

11142 MineSweeper II

*“Minesweeper is more than a game,
it’s a way of life!”*

Recently, Mostafa has learned to play Minesweeper. He likes playing the game so much, but he cannot detect the mines of some states of the game. Thus, he decided to write a program to do the task for him. But he couldn’t and he asks you to write the program!

Here is an explanation of a game state:

- The game has an $M \times N$ board.
- Some cells are not marked, and some are marked.
- Unmarked cells are identified by a ‘.’ (without single-quotes) character.
- If a cell is marked with ‘X’, it means that there is a mine in that cell.
- If a cell is marked with ‘E’, it means that there is no mine in that cell, and in the cells adjacent to it (every cell has 8 adjacent cells).
- If a cell is marked with a digit $D = 1..8$, it means that there is no mine in that cell, but there are exactly D adjacent cells which contain mines.

Given a *valid* state of the game, Your task is to determine the unmarked cells that certainly contain a mine.

Note: There are no more than 35 unmarked cells.

Input

The first line of input gives the number of cases T . Then, T test cases follow. Each one starts with a line containing number of rows ($1 \leq M \leq 10$) and number of columns ($1 \leq N \leq 10$) and the number of unmarked cells with bombs ($c \leq 15$). Each of next M lines contain exactly N characters. These lines demonstrate a state of the game. There will be a blank line after each test case.

Output

For the x -th test case, your program must output the line containing ‘Case # x :’, followed by M lines each containing N characters, which demonstrate the same state of the game, with all unmarked cells certainly containing a mine, changed to ‘X’.

Sample Input

```
4
2 2 2
22
..
3 3 0
```

```
121
X.X
..1
```

```
3 3 0
...
...
X1.
```

```
3 4 1
.2X.
121.
EEEE
```

Sample Output

```
Case #1:
22
XX
```

```
Case #2:
121
X.X
..1
```

```
Case #3:
...
...
X1.
```

```
Case #4:
X2X.
121.
EEEE
```