



中國人民大學  
RENMIN UNIVERSITY OF CHINA

# 期中复习专题-1

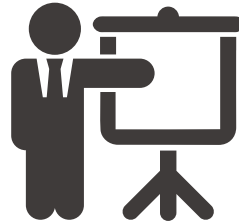
## (条件与枚举)

余力

buaayuli@ruc.edu.cn



中國人民大學  
RENMIN UNIVERSITY OF CHINA



---

# 1. 条件语句

---

if ( 表达式 )

if ( 表达式 )          语句 1;

语句 1;          else  
                     语句 2;

if (<条件表达式1>) <语句1>

**else** if (<条件表达式2>) <语句2>

else if (<条件表达式3>) <语句3>

.....

else if (<条件表达式n>) <语句n>

else <语句n+1>

switch ( 表达式 )

```
{  
    case 常量1: 语句1;  
    case 常量2: 语句2;  
        ⋮  
    case 常量n: 语句n;  
    default:     语句n+1;  
}
```

max = (a>b)? a : b ;

if (a>b)

    max = a;

else

    max = b;

# 闰年

---

```
#include <stdio.h>
int main()
{int year, leap;
 printf("enter year:"); scanf("%d",&year);
 if (year%4==0)
     if(year%100==0)
         if(year%400==0)    leap=1;
         else    leap=0;
     else    leap=1;
 else    leap=0;
 if (leap)    printf("%d is ",year);
 else printf("%d is not ",year);
 printf("a leap year.\n");
 return 0;
}
```

# 密文

---

```
char c;  
c=getchar();  
while(c!='\n')  
{ if((c>='a' && c<='z') || (c>='A' && c<='Z'))  
    { if(c>='W' && c<='Z' || c>='w' && c<='z')  
        c=c+4-26;  
    else c=c+4; }  
    printf("%c",c);  
    c=getchar();  
}
```

# 一元二次方程

```
#include <stdio.h>
#include <math.h>
int main ( )
{
    double a,b,c,disc,x1,x2,p,q;
    scanf("%lf%lf%lf",&a,&b,&c);
    disc=b*b-4*a*c;

    if (disc<0)
        printf( "has not real roots\n" );
    else
    {
        p=-b/(2.0*a);
        q=sqrt(disc)/(2.0*a);
        x1=p+q;
        x2=p-q;
        printf( "real roots:\nx1=%7.2f\n
                x2=%7.2f\n" ,x1,x2);
    }
    return 0;
}
```

## #292 分段函数

已知有如下分段离散函数  $f(x)$ ，其中自变量  $x$  为  $[-1000, 1000]$  之间的实数。

$$f(x) = \begin{cases} |x - 1| - 2 & (|x| \leq 1); \\ \frac{1}{1 + x^2} & (|x| > 1); \end{cases}$$

请编写一个程序计算函数  $f(x)$  的值。注意，当函数值有小数时四舍五入保留 2 位小数。

【输入格式】 输入一个实数  $x$ 。

【输出格式】 输出  $f(x)$  函数值，当函数值有小数时四舍五入保留 2 位小数。

【输入样例 1】

0.5

【输出样例 1】

0.31

## #89 奖金发放

某企业需要一个奖金发放计算器，依据利润区间值和该区间的提成率，以及该企业当月实际利润，计算可用于发放的奖金金额。

例如：区间分界值和提成率如下表：

区间分界值	奖金提成率	含义
100,000	10%	当利润低于 100,000 (含) 时，奖金可提 10%。
200,000	7.5%	高于 100,000，低于或等于 200,000 的部分，按 7.5% 提成。
400,000	5%	高于 200,000，低于或等于 400,000 的部分按 5% 提成。
600,000	3%	高于 400,000，低于或等于 600,000 的部分按 3% 提成。
1,000,000	1.5%	高于 600,000，低于或等于 1,000,000 的部分按照 1.5% 提成。
-1	1%	高于 1,000,000 的部分按照 1% 提成。

当利润为 150,000 时，应发放的奖金总数为：

$$100000 \times 10\% + (150000 - 100000) \times 7.5\% = 13750$$

使用该计算方法，当用户输入不同的区间分界值、提成率和当月利润，计算并输出应发奖金总数。

【输入格式】

输入 n+2 行。

第 1 行包含一个整数 n，表示有 n 行分界值和奖金提成率。

中间 n 行每行包含 1 个正整数和 1 个小数，整数表示区间分界值（并且按从小到大顺序输入），小数表示提成率，两个数字之间用一个空格分隔。注意：第 n 个区间分界值为 -1，表示利润超出上一个区间分界值的其余情况；另外，输入的区间保证是连续、正确的。

