

Spring 2019 COP 3502H Program #1: Minesweeper Setup (minesweeper.c)

The programming assignment is based upon the game Minesweeper. For the purposes of this assignment, the game board is an 8 x 8 grid of squares, with some subset of the squares containing bombs. In the game, the player can reveal a single square. If the square is a bomb, a bomb is shown (and the game ends with the player losing). If the square is NOT a bomb, then a number is shown, representing the number of adjacent squares (to the revealed square) that contain bombs. In implementing this game, it's useful to be able to pre-compute what would be revealed for each square, if the player clicked it.

The Problem

Thus, for this assignment, you will write a program that will, given the locations of each bomb on a minesweeper grid, calculate and print out a fully revealed board (stars representing bombs and numbers representing the number of adjacent squares with bombs).

Note: For this assignment and all other assignments, you will read in input from standard input without prompting the user to enter anything. All output will be to standard output. If there is an exception to these rules, it will be clearly stated in the assignment.

Input Format

The first line of the input file contains a single positive integer, n , representing the number of minesweeper boards to construct.

The first line of each board contains a single positive integer, m , representing the number of bombs for that board. The following m lines will contain two non-negative integers in between 0 and 7, inclusive, representing the row and column, respectively, of the location of a bomb. No two of these m lines will be identical.

Output Format

For each board, output to the screen, the following header line:

Board #k:

where k is the number of the board, starting with 1.

Follow this with the board printed out over 8 lines, with a space in between each entry.

Output a blank line after the output for each input case.

Sample Input

```
2
3
0 0
2 1
7 7
10
0 0
0 1
0 2
0 3
0 4
0 5
0 6
0 7
7 0
7 1
```

Corresponding Sample Output

Board #1:

```
* 1 0 0 0 0 0 0
2 2 1 0 0 0 0 0
1 * 1 0 0 0 0 0
1 1 1 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 1 1
0 0 0 0 0 0 1 *
```

Board #2:

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