code jam

System.out.println("hello, world!");

Time Remaining: 21 hours 19 min

Contest scoreboard | Sign in

Qualification Round 2014

A. Magic Trick

B. Cookie Clicker Alpha

C. Minesweeper Master

D. Deceitful War

Questions asked

- Submissions

Magic Trick

6pt	Not attempted **7702/8614 users** correct (89%)

Cookie Clicker Alpha

8pt	Not attempted **4791/5150 users** correct (93%)
11pt	Not attempted **4517 users** attempted

Minesweeper Master

11pt	Not attempted **377/1136 users** correct (33%)
24pt	Not attempted **286 users** attempted

Deceitful War

14pt	Not attempted **920/1087 users** correct

Problem C. Minesweeper Master

You are not eligible to compete in this contest.

Small input 11 points	Only contestants can download the input.
Large input 24 points	Only contestants can download the input.

Problem

Minesweeper is a computer game that became popular in the 1980s, and is still included in some versions of the *Microsoft Windows* operating system. This problem has a similar idea, but it does not assume you have played *Minesweeper*.

In this problem, you are playing a game on a grid of identical cells. The content of each cell is initially hidden. There are **M** mines hidden in **M** different cells of the grid. No other cells contain mines. You may click on any cell to reveal it. If the revealed cell contains a mine, then the game is over, and you lose. Otherwise, the revealed cell will contain a digit between 0 and 8, inclusive, which corresponds to the number of neighboring cells that contain mines. Two cells are neighbors if they share a corner or an edge. Additionally, if the revealed cell contains a 0, then all of the neighbors of the revealed cell are automatically revealed as well, recursively. When all the cells that don't contain mines have been revealed, the game ends, and you win.

For example, an initial configuration of the board may look like this ('*' denotes a mine, and 'c' is the first clicked cell):

```
*..*...**.
....*.....
..c..*....
........*.
..........
```

	(85%)
16pt	Not attempted **893 users** attempted

- Top Scores

Gennady.Korotkevich	90
surwdkgo	90
Eryx	90
DoublePointer	90
Marcinsmu	90
SnapDragon	90
drazil	90
sevenkplus	90
Krazul	90
Al.Cash	90

There are no mines adjacent to the clicked cell, so when it is revealed, it becomes a 0, and its 8 adjacent cells are revealed as well. This process continues, resulting in the following board:

```
*..*...**.
1112*.....
00012*....
00001111*.
00000001..
```

At this point, there are still un-revealed cells that do not contain mines (denoted by '.' characters), so the player has to click again in order to continue the game.

You want to win the game as quickly as possible. There is nothing quicker than winning in one click. Given the size of the board (**R** x **C**) and the number of hidden mines **M**, is it possible (however unlikely) to win in one click? You may choose where you click. If it is possible, then print any valid mine configuration and the coordinates of your click, following the specifications in the *Output* section. Otherwise, print "Impossible".

Input

The first line of the input gives the number of test cases, **T**. **T** lines follow. Each line contains three space-separated integers: **R**, **C**, and **M**.

Output

For each test case, output a line containing "Case #x:", where x is the test case number (starting from 1). On the following **R** lines, output the board configuration with **C** characters per line, using '.' to represent an empty cell, '*' to represent a cell that contains a mine, and 'c' to represent the clicked cell.

If there is no possible configuration, then instead of the grid, output a line with "Impossible" instead. If there are multiple possible configurations, output any one of them.

Limits

0 ≤ **M** < **R** * **C**.

Small dataset

1 ≤ **T** ≤ 230.
1 ≤ **R**, **C** ≤ 5.

Large dataset

1 ≤ **T** ≤ 140.
1 ≤ **R**, **C** ≤ 50.

Sample

Input

```
5
5 5 23
3 1 1
2 2 1
4 7 3
10 10 82
```

Output

```
Case #1:
Impossible
Case #2:
c
.
*
Case #3:
Impossible
Case #4:
......*
.c....*
.......
..*....
Case #5:
**********
**********
**********
****....**
***.....**
***.c...**
***....***
**********
**********
**********
```