第 n+2 行一个整数，表示当月利润，可正可负，如果利润为负，则输出 "NO"。

【输出格式】

1 行仅 1 个数字，表示为当月应发奖金总数，保留小数点后两位数。

如果利润为负，则输出 "NO"。



```
if (p < 0) → printf("NO"); ↵
```

```
else if (n == 1) { ↵
```

```
→ if (p > m[0]) → w = p * q[0]; ↵
```

```
→ else → → w = 0.00; ↵
```

```
→ printf("%.2lf", w); ↵
```

```
} else ↵
```

```
→ { ↵
```

```
→ → for (i = 0; i < n; i++) ↵
```

```
→ → → if (p > m[i]) { ↵
```

```
→ → → → if (i == 0) → w = w + m[0] * q[0]; ↵
```

```
→ → → → else if (i == n - 1) → w = w + (p - m[i - 1]) * q[i]; ↵
```

```
→ → → → → else → → w = w + (m[i] - m[i - 1]) * q[i]; } ↵
```

```
→ → → else { ↵
```

```
→ → → → if (i == 0) → w = p * q[0]; ↵
```

```
→ → → → else → → w = w + (p - m[i - 1]) * q[i]; ↵
```

```
→ → → → break; → → → } ↵
```

```
→ → printf("%.2lf", w); ↵
```

```
→ } ↵
```



中國人民大學  
RENMIN UNIVERSITY OF CHINA



---

## 2. 枚举

---

# #147 教室排课

```
... for(i=0;i<8;i++) ↵
...     for(j=0;j<8;j++) ↵
...         for(k=0;k<8;k++) ↵
...             for(h=0;h<8;h++) ↵
...                 if(a<=s[i]&&b<=s[j]&&c<=s[k]&&d<=s[h] ↵
...                     &&!=j&&!=k&&k!=h&&!=k&&!=h&&j!=h) ↵
...                     {printf("%d%d%d%d\n",i+1,j+1,k+1,h+1); ↵
...                         flag++;} ↵
```

```
for(i=0;i<8;i++) ↵
```

```
→ if(a<=s[i]) ↵
```

```
→ → for(j=0;j<8;j++) ↵
```

```
→ → → if(b<=s[j]) ↵
```

```
→ → → → for(k=0;k<8;k++) ↵
```

```
→ → → → → if(c<=s[k]) ↵
```

```
→ → → → → → for(h=0;h<8;h++) ↵
```

```
→ → → → → → → if(d<=s[h]) ↵
```

```
→ → → → → → → → if(!=j&&!=k&&!=h&&!=k&&!=h&&!=k&&!=h) ↵
```

```
→ → → → → → → → → flag=printf("%d%d%d%d\n",i,j,k,h); ↵
```

```
int s[]={120,40,85,50,100,140,70,100}; ↵
```

```
scanf("%d%d%d%d",&a,&b,&c,&d); ↵
```

# #494 选择奖品

→ A 同学要元旦晚会的某一轮抽奖环节选择奖品。他的奖品库中有  $n$  样奖品，编号从 1 到  $n$ ，每个奖品的价值各不相同。这轮抽奖计划颁发 2 个奖品，且这 2 个奖品的价值总数为  $X$  元。

→ 请你帮 A 同学找到合适的 2 个奖品，并输出奖品的编号，按照编号从小到大的顺序输出。

【输入格式】

共 3 行。

第 1 行，1 个整数  $n$ ，表示奖品库中奖品的个数；

第 2 行， $n$  个正整数，每两个正整数之间用一个空格隔开，依次表示编号从 1 到  $n$  的每个奖品的价值；

第 3 行，一个整数  $X$ ，用于颁奖的 2 个奖品的总价值。

注意：输入数据保证有且只有一组解。

【输出格式】

共 1 行，两个用空格隔开的整数，表示选出的 2 个奖品的编号，从小到大的顺序输出。

【输入样例】

4

2 11 7 15

9

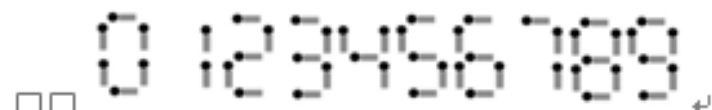
【输出样例】

1 3

```
scanf("%d",&n);  
for(i=0;i<n;i++)  
    scanf("%d",&r[i]);  
scanf("%d",&v);  
for(i=0;i<n-1;i++)  
    for(j=i+1;j<n;j++)  
        if(r[i]+r[j]==v)  
            printf("%d-%d\n",i+1,j+1);
```

# #131 火柴棒等式

给你  $n$  根火柴棒，你可以拼出多少个形如 “ $A + B = C$ ” 的等式？等式中的  $A$ 、 $B$ 、 $C$  是用火柴棒拼出的整数（若该数非零，则最高位不能是 0）。用火柴棒拼数字 0-9 的拼法如图所示：



