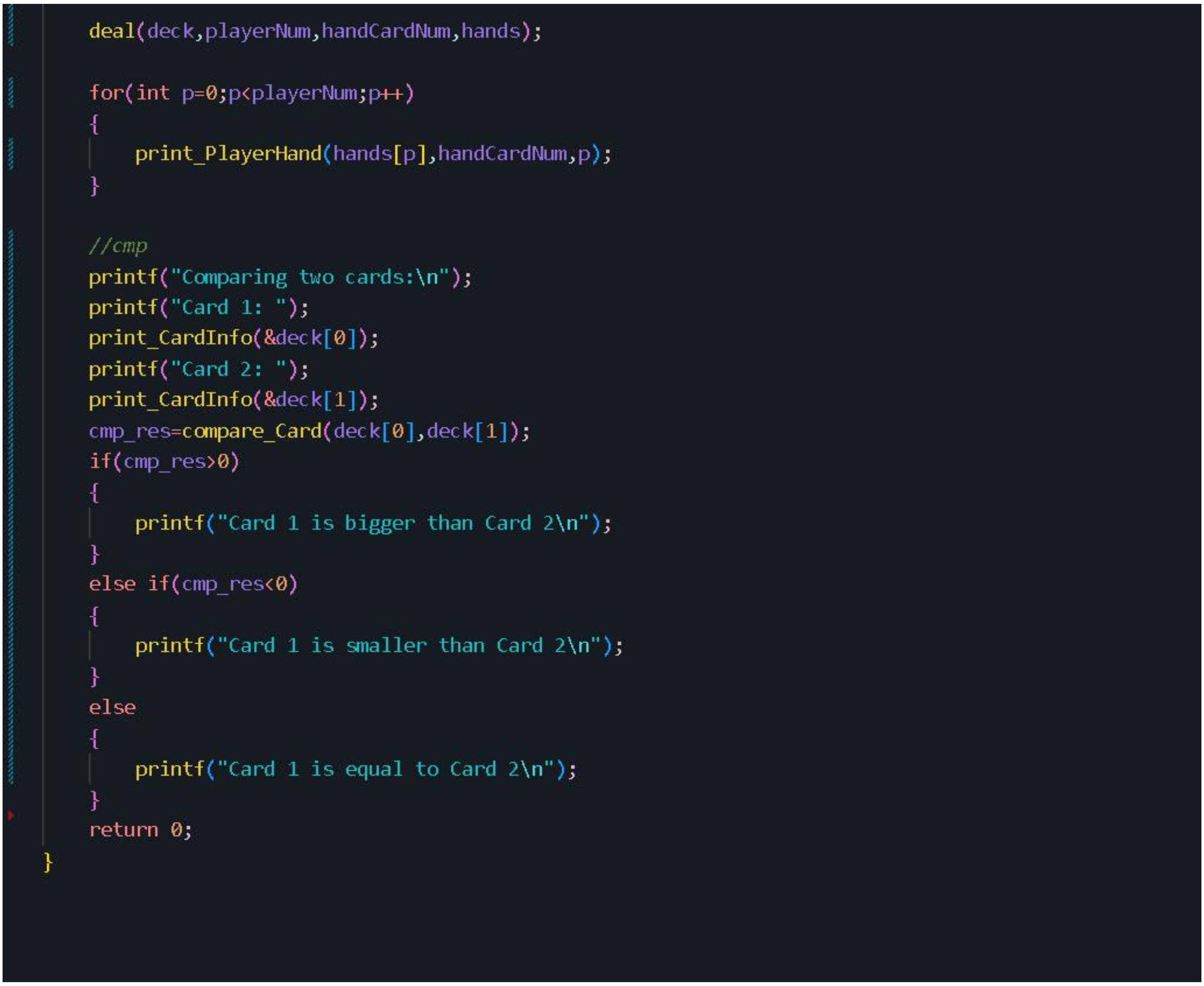


```c
void print_CardInfo(const Card *c)
{
    printf("%s-%s\n",Suits[c->suit],Faces[c->face]);
}

void deal(const Card deck[],int playerNum,int handCardNum,
            Card hands[][MAX_HAND_CARDS_NUM])
{
    int CardId=0;

    for(int person=0;person<playerNum;person++)
    {
        for(int handCardid=0;handCardid<handCardNum;++handCardid)
        {
            hands[person][handCardid]=deck[CardId++];
        }
    }
}

int compare_Card(Card a,Card b)
{
    return a.face-b.face;
}
void print_PlayerHand(const Card hand[],int handCardNum,int playerId)
{
    printf("##########################\n\nPlayer %d :\n ",playerId+1);
    for(int i=0;i<handCardNum;i++)
```

通过计数器去取洗好的牌堆里的牌



```c
          Card hands[][MAX_HAND_CARDS_NUM])
    for(int person=0;person<playerNum;person++)
        for(int handCardid=0;handCardid<handCardNum;++handCardid)
    }
}

int compare_Card(Card a,Card b)
{
    return a.face-b.face;
}
void print_PlayerHand(const Card hand[],int handCardNum,int playerId)
{
    printf("##########################\n\nPlayer %d :\n ",playerId+1);
    for(int i=0;i<handCardNum;i++)
    {
        print_CardInfo(&hand[i]);

    }
    printf("\n");
    printf("##########################\n\n");
}
int main()
{
    Card deck[DECK_SIZE];
    Card hands[MAX_PLAYERS_NUM][MAX_HAND_CARDS_NUM];
    int cmp_res;
    int playerNum,handCardNum;
    printf("输入玩家人数( 3 or 4 ): ");
    scanf("%d",&playerNum);

```



```
    deal(deck,playerNum,handCardNum,hands);

    for(int p=0;p<playerNum;p++)
    {
        print_PlayerHand(hands[p],handCardNum,p);
    }

    //cmp
    printf("Comparing two cards:\n");
    printf("Card 1: ");
    print_CardInfo(&deck[0]);
    printf("Card 2: ");
    print_CardInfo(&deck[1]);
    cmp_res=compare_Card(deck[0],deck[1]);
    if(cmp_res>0)
    {
        printf("Card 1 is bigger than Card 2\n");
    }
    else if(cmp_res<0)
    {
        printf("Card 1 is smaller than Card 2\n");
    }
    else
    {
        printf("Card 1 is equal to Card 2\n");
    }
    return 0;
}
```

这里稍微选两张比个大小