

湖南师范大学第三届程序 **ACM** 大学生程序设计竞赛
The 3th Hunan Normal University Programming Contest

主办：湖南师范大学数学与计算机学院

2010 年 10 月 31 日

本次比赛 8 道题目，共 20 页。如有缺页，请立即通知赛场工作人员。

所有题目均采用标准输入输出，请不要读写任何文件。

所有题目的正确输出是惟一的。你的输出只有和正确输出完全一致时才能通过。

A. 原码转补码

【Limit】 Time Limit: 1 Seconds Memory Limit: 65536 KB

【Description】

原码(true form)是一种计算机中对数字的二进制定点表示方法。原码表示法在数值前面增加了一位符号位（即最高位为符号位）：正数该位为 0，负数该位为 1（0 有两种表示：+0 和 -0），其余位表示数值的大小。这样一个数的 8 位原码表示中，第一位是符号位，剩余七位是数值的二进制表示。例如：-8 的原码为 10001000，其中左边第一个是表示该数是一个负数，后面七位 0001000 表示该数的绝对值 8。

补码是机器中最常用的一直数字表示形式。把原码转换成补码有一条很简单的规则：如果原码符号位是 0，则补码与原码相同；否则，把原码除符号位外的其他各位取反加一得到补码。例如：

-8 的原码为 10001000，符号位为 1，所以把 0001000 取反得 1110111，再加 1 得 1111000，所以 -8 的补码为 11111000。

当然，原码取反加一变补码，有发生溢出的可能。**这里我们规定，溢出时，丢掉进位。**例如 10000000，把数值位 0000000 取反后得 1111111，加 1 的 10000000（1 是由进位产生），把 1 丢掉的 0000000，再加上前面的符号位 1，最终得 10000000。

【Input】

第一行为一个正整数 N，表示测试数据组数。
接下来是 N 行，每行表示一个 8 位二进制原码。

【Output】

输出每个原码对应的补码，每个补码占独立的一行。

【Sample Input】

```
3
10001000
00000000
10000000
```

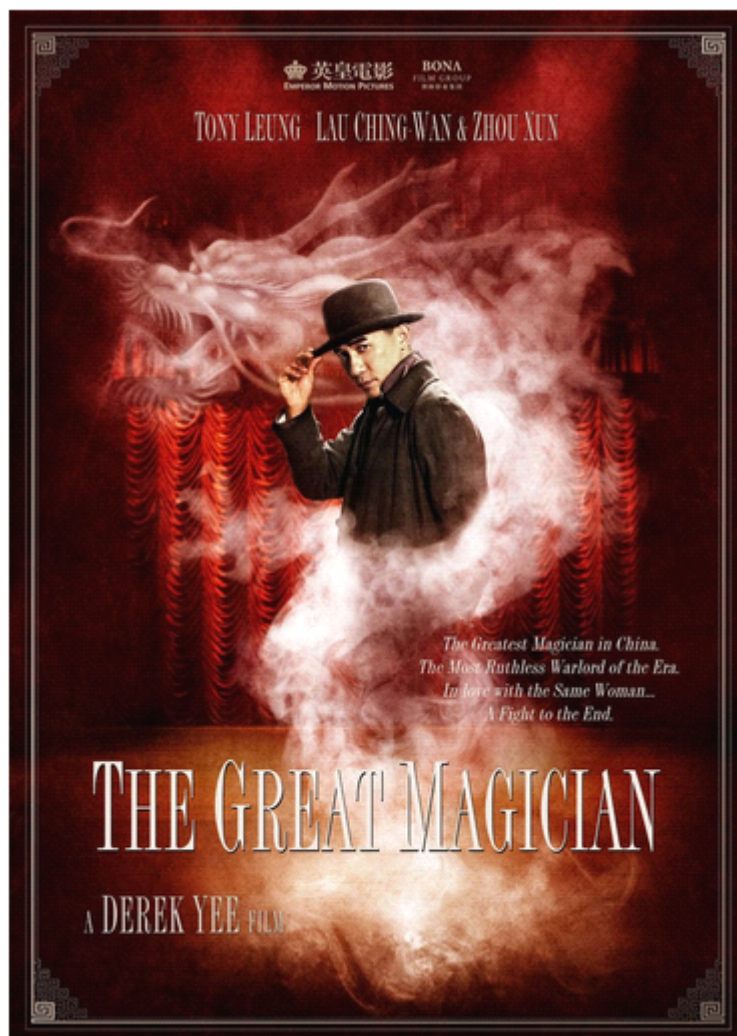
【Sample Output】

```
11111000
00000000
10000000
```