□□注意：

□□1. 加号与等号各自需要 2 根火柴棒

□□2. 如果  $A \neq B$ ，则  $A + B = C$  与  $B + A = C$  视为不同的等式（ $A$ 、 $B$ 、 $C >= 0$ ）

□□3.  $n$  根火柴棒必须全部用上

输入格式：共一行，有一个整数  $n$  ( $n \leq 24$ )。

输出格式：共一行，有一个整数，表示能拼成的不同等式的数目。

输入样例

18

输出样例

9

$$0 + 4 = 4$$

$$0 + 11 = 11$$

$$1 + 10 = 11$$

$$2 + 2 = 4$$

$$2 + 7 = 9$$

$$4 + 0 = 4$$

$$7 + 2 = 9$$

$$10 + 1 = 11$$

$$11 + 0 = 11$$

```
for (int i = 0; i <= 1000; i++)  
→ for (int j = 0; j <= 1000; j++)  
→ → if (a[i] + a[j] + a[i + j] + 4 == n)  
→ → → total++; //方案数加 1
```

```
for (int i = 1; i <= 2000; i++) {
```

**a[i]=?**

```
}
```

# #134 猜数字

猜数字游戏是文曲星上的一款打发时间的小游戏。游戏的规则是这样的：计算机随机产生一个四位数，然后玩家猜这个四位数是什么。每猜一个数，计算机都会告诉玩家猜对几个数字，其中有几个数字在正确的位置上。比如计算机随机产生的数字为 1122。如果玩家猜 1234,因为 1,2 这两个数字同时存在于这两个数中，而且 1 在这两个数中的位置是相同的，所以计算机告诉玩家猜对了 2 个数字，其中一个在正确的位置。如果玩家猜 1111,那么计算机告诉他猜对 2 个数字，有 2 个在正确的位置。现在给你一段猜数字的过程，你的任务是根据这段对话确定这个四位数是什么。

输入格式

□□ 输入第一行为一个正整数  $N(1 \leq N \leq 10)$ ，表示在这段对话中共有  $N$  次问答（不允许出现重复问答）。在接下来的  $N$  行中，每行三个整数  $A,B,C$ 。游戏者猜这个四位数为  $A$ ，然后计算机回答猜对了  $B$  个数字，其中  $C$  个在正确的位置上。

输出格式

□□ 一行，如果根据这段对话能确定这个四位数，则输出这个四位数，若不能，则输出 "Not sure"。

输入样例

6

4815 2 1

5716 1 0

7842 1 0

4901 0 0

8585 3 3

8555 3 2

输出样例

3585

```

→ for(int N1 = 1; N1 <= 9; N1++) {
→     → for(int N2 = 0; N2 <= 9; N2++) {
→         → for(int N3 = 0; N3 <= 9; N3++) {
→             → for(int N4 = 0; N4 <= 9; N4++) {
→                 → for(int i = 0; i < N; i++) {

```

猜的数与枚举的数进行匹配

每个数是否满足条件

```

→         →         →         →         →     }
→         →         →         →         →     if(flag == N) {
→         →         →         →         →         → printf("%d%d%d%d", N1, N2, N3, N4);
→         →         →         →         →         → flag1 = 1;
→         →         →         →         →     }
→         →         →         →         →     flag = 0;
→         →         →         →         → }

```



## #424 算数比赛

给定一个正整数集合，集合中的数各不相同，请你快速回答：集合中有多少个数，恰好等于本集合中另外两个（不同的）数之和？↵

**输入格式**↵

输入共 2 行，第 1 行包含一个整数  $n$ ，表示给出的正整数个数。↵

第 2 行有  $n$  个正整数，每两个正整数之间用一个空格隔开，表示集合中的正整数。↵

**输出格式**↵

输出共 1 行，包含一个整数，表示本题答案。↵

**样例输入**↵

5↵

5 4 3 2 1↵

**样例输出**↵

3↵

**样例说明**↵

由于  $1+2=3$ ， $1+3=4$ ， $1+4=5$ ， $2+3=5$  故满足测试要求的答案为 3。注意，加数和被加数必须是集合中两个不同的数。

```

for(i=0; i<n; i++)
    for(j=i+1; j<n; j++)
        { for(k=j+1; k<n; k++)
            -
            -
            -
            if(count==1)
            - - - }

```

逐个检查每个数  
 $a[i] == a[j] + a[k]$

优化: 1 排序起步  $j=i+1$   $k=j+1$  2 使用break

```

for(j=i+1; j<n; j++)
    { for(k=j+1; k<n; k++)

        }

