

# 吉哥系列故事——礼尚往来

---

Time Limit: 0.2 Seconds    Memory Limit: 65536K

---

吉哥还是那个吉哥  
那个江湖人称“吼吼哥”的基哥

每当节日来临，女友众多的吼吼哥总是能从全国各地的女友那里收到各种礼物。  
有礼物收到当然值得高兴，但回礼确是件麻烦的事！  
无论多麻烦，总不好意思收礼而不回礼，那也不是吼吼哥的风格。

现在，即爱面子又抠门的吼吼哥想出了一个绝妙的好办法：他准备将各个女友送来的礼物合理分配，再回送不同女友，这样就不用再花钱买礼物了！

假设吼吼哥的  $n$  个女友每人送他一个礼物（每个人送的礼物都不相同），现在他需要合理安排，再回送每个女友一份礼物，重点是，回送的礼物不能是这个女友之前送他的那个礼物，不然，吼吼哥可就摊上事了，摊上大事了.....

现在，吼吼哥想知道总共有多少种满足条件的回送礼物方案呢？

## Input

输入数据第一行是个正整数  $T$ ，表示总共有  $T$  组测试数据（ $T \leq 100$ ）；  
每组数据包含一个正整数  $n$ ，表示吼吼哥的女友个数为  $n$ （ $1 \leq n \leq 100$ ）。

## Output

请输出可能的方案数，因为方案数可能比较大，请将结果对  $10^9 + 7$  取模后再输出。  
每组输出占一行。

## Sample Input

```
3
1
2
4
```

## Sample Output

```
0
1
9
```

# 1002 XCOM-Enemy Unknown

Time Limit: 0.2 Seconds    Memory Limit: 65536K

XCOM-Enemy Unknown 是一款很好玩很经典的策略游戏。  
在游戏中,由于未知的敌人--外星人入侵,你团结了世界各大国家进行抵抗。



随着游戏进展,会有很多的外星人进攻事件.每次进攻外星人会选择 3 个国家攻击,作为联盟的指挥者,你要安排有限的联盟军去支援其中一个国家,抵抗进攻这个国家的外星人.



战斗胜利之后这个被支援的国家恐慌值就会-2 点,而其他两个未被支援的国家恐慌值就会+2 点,同时和这两个国家在相同大洲的其他国家恐慌值也会+1 点.当一个国家的恐慌值超过 5 点,这个国家就会对联盟失去信心从而退出联盟.

现在给你外星人将会进攻的地点,问你最多能在不失去任何一个国家信任的情况下抵挡多少次外星人的进攻.

## Input

第一行有一个整数  $\tau$  代表接下来有  $\tau$  组数据;  
每组数据第一行是三个整数  $n, m, k$  分别代表联盟国家的个数,大洲的个数,外星人的进攻次数;  
第二行是  $n$  个数字代表各个国家所属的大洲(大洲序号从 0 到  $m-1$ );  
第三行是  $n$  个数字代表各个国家初始的恐慌值;  
接下去  $k$  行代表外星人进攻;  
每行有三个数字,表示该次外星人进攻的国家(国家序号从 0 到  $n-1$ ).

### [Technical Specification]

$0 < T \leq 100$

$8 < n \leq 16$

$2 < m \leq 5$

$0 < k \leq 100$

$0 < \text{初始恐慌值} \leq 5$

每个州至少有三个国家

每次外星人进攻一定发生在不同州的三个国家

### Output

首先输出 case 数 (见 sample), 接着输出在不失去任何一个国家的情况下能抵挡外星人进攻最多的次数.

### Sample Input

```
1
9 3 2
0 0 0 1 1 1 2 2 2
3 3 3 3 3 3 3 3 3
0 3 6
0 3 6
```

### Sample Output

Case #1: 1

### Hint

第一次如果选择支援国家 0, 那么国家 3 和 6 的恐慌值就会增加到 5, 第二次不管怎么选, 国家 3 和 6 总会有一个超过 5.

