

## 标 题: 2007 考研上机考试题目 1--最小长方形(35 分)

### 题目要求:

给定一系列 2 维平面点的坐标(x, y), 其中 x 和 y 均为整数, 要求用一个最小的长方形框将所有点框在内。长方形框的边分别平行于 x 和 y 坐标轴, 点落在边上也算是被框在内。

具体的输入输出格式规定如下:

**输入格式:** 测试输入包含若干测试用例, 每个测试用例由一系列坐标组成, 每对坐标占一行, 其中|x|和|y|小于 231; 一对 0 坐标标志着一个测试用例的结束。注意(0, 0)不作为任何一个测试用例里面的点。一个没有点的测试用例标志着整个输入的结束。

**输出格式:** 对每个测试用例, 在 1 行内输出 2 对整数, 其间用一个空格隔开。第 1 对整数是长方形框左下角的坐标, 第 2 对整数是长方形框右上角的坐标。

### 输入样例:

12 56

23 56

13 10

0 0

12 34

0 0

0 0

### 输出样例:

12 10 23 56

12 34 12 34

-----

```
#include <iostream>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    int x1,y1,x2,y2,x,y;
```

```
    while(cin>>x>>y&&(x||y)){
```

```
        x1=x2=x; y1=y2=y;
```

```
        while(cin>>x>>y&&(x||y))
```

```
            x1=x1<x?x1:x,y1=y1<y?y1:y,x2=x2>x?x2:x,y2=y2>y?y2:y;
```

```
        cout<<x1<<" "<<y1<<" "<<x2<<" "<<y2<<endl;
```

```
    }
```

```
    return 0;
```

```
}
```

## 标 题: 2007 考研上机考试题目 2--统计字符(25 分)

### 题目要求:

统计一个给定字符串中指定的字符出现的次数

具体的输入输出格式规定如下:

**输入格式:** 测试输入包含若干测试用例, 每个测试用例包含 2 行, 第 1 行为一个长度不超过 5 的字符串, 第 2 行为一个长度不超过 80 的字符串。注意这里的字符串包含空格, 即空格也可能是要求被统计的字符之一。当读到 '#' 时输入结束, 相应的结果不要输出。

**输出格式:** 对每个测试用例, 统计第 1 行中字符串的每个字符在第 2 行字符串中出现的次数, 按如下格式输出:

c0 n0

c1 n1

c2 n2

...

其中  $c_i$  是第 1 行中第  $i$  个字符,  $n_i$  是  $c_i$  出现的次数。

**输入样例:**

I

THIS IS A TEST

i ng

this is a long test string

#

**输出样例:**

I 2

i 3

5

n 2

g 2

注: 第 2 个测试用例中, 空格也是被统计的字符之一。

-----

```
#include <iostream>
```

```
#include <string>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    string pat, line;
```

```
    while(getline(cin, pat)&&pat!="#") {
```

```
        int co[5]={0, 0, 0, 0, 0};
```

```
        getline(cin, line);
```

```
        for(int i=0, j; i<line.size(); ++i) {
```

```
            for(j=0; j<pat.size()&&pat[j]!=line[i]; ++j);
```

```
            co[j]+=j<pat.size();
```

```
        }
```

```
        for(int i=0; i<pat.size(); ++i) {
```

```
            cout<<pat[i]<<" "<<co[i]<<endl;
```

```
        }
```

```
    }
```

```
    return 0;  
}
```

### 标 题: 2007 考研上机考试题目 3--游船出租(18 分)

#### 题目要求:

现有公园游船租赁处请你编写一个租船管理系统。当游客租船时, 管理员输入船号并按下 S 键, 系统开始计时; 当游客还船时, 管理员输入船号并按下 E 键, 系统结束计时。船号为不超过 100 的正整数。当管理员将 0 作为船号输入时, 表示一天租船工作结束, 系统应输出当天的游客租船次数和平均租船时间。

注意: 由于线路偶尔会有故障, 可能出现不完整的纪录, 即只有租船没有还船, 或者只有还船没有租船的纪录, 系统应能自动忽略这种无效纪录。

具体的输入输出格式规定如下:

**输入格式:** 测试输入包含若干测试用例, 每个测试用例为一整天的租船纪录, 格式为

船号 (1~100) 键值 (S 或 E) 发生时间 (小时:分钟)