B. 大魔术师

【Limit】 Time Limit: 1 Seconds Memory Limit: 65536 KB

【Description】



LittleMing 刚刚看了《大魔术师》这部电影，他对魔术产生了浓厚的兴趣。于是，他决定向大魔术师 BigMing 学习魔术。

今天，BigMing 要教个 LittleMing 的魔术是这样的：道具是 13 张黑桃扑克 A,2,3,4,5,6,7,8,9,10,J,Q,K。这 13 张牌预先排好，正面朝下，拿在魔术师的手里，从上面开始，第一次数一张牌翻过来放在桌面上，正好是 A；第二次数两张牌，数 1 的那张放在手中扑克的下面，数 2 的那张翻过来放在桌面上正好是 2；……如此下去，放在桌面上的牌最后正好是 A,2,3,4,5,6,7,8,9,10,J,Q,K 的顺序（从下往上）。

LittleMing 立马发现这个魔术的关键就在预先必须安排好 13 张牌的顺序，所以他很快掌握的这个魔术。现在他想把这个魔术的难度加大：假设有 N 张牌，这 N 张牌依次标有 1~N

The 3th Hunan Normal University Programming Contest

这 N 个数。魔术师表演时，将挑选一位观众来决定 N 张牌被魔术师翻出的顺序（假设为 $x[1..n]$ ）。这时魔术师应该通过安排这 N 张牌的顺序，使得

(1) 第一次数一张牌翻过来放在桌面上，正好是 $x[1]$ ；

(2) 第二次数两张牌，数 1 的那张放在手中扑克下面，数 2 的那张翻过来放在桌面上正好是 $x[2]$;

.....

(3) 第 i 次数 i 张牌，数 1 的放在手中扑克下面，数 2 的放在手中扑克下面，.....，数 $i-1$ 的放在手中扑克下面，数 i 的那张翻过来放在桌面上正好是 $x[i]$;

.....