## 1003 威威猫系列故事——数字分配

Time Limit: 5.0 Seconds    Memory Limit: 65536K

住到直角坐标系的威威猫并没有躲开烦恼，难得去晒一次被子，竟然也被洪水冲走了...  
思考再三，威威猫决定报名参军，准备到革命的大熔炉去锻炼自己，于是不久，威威猫就成为了一名光荣的革命军人，并被委派到某边防小岛负责信号站维护工作。

在这个小岛上有 36 所信号站，呈矩阵排列，我们把左上角的信号站坐标标注为  $(0, 0)$ ，右下角的信号站坐标标注为  $(5, 5)$ 。

每个信号站可以分配 1~5 个服务频率，来服务周围的用户群。

可以分配的频率为 1~30，由于技术原因，相同或者相邻的频率存在干扰，干扰因素如下：

1. 同一个信号站中相同的两个频率会造成无穷大的干扰
2. 同一个信号站中相邻的两个频率会造成 A 点干扰值
3. 曼哈顿距离为 1 的两个信号站中如果有两个相同的频率，会造成 B 点干扰值
4. 曼哈顿距离为 1 的两个信号站中如果有两个相邻的频率，会造成 C 点干扰值
5. 曼哈顿距离为 2 的两个信号站中如果有两个相同的频率，会造成 D 点干扰值
6. 曼哈顿距离为 2 的两个信号站中如果有两个相邻的频率，会造成 E 点干扰值

一对频率产生的干扰只算一次。

现在给你各个信号站的初始服务频率分配情况，让你改变  $(2, 2)$   $(2, 3)$   $(3, 2)$   $(3, 3)$  四个信号站中的服务频率（不能改变频率个数），使得总体的干扰值尽量的小。

### Input

第一行有一个数字 T 代表有 T 组数据；

每组数据的第一行有 5 个数字 A B C D E 代表上述题目中的干扰值；

接下去有  $6 \times 6$  的数据块描述各个信号站的初始频率分配情况；

每个数据块为  $[n: a_1, a_2 \dots a_n]$  的描述形式，n 为该站的服务频率个数， $a_i$  表示初始频率分配情况（见 sample）；

### [Technical Specification]

所有数据都是整数

$1 \leq T \leq 50$

$1 \leq A, B, C, D, E \leq 100$

$1 \leq n \leq 5$

$1 \leq a_i \leq 30$

数据保证每个信号站初始分配情况不会出现相同的频率

### Output

首先输出 case 数（见 sample），接着输出改变服务频率之后总体最小的干扰值。

### Sample Input

```
1
5 4 3 2 1
[1:1] [1:2] [1:3] [1:4] [1:5] [1:6]
[1:1] [1:2] [1:3] [1:4] [1:5] [1:6]
[1:1] [1:2] [5:1, 2, 3, 4, 5] [5:1, 2, 3, 4, 5] [1:5] [1:6]
[1:1] [1:2] [5:1, 2, 3, 4, 5] [5:1, 2, 3, 4, 5] [1:5] [1:6]
[1:1] [1:2] [1:3] [1:4] [1:5] [1:6]
[1:1] [1:2] [1:3] [1:4] [1:5] [1:6]
```

### Sample Output

Case #1: 363

# 1004 VIM-adventures

Time Limit: 5.0 Seconds Memory Limit: 65536K

<http://vim-adventures.com/>

vim 是一个非常方便的文本编辑器,使用它可以快速的编辑文本

vim-adventures 是一个很有意思的解谜游戏,需要玩家使用一些 vim 下的指令来移动光标来解决一系列谜题

为了化简题意此题我们只考虑如下指令:

h: 让光标向左边移动一格

j: 让光标向下边移动一格

k: 让光标向上边移动一格

l: 让光标向右边移动一格

w: 让光标移动到下一个"word 的开头"

e: 让光标移动到下一个"word 的结尾"

b: 让光标移动到前一个"word 的开头"

ge: 让光标移动到前一个"word 的结尾"

W: 让光标移动到下一个"WORD 的开头"

E: 让光标移动到下一个"WORD 的结尾"

B: 让光标移动到前一个"WORD 的开头"

gE: 让光标移动到前一个"WORD 的结尾"

(注意断句,是"下一个|word的结尾",而不是"下一个 word| 的结尾".如"1234 5678",2 的下一个 word 的结尾是 4 而不是 8,其他同理)

注:

其中一个 word 由字母/数字/下划线序列或者其他的非空字符序列组成.由 white space (空格或者换行符) 分开;

其中一个 WORD 由非空字符序列组成.由 white space (空格或者换行符) 分开;

下一个/前一个 word/WORD 可以跨行,如当前 word 是该行最后一个 word,那么下一个 word 就是下一行的第一个 word;

如果没有下一个/前一个 word/WORD,那么按下 w/e/b/ge/W/B/E/gE 之后会让光标移动到所能移动的最远处;

例子:

This is a line with example text

<----<--<--<----<----<-----<---

b b b b b b b

This is a line with example text

<--<--<--<----<----<-----<-----

ge ge ge ge ge ge ge

This is a line with example text

---->-->-->---->---->----->-->

w w w w w w w

This is a line with example text

--->-->-->---->----->----->-->

e e e e e e e

while (((true)))) printf("Hello VIM!")