每一天的纪录保证按时间递增的顺序给出。当读到船号为-1 时, 全部输入结束, 相应的结果不要输出。

**输出格式:** 对每个测试用例输出 1 行, 即当天的游客租船次数和平均租船时间(以分钟为单位的精确到个位的整数时间)。

#### 输入样例:

```
1 S 08:10  
2 S 08:35  
1 E 10:00  
2 E 13:16  
0 S 17:00  
0 S 17:00  
3 E 08:10  
1 S 08:20  
2 S 09:00  
1 E 09:20  
0 E 17:00  
-1
```

#### 输出样例:

```
2 196  
0 0  
1 60
```

```
-----  
#include <iostream>  
#include <cstdio>  
using namespace std;
```

```
int main()  
{
```

```

int n, t[100], h, m, co=0, to=0;
char cmd[10], tt[10];
for(memset(t, -1, sizeof(t)); cin >> n && n >= 0; ) {
    cin >> cmd >> tt;
    if(n > 0) {
        sscanf(tt, "%d:%d", &h, &m);
        if(cmd[0] == 'S') t[n-1] = h*60+m;
        else if(t[n-1] >= 0) ++co, to += h*60+m-t[n-1], t[n-1] = -1;
    } else {
        cout << co << " " << (co?to/co+(to%co*2>=co):0) << endl;
        co = to = 0;
    }
}
return 0;
}

```

### 标 题：2007 考研上机考试题目 4--EXCEL 排序(18 分)

#### 题目要求：

Excel 可以对一组纪录按任意指定列排序。现请你编写程序实现类似功能。

具体的输入输出格式规定如下：

**输入格式：**测试输入包含若干测试用例。每个测试用例的第 1 行包含两个整数 N ( $\leq 100000$ ) 和 C，其中 N 是纪录的条数，C 是指定排序的列号。以下有 N 行，每行包含一条学生纪录。每条学生纪录由学号（6 位数字，同组测试中没有重复的学号）、姓名（不超过 8 位且不包含空格的字符串）、成绩（闭区间 [0, 100] 内的整数）组成，每个项目间用 1 个空格隔开。当读到 N=0 时，全部输入结束，相应的结果不要输出。

**输出格式：**对每个测试用例，首先输出 1 行 “Case i:”，其中 i 是测试用例的编号（从 1 开始）。随后在 N 行中输出按要求排序后的结果，即：当 C=1 时，按学号递增排序；当 C=2 时，按姓名的非递减字典序排序；当 C=3 时，按成绩的非递减排序。当若干学生具有相同姓名或者相同成绩时，则按他们的学号递增排序。

#### 输入样例：

```

3 1
000007 James 85
000010 Amy 90
000001 Zoe 60
4 2
000007 James 85
000010 Amy 90
000001 Zoe 60
000002 James 98
4 3
000007 James 85
000010 Amy 90

```

```
000001 Zoe 60
000002 James 90
0 0
```

输出样例:

Case 1:

```
000001 Zoe 60
000007 James 85
000010 Amy 90
```

Case 2:

```
000010 Amy 90
000002 James 98
000007 James 85
000001 Zoe 60
```

Case 3:

```
000001 Zoe 60
000007 James 85
000002 James 90
000010 Amy 90
```

```
-----

#include <iostream>
#include <string>
#include <algorithm>
#include <cstdio>
using namespace std;

int n, c, aa[100010][2], ref[100010], ca=0;
char ss[100010][10];

bool cmp(int a, int b)
{
    if(c==3&&aa[a][1]!=aa[b][1])return aa[a][1]<aa[b][1];
    if(c==2&&strcmp(ss[a],ss[b]))return strcmp(ss[a],ss[b])<0;
    return aa[a][0]<aa[b][0];
}

int main()
{
    while(cin>>n>>c&&n) {
        printf("Case %d:\n", ++ca);
        for(int i=0; i<n; ++i)
            cin>>aa[i][0]>>ss[i]>>aa[i][1], ref[i]=i;
        sort(ref, ref+n, cmp);
        for(int i=0; i<n; ++i) {
```

```

printf("%06d %s %d\n", aa[ref[i]][0], ss[ref[i]], aa[ref[i]][1]);
    }
}
return 0;
}

```

## 标 题: 2007 考研上机考试题目 5--畅通工程(12 分)

### 题目要求:

省政府“畅通工程”的目标是使全省任何两个村庄间都可以实现公路交通(但不一定有直接的公路相连,只要能间接通过公路可达即可)。经过调查评估,得到的统计表中列出了有可能建设公路的若干条道路的成本。现请你编写程序,计算出全省畅通需要的最低成本。