最终，魔术师亮出的牌的顺序将和 $x[1..n]$ 这个排列相同。

现在，关键的问题是，魔术师应怎样安排这 N 张牌的顺序，才能使亮他出的牌的顺序将和 $x[1..n]$ 这个排列相同？

【Input】

第一行一个数 CaseNumber，表示测试数据的组数。

对每组测试数据：

第一行一个整数 N ($N \leq 1000$)。

接下来一行表示有 N 个数，表示一个排列。

【Output】

对每个测试数据，输出魔术师应该安排的排列，每个两个数之间一个空格，最后一个数没有空格。

【Sample Input】

```
3
13
1 2 3 4 5 6 7 8 9 10 11 12 13
3
1 2 3
3
3 2 1
```

【Sample Output】

```
1 8 2 5 10 3 12 11 9 4 7 6 13
1 3 2
3 1 2
```

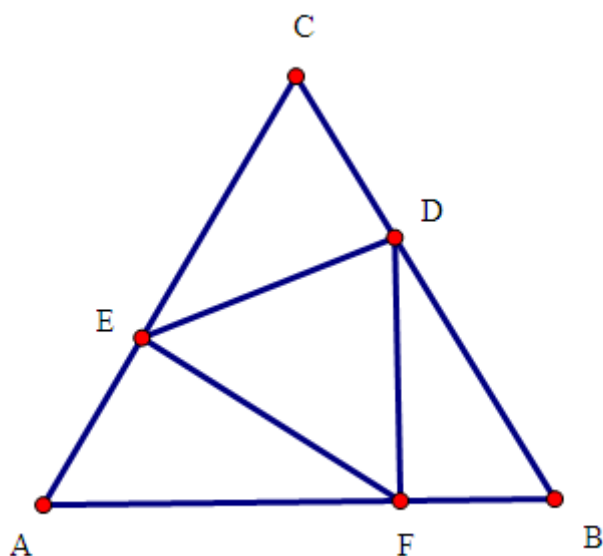
C. A Math Problem

【Limit】 Time Limit: 1 Seconds Memory Limit: 65536 KB

【Description】

LittleMing 今天碰到一道有趣的数学题，题目是这样的：在等边三角形 ABC 中，各边长度为 1。D, E, F 分别在边 BC, AC, AB 上，其中 $AF = \lambda_1 FB$ ， $BD = \lambda_2 DC$ ， $CE = \lambda_3 EA$ 。

若给出 λ_1 ， λ_2 ， λ_3 的值，请求出三角形 DEF 的面积。



【Input】

第一行有一个数字 CaseNumber，表示测试数据的组数。

每组测试数据有三个正数，分别为 λ_1 ， λ_2 ， λ_3 。

【Output】

每组测试输出一个保留 4 为小数的浮点数（四舍五入），表示三角形 DEF 的面积。

The 3th Hunan Normal University Programming Contest

【Sample Input】

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

【Sample Output】

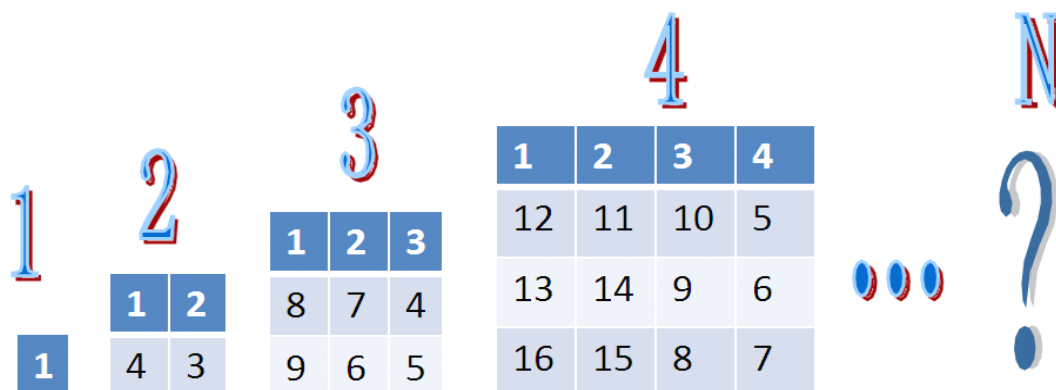
```
0.1083
0.1263
0.1443
```

D. Imagination

【Limit】 Time Limit: 1 Seconds Memory Limit: 65536 KB

【Description】

There are four matrix blow , Can you image the NEXT matrix from the given information?



【Input】

The first line is a integer which represents the case number .

For each test case , there will be a line which contains a integer N ($1 \leq N \leq 100$).

【Output】

Output the matrix which has N*N integer. Be careful that there is a blank space between two integer but no blank space after the last integer in a line.

