

## Noip2017 练习赛

|       |            |           |         |         |
|-------|------------|-----------|---------|---------|
| 题目名称  | 足球联赛       | 炮击坦克      | 冰岛      | 搭配购买    |
| 存盘文件名 | soccer     | catch     | ice     | buy     |
| 输入文件名 | soccer.in  | catch.in  | ice.in  | buy.in  |
| 输出文件名 | soccer.out | catch.out | ice.out | buy.out |
| 时限    | 1s         | 1s        | 2s      | 1s      |
| 内存限制  | 64M        | 64M       | 64M     | 64M     |

**【注意事项】：**请自行完成题目，切勿讨论。

### 题 1 足球联赛

#### 【问题描述】

在足球比赛中，各国主要以联赛方式进行：

联赛中，每个队伍都会两两进行主客场交手，获得胜利的队伍全取 3 分；失利的队伍一分不得；若双方战平，则各取 1 分。而统计球队积分，则难坏了小明，他想要请你开发一个软件，帮他统计一下整个赛季结束之后各个球队的积分情况，以便向外界公布。

#### 【输入】

输入文件名为 soccer.in，有多组测试数据：

第一行，包含一个整数 Num，表示测试数据的个数。（ $1 \leq \text{Num} \leq 10$ ）

每组测试数据，

第一行一个整数 N，表示共有 N 个球队。 $2 \leq N \leq 50$ 。

接下来 N 行，表示每个球队在主场作战的情况。每行共 N 个元素，“W”表示主队获胜，“D”表示双方战平，“L”表示主队失利。自己和自己用“-”表示。

#### 【输出】

输出文件 soccer.out 共 Num 行，

每行 N 个数，表示 N 个球队的积分情况，中间用空格分隔，最后一个球队后不要有空格。

#### 【输入输出样例 1】

| soccer.in | soccer.out |
|-----------|------------|
| 2         | 6 6 6      |
| 3         | 5 2 8      |
| -WW       |            |
| W-W       |            |
| WW-       |            |
| 3         |            |
| -DD       |            |
| L-L       |            |
| WD-       |            |

#### 【输入输出样例 2】

| soccer.in | soccer.out                                |
|-----------|-------------------------------------------|
| 2         | 14 7 12 8 10                              |
| 5         | 72 62 41 41 83 63 53 35 86 50 90 32 34 41 |
| -DWW      | 45 36 51 32 51 45                         |

|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |  |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|
| L-WLL<br>DD-WD<br>DDL-L<br>DDL-<br>20<br>-LWWLWDLDWWWWDDWDW<br>D-WWLDLWDLWDDWLWDD<br>LL-DLDWDLDLWWWLWDDW<br>LDD-LLLDLWLWWWWDWL<br>LWWW-DWDLWDWDWWWWDW<br>DLLWD-WWLLDDDLWWDWW<br>WWLWL-LLDWWWWDWWLW<br>LLLLLDW-LDLWDDLLLDWL<br>DWWWDDDD-DWWWDDWWWWDW<br>WWWLWLLWL-LWWWLWLLW<br>DWWWLWLLW-WDWWWLWLLW<br>DDDLWLLWLLW-DDDWLWDD<br>LWLWDLWLLDLW-DDDWDD<br>LLWWLWDDWLWL-WWDLW<br>WWWLWLLWLLWDD-WWLLW<br>DLWLLWLLWLLW-DWLLW<br>DLWWWLWLLWDDWL-WWD<br>LLDDLLWLLWLLWLLW-WW<br>LLWLLWLLWLLWDDWLW-W<br>LLWDLWDDWLLWDDWLL- |  |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|

## 题 2 炮击坦克

### 【问题描述】

在一个坐标轴上，有  $M$  辆坦克，第  $i$  辆坦克在时刻 0 处于  $pos[i]$  ( $pos[i]>0$ )， $speed[i]$  个单位，在时刻  $k$  就处于  $pos[i]+speed[i]*k$ 。

原点上有一炮台。炮台有  $N$  颗炮弹，在时刻 0 开始就可以发射炮弹，而且发射的顺序是你来确定的，每次只能发射一颗，一颗炮弹只能用一次。每个炮弹都有一个休息时间  $rest[i]$ ，如果在某次发射了第  $i$  颗炮弹，要间隔  $rest[i]$  后才能在发射。一颗炮弹只能消灭范围  $D(0$  到  $D)$  内的一辆坦克。