```

# #308 猪场分配

```
for(i=0;i<Farm_Count;i++){  
    → if(rongliang[i]<P1_Num) continue;↵  
    → for(j=0;j<Farm_Count;j++){  
        → if(rongliang[j]<P2_Num||j==i) continue;↵  
        → for(k=0;k<Farm_Count;k++){  
            → if(rongliang[k]<P3_Num||k==j||k==i) continue;↵  
            → cost_sum=base_cost[i]+base_cost[j]+base_cost[k];↵  
            → cost_sum=cost_sum+each_cost[i]*P1_Num+each_cost[j]*P2_Num+each_cost[k]*P3_Num;↵  
            → find=0;↵  
            → if(cost_sum<min_cost) → find=1;↵  
            → else-if(cost_sum==min_cost)↵  
            → if(ID[i]<P1_id) → find=1;↵  
            → else if(ID[i]==P1_id)↵  
            → if(ID[j]<P2_id) → find=1;↵  
            → else-if(ID[j]==P2_id && ID[k]<P3_id) → find=1;↵  
            → if(find==1){  
                → min_cost=cost_sum;↵  
                → P1_id=ID[i];P2_id=ID[j];P3_id=ID[k];↵  
            → }  
        → }  
    → }  
}
```

复杂的枚举



中國人民大學  
RENMIN UNIVERSITY OF CHINA



---

## 3. 运行错误

---

# Runtime error

---

- Runtime Error常见出错的原因：
  - 1、**数组开得太小了**，导致访问到了不该访问的内存区域（如int a[4]; a[99999]=9;）
  - 2、发生**除零**错误
  - 3、**大数组定义**在函数内,导致程序栈区耗尽（int a[1000000000];）
  - 4、指针用错了，导致访问到不该访问的内存区域
  - 5、还有可能是程序抛出了未接收的异常

**超时：Time Limit Exceeded**

两个字符串  $S1$ 、 $S2$  的差定义如下： $S1$  和  $S2$  的按字典序比较大小。若  $S1$  和  $S2$  相等，则输出为 0；若它们不相等，则输出其第一个不同字符的 ASCII 码的差值，且如果  $S1 > S2$ ，则差值为正；如果  $S1 < S2$ ，则差值为负。

## #288 字符串之差

输入格式

□□ 共 2 行，第 1 行是字符串  $S1$ ，第 2 行是字符串  $S2$ 。注意：输入数据保证每一个字符串不是另一个的前缀，且长度在 100 以内。

输出格式

□□ 1 行仅 1 个整数，表示字符串之差。

输入样例

java basic

输出样例

8

#741620 #288 字符串之差 Runtime Error

```
int main()
{
    char s[101], t[101]; // 如果长度为 100，就会 runtime-error
    int a = 0, i;
    gets(s); gets(t);
    for (i = 0; i <= 101; i++)
        if (s[i] == t[i]) continue;
        else { a = s[i] - t[i]; break; }
    printf("%d", a);
    return 0;
}
```

测试点 #3

2

得分：0

输入文件 (data4.in)

zkMwSyBHefc0aKmcDwStujvaQ5kxaIvmBgDjWxKVewxDDNaBKymOzbN1lKDIdgXgUy1QfVooPMq/  
<102 bytes omitted>

输出文件 (data4.out)

-25

选手输出

# Compile Error

#407279	#1 输出Hello world	</> Compile Error	0	0 ms
#407278	#307 生辰八字	✖ Wrong Answer	10	60 ms

#407279      #1 输出Hello world      </> Compile Error      0      0 ms      0 KB      [cpp /](#)

/sandbox/1/a.cpp: In function 'int main()':

/sandbox/1/a.cpp:3:1: **error:** 'printf' was not declared in this scope

3 | **printf**("good")

| ^~~~~~

/sandbox/1/a.cpp:1:1: **note:** 'printf' is defined in header '<cstdio>'; did you forget to '#include <cstdio>'?

+++ |+#include <cstdio>

1 | int main()

C++

GCC 8.2.0

C++ 11

GCC 8.2.0

```
1  int main()
2  {
3  printf("good")
4
5  }
```

# Wrong Answer ( #90 刷题高手 )

老师给了一个题目的列表，要求同学们在 YOJ 上做题。一段时间后，老师想统计一下同学们刷题的数量，找出刷题高手。

【输入格式】↵

第一行，多个整数，第一个整数  $n$  表示老师要求的题目个数( $1 \leq n \leq 100$ )。后面有  $n$  个整数，表示老师要求的题号。↵

第二行，两个整数  $m$  和  $k$ ，表示有  $m$  个学生 ( $1 \leq m \leq 100$ )，以及老师希望找出刷题最多的  $k$  个同学 ( $1 \leq k \leq m$ )。↵