【Sample Input】

4
1
2
3
4

【Sample Output】

```
1
1 2
4 3
1 2 3
8 7 4
9 6 5
1 2 3 4
12 11 10 5
13 14 9 6
16 15 8 7
```

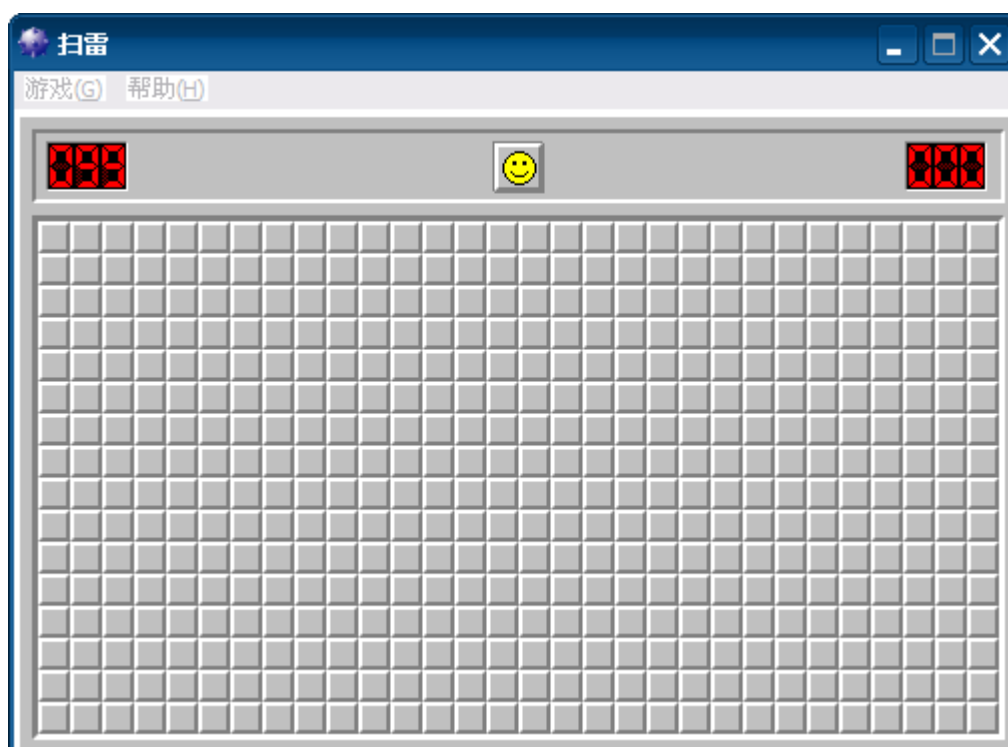

E. The Minesweeper Game

【Limit】Time Limit: 1 Seconds Memory Limit: 65536 KB

【Description】

注:以下说明来自百度百科: <http://baike.baidu.com/view/30088.htm#1>

扫雷简介



扫雷最原始的版本可以追溯到 1973 年一款名为“方块”的游戏。

不久之后，“方块”被改写成了游戏“Rlogic”。在“Rlogic”里，玩家的任务是作为美国海军陆战队队员，为指挥中心探出一条没有地雷的安全路线，如果路全被地雷堵死就算输。两年后，汤姆·安德森在“Rlogic”的基础上又编写出了游戏“地雷”，由此奠定了现代扫雷游戏的雏形。

1981 年，微软公司的罗伯特·杜尔和卡特·约翰逊两位工程师在 Windows 3.1 系统上加载了该游戏，扫雷游戏才正式在全世界推广开来。

这款游戏的玩法是在一个 9*9(初级)，16*16(中级)，16*30(高级)，或自定义大小的方块矩阵中随机布置一定量的地雷(初级为 10 个，中级为 40 个，高级为 99 个)。由玩家逐个翻开方块，以找出所有地雷为最终游戏目标。如果玩家翻开的方块有地雷，则游戏结束。

扫雷游戏的目标是尽快找到雷区中的所有不是地雷的方块，而不许踩到地雷。点开的数字是几，则说明该数字旁边的 8 个位置中有几个雷，如果挖开的是地雷，则会输掉游戏。

游戏玩法

启动:

在游戏菜单上, 单击开局。

要启动计时器, 请单击游戏区中的任何方块。

注意:

通过**单击**即可挖开方块。如果挖开的是地雷, 则输掉游戏。

如果方块上出现数字, 则表示在其周围的八个方块中共有多少颗地雷。

要标记认为可能有地雷的方块, 请**右键单击**它。 示例游戏区包括雷区、地雷计数器和计时器。

一个未点开的方格, 右键单击一次, 将放上旗子, 表示确定该方格是雷; 再右键单击一次, 将取下旗子并标记问号(?), 表示该方格不确定是否为雷; 第三次右键单击, 则取下问号, 恢复初始状态; 第四次右键单击, 则又放上旗子……依此类推。

自定义游戏区域:

在“游戏”菜单上, 单击“自定义(选项)”。

通过指定水平和垂直显示的方块数来定制雷区尺寸, 并指定雷区中放置的地雷数。

自定义游戏区域有个漏洞, 可使你在 10 秒之内必胜。

方法如下: 在自定义中输入高度、宽度随意大的数值(因为扫雷区域有限, 所以会固定在高度 24、宽度 30 的区域里), 雷数自定, 越少越容易, 随意点击一处, 然后按照普通规则继续即可。

策略技巧:

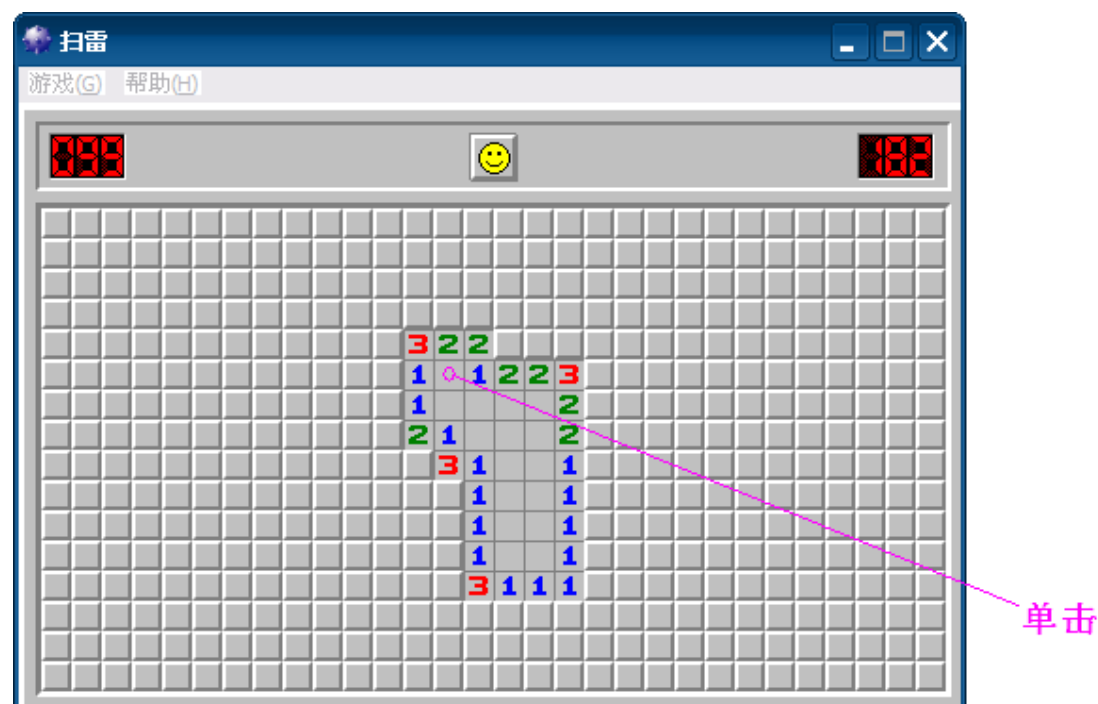
如果无法判定某方块是否有雷, 请用**右键单击**两次给它标记一个问号(?)。之后, 可以用鼠标右键单击方块一次将该方块标记为地雷或者用鼠标右键单击方块两次去掉标记。