最多能消灭多少辆坦克？

### 【输入格式】

第一行：  $N, M, D(N, M \leq 1000)$

接下来  $N$  行：  $rest$

接下来  $M$  行：  $pos, speed$

全部都是 `longint` 内的正整数。

### 【输出格式】

输出最多能消灭坦克数量。

**【输入样例】**

```
3 3 3
3
2
1
4 1
1 1
2 1
```

**【输出样例】**

```
2
```

### 题 3 冰岛

**【问题描述】**

假设你在一个  $n*n$  的冰面上，并且你想到达这个冰面的某处，可是由于冰面太滑了，所以当你向某个方向出发后，你没有办法使自己停下来直到你碰到了某个障碍物——因为你可以抓住障碍物使得你的身体停止运动。

因为你已经知道了整个地图，所以你决定在行动之前先计算出最快可到达目标的路线，使得你可以不用走太多冤枉路，这时你决定编程解决这个问题……

**【输入格式】**

第一行包括一个正整数  $n(n \leq 1000)$

以下  $n$  行，每行包括  $n$  个数字（0 或 1），0 表示该点为空地可以滑行，1 表示该点为障碍物（障碍物无法穿过）。保证最外圈的地形为障碍物，也就是你无法离开这个地图。

接下来 1 行包括 2 个整数  $x, y$  ( $1 \leq x, y \leq n$ )，表示一开始你处于坐标  $(x, y)$

再接下来 1 行包括 2 个整数  $x_2, y_2$  ( $1 \leq x_2, y_2 \leq n$ )，表示你想要到达的目标为  $(x_2, y_2)$

**【输出格式】**

只有一个整数  $t$ ，表示能到达目标的最短时间（假设每经过一次滑行需要花费 1 单位的时间，无论这次滑行距离的长短）。所谓到达目标要求必须停留在  $(x_2, y_2)$ ，也就是你不能在到达之后被迫滑向下一个点。当你无法到达目标点时，你只须输出一行字符串 'impossible'。

**【输入样例 1】**

```
5
1 1 1 1 1
1 0 0 1 1
1 0 0 0 1
1 0 0 0 1
1 1 1 1 1
2 2
4 3
```

**【输出样例 1】**

```
2
```

说明：由  $(2, 2)$  到  $(2, 3)$ ，再由  $(2, 3)$  到  $(4, 3)$ ，2 次滑行到达终点。

**【输入样例 2】**

```
4
```

```
1 1 1 1
1 0 1 1
1 1 0 1
1 1 1 1
2 2
3 3
```

**【输出样例 2】**

```
impossible
```

**【数据范围】**

20%数据满足  $n \leq 5$   
40%数据满足  $n \leq 10$   
60%数据满足  $n \leq 200$   
100%数据满足  $n \leq 1000$

## 题 4 搭配购买

**【题目描述】**

Joe 觉得云朵很美, 决定去山上的商店买一些云朵。商店里有  $n$  朵云, 云朵被编号为  $1, 2, \dots, n$ , 并且每朵云都有一个价值。但是商店老板跟他说, 一些云朵要搭配来买才好, 所以买一朵云则与这朵云有搭配的云都要买。

但是 Joe 的钱有限, 所以他希望尽量买的价值越多越好。

**【输入描述】**

第 1 行  $n, m, w$ , 表示  $n$  朵云,  $m$  个搭配, Joe 有  $w$  的钱  
第  $2 \sim n+1$  行, 每行  $c_i, d_i$  表示  $i$  朵云的价钱和价值  
第  $n+2 \sim n+1+m$  行, 每行  $u_i, v_i$ , 表示买  $u_i$  就必须买  $v_i$ , 同理, 如果买  $v_i$  就必须买  $u_i$ 。

**【输出描述】**

一行, 表示可以获得的最大价值

**【输入样例】**

```
5 3 10
3 10
3 10
3 10
5 100
10 1
1 3
3 2
4 2
```

**【输出样例】**

```
1
```

**【数据范围】**

30%的数据保证:  $n \leq 100$   
50%的数据保证:  $n \leq 1,000; m \leq 100; w \leq 1,000$   
100%的数据保证:  $n \leq 10,000; 0 \leq m \leq 5000; w \leq 10,000$