后面  $m$  行，每行格式如下：↵

一个正整数  $sno$ ，表示同学的学号，8 位数字；↵

一个正整数  $p$ ，表示该同学刷了多少题， $0 \leq p \leq 100$ ；↵

后面  $p$  个正整数，表示该同学成功通过的题目编号。↵

【输出格式】↵

输出为一行，至少  $k$  个整数，表示刷题数量最多的前  $k$  名同学的学号，按刷题数量从高到低排列，输出的学号之间用一个空格分隔。↵

注意↵

1) 若同学们刷的题目不是老师所要求的，则不计入成绩。↵

2) 允许并列。例如若  $k=3$ ，且有多位同学并列排名第 2，则成绩第 3 的同学也应该输出。当有并列情况时，有可能出现所有学生的排名均小于  $k$ 。此时，将所有同学学号按成绩高低输出即可。↵

3) 出现并列时，学号较小的同学先输出。↵

【输入样例】↵

~~5~~·~~1~~·~~3~~·~~5~~·~~7~~·~~9~~·//老师要求的题号↵

~~3~~·~~1~~↵

10016655·~~5~~·2·4·6·8·10↵

10055236·~~1~~·5↵

10001799·~~4~~·1·2·4·6↵

【输出样例】↵

10001799·10055236↵



```
#include <stdio.h>
struct Student {
    → int sno; //学号
    → int p; //题目数
    → int t[102]; //题号
    → int sum; //有效题目数
} a[102];
```

```
int main() {
    → int n, m, k, i, j;
    → int tec[102]; //老师要求的题号
```

```
    → scanf("%d", &n);
    → for (i = 1; i <= n; i++) {
        → scanf("%d", &tec[i]);
        → int x;
        → scanf("%d%d", &m, &k);
        → for (j = 1; j <= m; j++) {
            → scanf("%d%d", &a[j].sno, &a[j].p);
            → for (j = 1; j <= a[j].p; j++) {
                → scanf("%d", &a[j].t[j]);
                → for (x = 1; x <= n; x++) {
                    → if (a[j].t[j] == tec[x]) → a[j].sum++;
                }
            }
        }
    }
```

```
int tsum, tsno; //这一步把 sum 有效题目数从大到小排，同时学号从小到大排
for (i = 1; i <= m - 1; i++) {
```

```
    → for (j = 1; j <= m - 1 - i; j++) {
        → if (a[j].sum < a[j + 1].sum || a[j].sum == a[j + 1].sum && a[j].sno > a[j + 1].sno) {
            → tsum = a[j].sum; a[j].sum = a[j + 1].sum; a[j + 1].sum = tsum;
            → tsno = a[j].sno; a[j].sno = a[j + 1].sno; a[j + 1].sno = tsno;
        }
    }
```

```
int y = 1;
for (i = 1; i <= k; i++) { //逐轮输出，一共 k 轮
```

```
    → printf("%d", a[y].sno); //输出每一轮的第一名学号
    → for (j = y + 1; j <= m; j++) //判断第一名后面有没有一样的，一样的继续输出
        → if (a[j].sum == a[y].sum) {
            → y++; printf("%d", a[j].sno);
        }
    → else if (a[j].sum != a[y].sum) {
        → y++; break;
    }
```

```
}
```

**我的代码只能得60分，一些测试点通不过，不知道错误出在哪里？**

#604752

#90 刷题高手

✖ Wrong Answer

▼ 测试点 #0

☰ ✖ 2

输入文件 (data0.in)

```
5 1 3 5 7 9
3 1
10016655 5 2 4 6 8 10
10055236 1 5
10001799 4 1 2 4 6
```

输出文件 (data0.out)

```
10001799 10055236
```

选手输出

```
10055236
```

▼ 测试点 #1

☰ ✖ 2

得分: 0

用时: 8 ms

```
freopen("#90/data0.in", "r", stdin);
```

## 第一步：先在输出前插入排序后的输出，看排序是否正确

```
/*for (i = 1; i <= m; i++) //逐轮输出，一共k轮
```

```
    printf("%d %d \n", a[i].sno, a[i].sum); //输出每一轮的第一名学号
```

```
    printf("+++上面是排序后的结果++++ \n");*/
```

结果发现，排序有问题，如下是第0个测试点

```
10055236 1
10016655 0
10001799 1
+++上面是排序后的结果++++
```

## 第二步：检查排序

```
for (i = 1; i <= m - 1; i++) {
```

```
→ for (j = 1; j <= m - i - 1; j++) //应该修改成 for (j = 1; j <= m - i; j++) {
```

```
→     if (a[j].sum < a[j + 1].sum || (a[j].sum == a[j + 1].sum && a[j].sno > a[j + 1].sno)) {
```

```
→         tsum = a[j].sum; a[j].sum = a[j + 1].sum; a[j + 1].sum = tsum;
```

```
→         tsno = a[j].sno; a[j].sno = a[j + 1].sno; a[j + 1].sno = tsno;
```

```
→     }
```