如果编号方块周围地雷没有全部标记, 在同时使用**双击**后, 其他隐藏或未标记的方块将被按下一次(即闪烁一下)。

编程任务

现在是一个高度为 16, 宽度为 30 的雷区, 共计 99 个雷。题目将随机生成这样一个的带有地雷的地图, 当玩家首次单击某个方格时, 请你编写程序根据游戏规则对玩家的动作做出反应。

例如下图中, 玩家第一次点击了第 6 行第 14 行时, 游戏程序给出了如下反应。你应该输出这个地图, 该地图用一个二维字符数组表示, 其中未点开的方格用'#'表示; 已经点开但是八个方向都没有地雷的格子用'0'表示; 已经点开但八个方向至少有一个地雷的方格用该方格八个方向地雷数之和表示, 显然这个数字在 1-8 之间。



【Input】

第一行，有一个正整数 T ，表示测试数据的组数。

接下来有 T 组测试数据，每组测试数据之间用一个空行隔开。

对每组测试数据：

第一行有两个正整数 (x,y) ，表示玩家首次点击的方格位置为第 x 行第 y 列。输入数据保证玩家首次点击的方格一定不是地雷。

接下来是一张系统随机生成的带地雷的地图：方便起见，每张地图用一个 $16*30$ 的二维字符数组表示，其中字符 '*' 表示该方格放置了地雷，字符 '.' 表示该方格没有放置地雷。

【Output】

对每组测试数据，输出一个 $16*30$ 的二维字符数组。该二维数组就是【编程任务】中所描述的二维数组。

Be Careful: 各组输出数据之间应用一个空行隔开，最后一组结尾没有空行。

The 3th Hunan Normal University Programming Contest

【Sample Input】

2

3 24

```
*.*.....*.*.*.....*.  
...**.*.*.*.....**  
*.....**.*.*.....*  
..*.....*.....**.*.*.....  
...*...**.*.....**.*.....  
.*.*.....*.....*.....  
.....*.*.*.*.....**..  
.....*.*.....*.*.*..  
.....*.....*.....*  
...*.*.*.*.*.*.....*.*.*.  
...*.*.....*.....*.....  
.....*.....*.....*.*.*.  
...*.*.....*.....*.....  
..*.*.*.*.....*.....*.*.*.  
...*.....**.*.....*.....  
.....**.*.*.....
```

1 18

```
.....*.*.*.....  
*.*.*.....*.*.*.....  
.*.*.....**.*.*.....  
..*.*.....*.....*.....  
.*.....*.*.....*.....*.....  
.....**.*.*.....*.....*.....  
...*.....*.....**.*.....  
.....*.....*.....*.....*.....  
*.*.*.*.....*.....*.....*.....  
.*.....**.*.....*.....*.....*.....  
..*.....**.*.....  
...*.....*.....*.....*.....  
..*.....*.....*.....*.....  
.....*.....*.....*.....*.....  
..*.....*.....*.....**.*.....
```

【Sample Output】

```
#####100000001###  
#####111100001###  
#####210012####  
#####3101####  
#####102####  
#####102####  
#####114####  
#####  
#####  
#####  
#####  
#####  
#####  
#####  
#####  
#####
```

[illegible]

【Hint】

如果你对扫雷游戏规则不是很熟悉，你可以通过启动你自己电脑上的【扫雷游戏】熟悉一下规则。点击【开始】-【所有程序】-【游戏】-【扫雷】启动扫雷游戏。



F. Adventure of Super Mario

【Limit】 Time Limit: 2 Seconds Memory Limit: 65536 KB

【Description】

After rescuing the beautiful princess, Super Mario needs to find a way home -- with the princess of course :-). He's very familiar with the 'Super Mario World', so he doesn't need a map, he only needs the best route in order to save time.



