8. Escape!

Program Name: Escape.java   Input File: escape.dat

You are out in the woods, camping alone when you wake up to a loud roar. A hungry bear has come up onto your campsite and thinks you smell delicious! The woods are pretty thick and you want to know what chance you have of escaping alive. In order to do this you whip out your laptop and write a program that can take in various configurations of obstacles, the bear, and yourself and will compute your probability of escape. Your program will compute every permutation of moves you can make, a simulation of the bear's logic, and then calculate how many solutions result in your escape and how many result in your demise. You obviously program pretty quickly, as that bear sure is hungry!

You have designed your program so the woods are represented by rectangular grid composed of the following characters:

- # – an Obstacle, you cannot move into this space
- E – Escape space, you live!
- B – Bear!
- Y – You
- . – empty space, you can move here

Your program will work as follows:

- You and the bear will alternate moving. You will move first and every time you make a move, the bear will make one right after unless the chase is over.
- Both you and the bear can move in any direction (horizontally, vertically or diagonally), but you cannot go back to a square that you've been to before. The bear has no such restriction.
- The bear will always move toward you, as you smell like dinner.
- If your move is to the same row or column as the bear, he will move one square towards you, in that row or column.
- If your move is not to the same row or column as the bear, the bear will move diagonally toward you.
- The bear has no problem moving into the Escape space if that means it gets closer to its dinner.
- If your move is to a spot that is adjacent to the bear, then the bear's next move will be to move into your square to eat you. Yum!
- If the move the bear wants to make is into an obstacle, he will stay put and look frustrated.
- If you cannot move because all adjacent spaces have been visited or contain obstacles, then the chase is over and is considered neither an escape for you nor a dinner for the bear.
- Every grid will result in at least one successful escape for you or dinner for the bear.

Input

The first line will contain a single integer n that indicates the number of escape grids to follow. For each escape grid:

- the first line will contain 2 integers in the form r c, where r is the number of rows in the grid, c is the number of columns, and both r and c are greater than zero and less or equal to five.
- The next r lines will contain c characters as described above with no spaces.
- **Note:** Each grid is guaranteed to have only one Bear, only one You, and at least one Escape.

Output

For each grid you will output the truncated percent chance of survival (integer based), which is the percentage of the number of the solutions that ended in your escape divided by the number of solutions that were either your escape or you being dinner. Each percent should be followed by the % character and each grid's output should be on its own line.

(continued on next page)

8. Escape! (cont.)

Example Input File

```
2
5 5
##...
..Y..
.....
##...
BE...
4 4
...Y
..#.
..B.
...E
```

Example Output to Screen

```
7%
0%
```