修改后从原先60分变成得90分，但还有一个测试点没有通过

**第三步：**通过通过对比data8.in测试点，本程序的输出与data8.out文件的输出差别，发现肯定是输出有问题，究其原因，是最后一个多输出了

```
10049645 10027157 10004098 10014748 10015665 10049994 10060437 10064361 10016742 10024081 10031733 10048332 10004657
10006748 10026909 10036790 10040611 10041638 10050622 10027047 10041002 10056851 10010188 10049382 10055339 10058745
10029904 10062459 10021022 10048269 10048426 10003729 10024049 10026576 10052272 10056482 10058357 10060905 10014838
10015937 10024245 10030274 10037759 10063588 10022677 10037784 10043709 10061844 10012750 10018925 10048223 10058073
10061111 10050678 10051435 10052581 10060246 10007821 10016787 10045221 10047385 10047708 10009261 10026299 10028184
10040403 10040477 10046254 10040464 10040296 10010685 10038160 10043024 10043905 10050793 10060020 10060849 10014459
10016508 10034344 10041359 10062118 10062173 10063581 10008890 10034838 10041409 10054577 10056632 10064697 10064697
10064697 10064697 10064697 10064697 10064697 10064697 10064697 10064697 10064697 10064697 10064697 10064697 10064697
10064697 10064697 10064697 10064697 10064697 10064697 10064697 10064697 10064697 10064697 10064697 10064697 10064697
10064697 10064697 10064697 10064697 10064697 10064697 10064697 10064697 10064697 10064697 10064697 10064697 10064697
10064697 10064697 10064697 10064697 10064697 10064697 10064697 10064697 10064697 10064697 10064697 10064697 10064697
10064697 10064697 10064697
```

**第四步：**再检查输出，结果发现第8个测试点是 $m=90, k=78$ ，事实上当输出不到 $k=78$ 时，90人都已经全部输出完了，事实上当输出到第22个不同分数时，所有人都已经输出，但按作者程序，还是继续循环，所以应该加一个结束循环的语句 `if (y == m) break;` //需要加上这一条

```
for (i = 1; i <= k; i++) { //逐个分数输出，一共k轮
    输出第i个分数
}
```

```
int y = 1;
```

```
for (i = 1; i <= k; i++) { // 逐轮输出，一共 k 轮
```

```
→ printf("%d", a[y].sno); // 输出每一轮的第一名学号
```

```
→ for (i = 1; i <= k; i++) { // 逐轮输出，一共 k 轮
```

```
→ → printf("%d", a[y].sno); // 输出每一轮的第一名学号
```

```
→ → for (j = y + 1; j <= m; j++) // 判断第一名后面有没有一样的，一样的继续输出
```

```
→ → → if (a[j].sum == a[y].sum) {
```

```
→ → → → y++; printf("%d", a[j].sno); }
```

```
→ → → else if (a[j].sum != a[y].sum) {
```

```
→ → → → y++; break; }
```

```
→ }
```

```
→ if (y == n) break; // 需要加上这一条
```

```
}
```

```
for (i = 0, j = 0; i < top_k && j < stu_num; j++) {
```

```
→ printf("%d", sno[j]);
```

```
→ if (ture[j] > ture[j + 1]) i++; }
```

```

int n, m, k, p[200], a, b, c, q[200], w[200];
long sno[200];
scanf("%d", &n);
for (a = 1; a <= n; a++)
    → scanf("%d", &q[a]);
scanf("%d %d", &m, &k);
for (a = 1; a <= m; a++) {
    → scanf("%d", &sno[a]);
    → scanf("%d", &p[a]);
    → int u = p[a];
    → for (b = 1; b <= u; b++) {
        → scanf("%d", &w[b]);
        → l = 0;
        → for (c = 1; c <= n; c++)
            → if (w[b] != q[c]) → l++;
        → if (l == n) p[a]--;
    }
}

```

```

for (a = 1; a <= m - 1; a++)
    → for (b = 1; b <= m - a; b++)
        → if (p[b] < p[b + 1]) {
            → int t;
            → t = p[b];
            → p[b] = p[b + 1];
            → p[b + 1] = t;
            → t = sno[b];
            → sno[b] = sno[b + 1];
            → sno[b + 1] = t;
        }
for (int i = 0, a = 1; a <= m - 1; a++)
    → if (p[a] == p[a + 1]) {
        → i++;
        → for (b = a; b >= a - i + 1; b--)
            → if (sno[b] > sno[b + 1]) {
                → int y;
                → y = sno[b];
                → sno[b] = sno[b + 1];
                → sno[b + 1] = y;
            }
        → else → i = 0;
    }

```

```

int fsx;
for (a = 1; a <= k; a++) {
    → if (p[a] == p[a + 1]) → k++;
    → fsx = p[a];
}
for (a = 1; a <= m; a++)
    → if (p[a] >= fsx) → printf("%ld", sno[a]);

```

测试点 #8

✖ 2

得分: 0

用时: 8 ms

输入文件 (data8.in)