There are A Villages and B Castles in the world. Villages are numbered $1..A$, and Castles are numbered $A+1..A+B$. Mario lives in Village 1, and the castle he starts from is numbered $A+B$. Also, there are two-way roads connecting them. Two places are connected by at most one road and a place never has a road connecting to itself. Mario has already measured the length of every road, but they don't want to walk all the time, since he walks one unit time for one unit distance (how slow!).

Luckily, in the Castle where he saved the princess, Mario found a magic boot. If he wears it, he can super-run from one place to another IN NO TIME. (Don't worry about the princess, Mario has found a way to take her with him when super-running, but he wouldn't tell you :-P)

Since there are traps in the Castles, Mario NEVER super-runs through a Castle. He always stops when there is a castle on the way. Also, he starts/stops super-runnings ONLY at Villages or Castles.

Unfortunately, the magic boot is too old, so he cannot use it to cover more than L kilometers at a time, and he cannot use more than K times in total. When he comes back home, he can have it repaired and make it usable again.

【Input】

The first line in the input contains a single integer T ($1 \leq T \leq 20$), indicating the number of test cases.

The 3th Hunan Normal University Programming Contest

Each test case begins with five integers A, B, M, L and K -- the number of Villages, the number of Castles($1 \leq A, B \leq 50$), the number of roads, the maximal distance that can be covered at a time($1 \leq L \leq 500$), and the number of times the boot can be used. ($0 \leq K \leq 10$) The next M lines each contains three integers Xi, Yi, Li. That means there is a road connecting place Xi and Yi. The distance is Li, so the walk time is also Li. ($1 \leq Li \leq 100$)

【Output】

For each test case in the input print a line containing a single integer indicating the minimal time needed to go home with the beautiful princess. It's guaranteed that Super Mario can always go home

【Sample Input】

```
1
4 2 6 9 1
4 6 1
5 6 10
4 5 5
3 5 4
2 3 4
1 2 3
```

【Sample Output】

```
9
```


G. Fill My Favourite Polygon

【Limit】 Time Limit: 2 Seconds Memory Limit: 65536 KB

【Description】

LittleMing love polygons very much . His favourite polygon has follow properties :

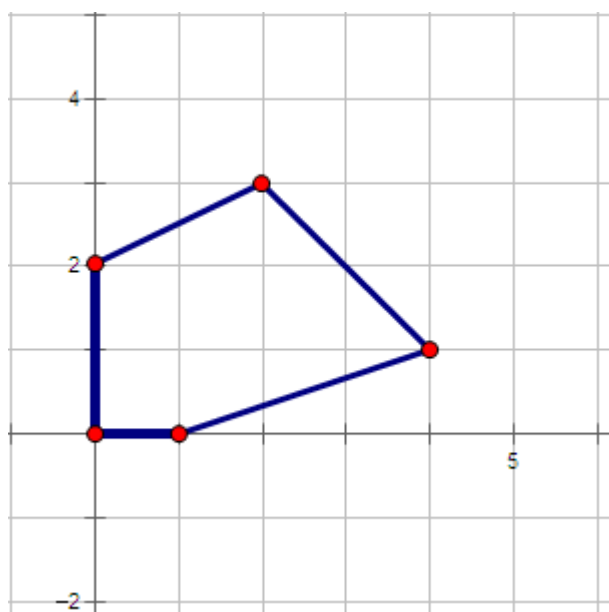
A) Each point of the Polygon (marked as Two-Dimensional Coordinate Point(x,y)) is integral point which means both x and y is integer ;

B) Never has two edges of the polygon cross each other ;

Today , What LittleMing interests is the integral point **inside** his favourite polygon . He want to calculate :

$$Sum = \sum_{(x,y)} (x \text{ XOR } y)$$

((x,y) is a integral point **inside** the favourite polygon , a integral point which is exactly at a edge of the polygon is not considered as inside the favourite polygon).



【Input】

The first line contains the number of scenarios.

For each scenario, The first line of the input contains a single integer n ($3 \leq n \leq 1000$), the number of vertexes in the polygon.