----->-->-->---->----->----->-->

W W W W W W W

(w,e,b,ge,W 等表示按下该键后可以让当前光标移动到对应的"<",">"位子.

前两个例子光标一开始在最右,后三个例子光标一开始在最左)

文本中还有一些不能走上去的位置,如上图中的石头,这些石头光标不管怎么样都没办法移动上去.

现在给你一段文本,要求你从起点位置按最少的按键将光标移动到终点位置.  
其中 ge 算两次按键,大写字母如 W 可以按 shift+w(算两次按键),或者在 Caps Lock 打开的状态下按一下 w,打开/关闭 Caps Lock 算一次按键.  
shift 不能按住不放,如在 Caps Lock 大写的情况下,按下 shift+G 和 shift+E 来实现 ge 算4次按键.  
起始状态 Caps Lock 是在小写状态

## Input

第一行一个整数 T 代表接下去有 T 组数据  
每组数据第一行是六个整数 N,M,sx,sy,ex,ey. 其中 N,M 代表文本的尺寸,(sx,sy) 代表起点位置,(ex,ey) 代表终点位置,下标从 0 开始  
接下去 N 行,每行有 M 个字符,代表文本  
接下去 N 行,每行有 M 个字符,代表石头,如果是 0 则表示没有石头,如果是 1 则表示有石头  
**[Technical Specification]**

0<T<=50  
0<N,M<=250  
0<=sx,ex<N  
0<=sy,ey<M  
代表文本的 N\*M 矩阵所有字符的 ascii 码在 32 到 126 之间  
每行字符都不以空格开头/结尾  
代表石头的 N\*M 矩阵只会包含 0 或者 1  
起点和终点一定没有石头

## Output

首先输出 case 数(见 sample),接着输出最小的按键使得光标从起点移动到终点.如果走不到终点则输出-1.

## Sample Input

```
8
1 10 0 0 0 9
wwwwwwww w
0111111110
1 10 0 0 0 9
WWW.WWWW W
0111111110
1 10 0 0 0 9
e eeeeeeee
0111111110
1 10 0 0 0 9
E EEEE.EEE
0111111110
1 10 0 9 0 0
bbbbbbbbb b
0111111110
1 10 0 9 0 0
BBBBB.BB B
0111111110
1 10 0 9 0 0
. gegegege
0111111110
1 10 0 9 0 0
. gEgE..gE
0111111110
```

## Sample Output

```
Case #1: 1
Case #2: 2
Case #3: 1
Case #4: 2
Case #5: 1
Case #6: 2
Case #7: 2
Case #8: 3
```

## Hint

更多例子可以到 [vim-adventures.com](http://vim-adventures.com) 玩一玩前三关 (免费)

游戏一开始只能使用 `hjkl` 四个按键, 其他按键要到后边关卡吃到对应按键才能用

第一关过河的那个地方是一个难点, 要注意小人的提示

在游戏中输入 `:help` [指令] 可以查看指令使用方法以及例子

# 1005 郑厂长系列故事——排兵布阵

Time Limit: 5.0 Seconds    Memory Limit: 65536K

郑厂长不是正厂长  
也不是副厂长  
他根本就不是厂长  
事实上  
他是带兵打仗的团长

一天，郑厂长带着他的军队来到了一个  $n*m$  的平原准备布阵。

根据以往的战斗经验，每个士兵可以攻击到并且只能攻击到与之曼哈顿距离为 2 的位置以及士兵本身所在的位置。当然，一个士兵不能站在另外一个士兵所能攻击到的位置，同时因为地形的原因平原上也不是每一个位置都可以安排士兵。

现在，已知  $n, m$  以及平原阵地的具体地形，请你帮助郑厂长计算该阵地，最多能安排多少个士兵。

## Input

输入包含多组测试数据；

每组数据的第一行包含 2 个整数  $n$  和  $m$  ( $n \leq 100, m \leq 10$ )，之间用空格隔开；

接下来的  $n$  行，每行  $m$  个数，表示  $n*m$  的矩形阵地，其中 1 表示该位置可以安排士兵，0 表示该地形不允许安排士兵。

## Output

请为每组数据计算并输出最多能安排的士兵数量，每组数据输出一行。

## Sample Input

```
6 6
0 0 0 0 0 0
0 0 0 0 0 0
0 0 1 1 0 0
0 0 0 0 0 0
0 0 0 0 0 0
0 0 0 0 0 0
```

## Sample Output

```
2
```