90 383 466 443 467 381 366 426 355 18 113 30 46 328 358 300 157 455 436 164 447 54 205 481 285 271 4  
<19293 bytes omitted>

输出文件 (data8.out)

10049645 10027157 10004098 10014748 10015665 10049994 10060437 10064361 10016742 10024081 10031733 1  
<709 bytes omitted>

选手输出

10049645 10027157 10004098 10014748 10015665 10049994 10060437 10064361 10016742 10024081 10031733 10048332 10004657 10006748 10  
<43 bytes omitted>

```
for (a = 1; a <= k; a++) {.
```

```
→ if (p[a] == p[a + 1]) → k++;.
```

```
→ fsx = p[a];.
```

```
→ printf("a=%d k=%d p[a]=%d fsx=%d\n", a, k, p[a], fsx); → .. }.
```

```
a=1 k=78 p[a]=25 fsx=25
a=2 k=78 p[a]=22 fsx=22
a=3 k=79 p[a]=21 fsx=21
a=4 k=79 p[a]=21 fsx=21
a=5 k=80 p[a]=18 fsx=18
a=6 k=81 p[a]=18 fsx=18
a=7 k=82 p[a]=18 fsx=18
a=8 k=82 p[a]=18 fsx=18
a=9 k=83 p[a]=17 fsx=17
a=10 k=84 p[a]=17 fsx=17
a=11 k=85 p[a]=17 fsx=17
a=12 k=85 p[a]=17 fsx=17
a=13 k=86 p[a]=16 fsx=16
a=14 k=87 p[a]=16 fsx=16
a=15 k=88 p[a]=16 fsx=16
a=16 k=89 p[a]=16 fsx=16
a=17 k=90 p[a]=16 fsx=16
a=18 k=91 p[a]=16 fsx=16
a=19 k=91 p[a]=16 fsx=16
a=20 k=92 p[a]=15 fsx=15
a=21 k=93 p[a]=15 fsx=15
a=22 k=93 p[a]=15 fsx=15
a=23 k=94 p[a]=14 fsx=14
a=24 k=95 p[a]=14 fsx=14
a=25 k=96 p[a]=14 fsx=14
a=26 k=96 p[a]=14 fsx=14
a=27 k=97 p[a]=13 fsx=13
a=28 k=97 p[a]=13 fsx=13
```

```
a=74 k=133 p[a]=2 fsx=2
a=75 k=134 p[a]=2 fsx=2
a=76 k=135 p[a]=2 fsx=2
a=77 k=135 p[a]=2 fsx=2
a=78 k=136 p[a]=1 fsx=1
a=79 k=137 p[a]=1 fsx=1
a=80 k=138 p[a]=1 fsx=1
a=81 k=139 p[a]=1 fsx=1
a=82 k=140 p[a]=1 fsx=1
a=83 k=141 p[a]=1 fsx=1
a=84 k=141 p[a]=1 fsx=1
a=85 k=142 p[a]=0 fsx=0
a=86 k=143 p[a]=0 fsx=0
a=87 k=144 p[a]=0 fsx=0
a=88 k=145 p[a]=0 fsx=0
a=89 k=146 p[a]=0 fsx=0
a=90 k=147 p[a]=0 fsx=0
a=91 k=148 p[a]=0 fsx=0
a=92 k=149 p[a]=0 fsx=0
a=93 k=150 p[a]=0 fsx=0
a=94 k=151 p[a]=0 fsx=0
a=95 k=152 p[a]=0 fsx=0
a=96 k=153 p[a]=0 fsx=0
a=97 k=154 p[a]=0 fsx=0
a=98 k=155 p[a]=0 fsx=0
a=99 k=156 p[a]=0 fsx=0
a=100 k=157 p[a]=0 fsx=0
a=101 k=158 p[a]=0 fsx=0
```

```
a=177 k=233 p[a]=0 fsx=0
a=178 k=233 p[a]=-1541905319 fsx=-1541905319
a=179 k=233 p[a]=32764 fsx=32764
a=180 k=234 p[a]=0 fsx=0
a=181 k=235 p[a]=0 fsx=0
a=182 k=236 p[a]=0 fsx=0
a=183 k=237 p[a]=0 fsx=0
a=184 k=238 p[a]=0 fsx=0
a=185 k=238 p[a]=0 fsx=0
a=186 k=238 p[a]=6413658 fsx=6413658
a=187 k=238 p[a]=0 fsx=0
a=188 k=238 p[a]=6413541 fsx=6413541
a=189 k=239 p[a]=0 fsx=0
a=190 k=240 p[a]=0 fsx=0
a=191 k=241 p[a]=0 fsx=0
a=192 k=242 p[a]=0 fsx=0
a=193 k=243 p[a]=0 fsx=0
a=194 k=244 p[a]=0 fsx=0
a=195 k=244 p[a]=0 fsx=0
a=196 k=244 p[a]=256 fsx=256
a=197 k=245 p[a]=0 fsx=0
a=198 k=246 p[a]=0 fsx=0
a=199 k=246 p[a]=0 fsx=0
a=200 k=246 p[a]=33 fsx=33
a=201 k=246 p[a]=10049645 fsx=10049645
a=202 k=246 p[a]=10027157 fsx=10027157
a=203 k=246 p[a]=10004098 fsx=10004098
a=204 k=246 p[a]=10014748 fsx=10014748
a=205 k=246 p[a]=10015665 fsx=10015665
a=206 k=246 p[a]=10049994 fsx=10049994
a=207 k=246 p[a]=10060437 fsx=10060437
a=208 k=246 p[a]=10064361 fsx=10064361
```