In the following n lines, each contain a pair of integral numbers, which describes the coordinates of the vertexes (xi, yi) ($0 \leq x_i, y_i \leq 1000$). The polygon in each test case starts from the first vertex to the second one, then from the second to the third, ...and so on. At last, it closes from

The 3th Hunan Normal University Programming Contest

the nth vertex to the first one.

【Output】

As shown below, the output of each scenarios should be the Sum MOD 1000000009 in a separate line .

【Sample Input】

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

【Sample Output】

```
8
```

【Hint】

Explanation For Sample Input:

All integral points inside polygon are : (1,1),(1,2),(1,3),(2,1),(2,2).

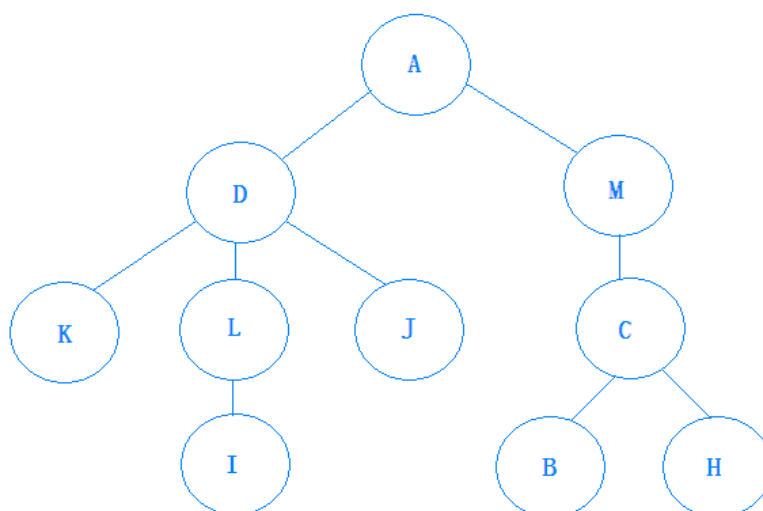
So the Sum = ((1 XOR 1) + (1 XOR 2) + (1 XOR 3) + (2 XOR 1) + (2 XOR 2)) MOD 1000000009 = 8 .

H. Counting Letter Tree

【Limit】 Time Limit: 1 Seconds Memory Limit: 65536 KB

【Description】

Letter Tree is a kind of tree whose node is a letter from A to Z. For example , the carton below shows a Letter Tree , node A is the root of tree and it has two children node D and node M (we can also say that parent of both node D and node M is node A).



What makes it special is that each node of Letter Tree has its Level Value which is between 1 and 5 . The relationship between a letter and its Level Value are showed in following table.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1

Now we raise the problem : Given a Letter Tree , can you count the number of subtrees so that the sum of Level Value of each node in the subtree exactly equals to 25 ?

【Input】

The first line in the input contains a single integer $T(1 \leq T \leq 10)$, indicating the number of test cases.

Each test case begins with a integer N , means the number of node who has one child or more children in the Letter Tree . Then , there will be N strings and each string is in a separate line. The string $s[1...Length]$ means that all nodes $s[2], s[3]...s[Length]$ are children of node $s[1]$.

The input will guarantee total number of node in a tree is less or equal than 26 and every node is a letter between A and Z. **We assume that the root of Letter Tree is always node A ,but the subtree need not meet the assumption.**

The 3th Hunan Normal University Programming Contest

【Output】

For each test , output the answer after the answer MOD 30011 in a separate line.

【Sample Input】

```
1
5
ADM
DKLJ
MC
LI
CBH
```

【Sample Output】

```
2
```

【Hint】

A subtree of a tree T is a tree consisting of a node in T and all of its descendants in T.

The subtree corresponding to the root node is the entire tree; the subtree corresponding to any other node is called a proper subtree (in analogy to the term proper subset).

(source from: http://en.wikipedia.org/wiki/Subtree#cite_note-1)