具体的输入输出格式规定如下:

**输入格式:** 测试输入包含若干测试用例。每个测试用例的第 1 行给出评估的道路条数  $N$ 、村庄数目  $M$  ( $< 100$ ) ; 随后的  $N$

行对应村庄间道路的成本, 每行给出一对正整数, 分别是两个村庄的编号, 以及此两村庄间道路的成本(也是正整数)。为简单起见, 村庄从 1 到  $M$  编号。当  $N$  为 0 时, 全部输入结束, 相应的结果不要输出。

**输出格式:** 对每个测试用例, 在 1 行里输出全省畅通需要的最低成本。若统计数据不足以保证畅通, 则输出“?”。

### 输入样例:

```

3 3
1 2 1
1 3 2
2 3 4
1 3
2 3 2
0 100

```

### 输出样例:

```

3
?

```

```

-----
#include <iostream>
using namespace std;
int main()
{
    int n,m,aa[110][110],v[110],min[110],ret,i,j,k,inf=(100<<16)+100;
    while(cin>>m>>n&&m) {
        memset(aa,100,sizeof(aa));
        while(m--){
            cin>>i>>j>>k;
            aa[j-1][i-1]=aa[i-1][j-1]=k;
        }
    }
}

```

```

for (ret=i=0; i<n; i++) min[i]=inf, v[i]=0;
for (min[j]=0; j<n; j++) {
    for (k=-1, i=0; i<n; i++)
        if (!v[i] && (k==-1 || min[i]<min[k])) k=i;
    for (v[k]=1, ret+=min[k], i=0; i<n; i++)
        if (!v[i] && aa[k][i]<min[i])
            min[i]=aa[k][i];
}
if (ret<inf) cout<<ret<<endl;
else cout<<"?"<<endl;
}
return 0;
}

```

标 题: 2007 考研上机考试题目 6--最大报销额 (12 分)

#### 题目要求:

现有一笔经费可以报销一定额度的发票。允许报销的发票类型包括买图书 (A 类)、文具 (B 类)、差旅 (C 类), 要求每张发票的总额不得超过 1000 元, 每张发票上, 单项物品的价值不得超过 600 元。现请你编写程序, 在给出的一堆发票中找出可以报销的、不超过给定额度的最大报销额。

具体的输入输出格式规定如下:

**输入格式:** 测试输入包含若干测试用例。每个测试用例的第 1 行包含两个正数 Q 和 N, 其中 Q 是给定的报销额度, N ( $\leq 30$ ) 是发票张数。随后是 N 行输入, 每行的格式为:

m Type\_1:price\_1 Type\_2:price\_2 ... Type\_m:price\_m

其中正整数 m 是这张发票上所开物品的件数, Type\_i 和 price\_i 是第 i 项物品的种类和价值。物品种类用一个大写英文字母表示。当 N 为 0 时, 全部输入结束, 相应的结果不要输出。

**输出格式:** 对每个测试用例输出 1 行, 即可以报销的最大数额, 精确到小数点后 2 位。

#### 输入样例:

```

200.00 3
2 A:23.50 B:100.00
1 C:650.00
3 A:59.99 A:120.00 X:10.00
1200.00 2
2 B:600.00 A:400.00
1 C:200.50
1200.50 3
2 B:600.00 A:400.00
1 C:200.50
1 A:100.00
100.00 0

```

#### 输出样例:

```

123.50

```

1000.00

1200.50

```
-----
#include <iostream>
#include <cstdio>
using namespace std;

double sum, a[40], b, bb;
int n, nn, m;
char ss[110];

double dfs(int cur, double v)
{
    if (cur >= n) return v;
    double ret = 0, ret2 = dfs(cur + 1, v);
    if (v + a[cur] <= sum + 1e-8) ret = dfs(cur + 1, v + a[cur]);
    return ret > ret2 ? ret : ret2;
}

int main()
{
    cout.precision(2); cout.setf(ios::fixed);
    while (cin >> sum >> nn && nn) {
        for (n = 0; nn--; n += a[n] <= 1000 + 1e-8 && bb <= 600 + 1e-8) {
            for (cin >> m, a[n] = bb = 0; m--; ) {
                cin >> ss;
                sscanf(ss + 2, "%lf", &b);
                bb = bb > b ? bb : b;
                a[n] += (ss[0] >= 'A' && ss[0] <= 'C' ? b : 1001);
            }
        }
        cout << dfs(0, 0) << endl;
    }
    return 0;
}
```