中國人民大學  
RENMIN UNIVERSITY OF CHINA



---

## 4. 基本模块

---



# 输入格式

3-5//以整数形式输入↵

2017101000-10-2.1-1.2-4.3-2.4-2.5-5.1-5.2-1.6-4.2-4.4↵

2017101001-9-2.1-2.2-2.5-3.2-1.1-1.3-4.3-4.4-5.2↵

2017101002-10-1.4-1.5-2.1-2.2-3.4-3.4-4.1-4.6-5.4-5.5↵

```
....scanf("%d-%d",&n,&k);↵
....for(s=0,i=0;i<n;i++)↵
....{...scanf("%s",&id[i]);↵
.....scanf("%d",&num[i]);↵
.....for(j=0;j<num[i];j++)↵
.....{...scanf("%d.%d",&date[s],&class1[s]);↵
.....s+=1;...}↵
....}↵
```

2-1//以整数形式输入↵

1-1-1-1-1-1-1-1//以整数形式输入↵

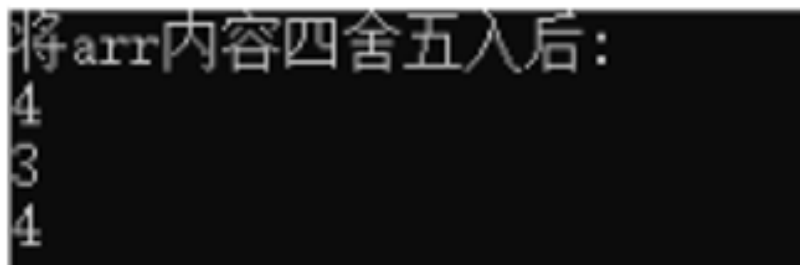
1012345678-1-1-1-1-1-1-1-1//以整数形式输入↵

1087654321-1-1-1-1-8-8-8-8//以整数形式输入↵

```
....scanf("%d-%d",&n,&k);↵
....for(i=1;i<=8;i++)↵
.....scanf("%d",&w[i]);↵
....for(j=0;j<n;j++){↵
.....scanf("%d",&m[j][0]);↵
.....for(i=1;i<=8;i++)↵
.....scanf("%d",&m[j][i]);}↵
```

# 输出

```
int main() {  
    → float arr[3] = {3.5, 3.4, 3.7};  
    → printf("将 arr 内容四舍五入后:\n");  
    → for (int i = 0; i < 3; i++)  
        → printf("%d\n", (int)(arr[i] + 0.5));  
}
```



```
将arr内容四舍五入后:  
4  
3  
4
```

long int , long long int, unsigned long;

```
printf("%lld\n", x);
```

```
printf("%lu\n", x);
```

```
float f = 1.67;
```

```
printf("%d", (int)(f));
```

# 排序

**$a[0] \dots a[n-1]$   $n$  个数排序**

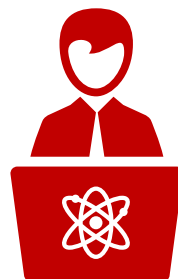
```
for (i = 0; i < n - 1; i++) // i = 0, 1, 2, ..., n-2 【n-1 轮】
    → for (j = 0; j < n - 1 - i; j++) // j = 0, 1, 2, ..., n-2-i 【n-1-i 次 PK】
        → if (a[j] < a[j + 1])
            → { tmp = a[j]; a[j] = a[j + 1]; a[j + 1] = tmp; }
```

**$a[1] \dots a[n]$   $n$  个数排序**

```
for (i = 1; i <= n - 1; i++) // i = 1, 2, 3, ..., n-1
    → for (j = 1; j <= n - i; j++) // j = 1, 2, ..., n-i
        → if (a[j] < a[j + 1])
            → { tmp = a[j]; a[j] = a[j + 1]; a[j + 1] = tmp; }
```



中國人民大學  
RENMIN UNIVERSITY OF CHINA



---

# 谢谢大家！

---

