7220 Dungeon Trap

Wijaya is a (self-proclaimed) hardcore gamer and also a talented programmer. His new game, Dungeon Trap (DT), is scheduled to be published by next month. DT is a puzzle-like game and is built for both iOS and Android platform. In each puzzle, player will be given a grid map of N rows and M columns. Each cell in the map is either:

- starting point, represented by character '`A`'
- goal point, represented by character '`B`'
- obstacle, represented by character '`0`' (zero)
- empty cell, represented by character '`1`' .. '`9`'

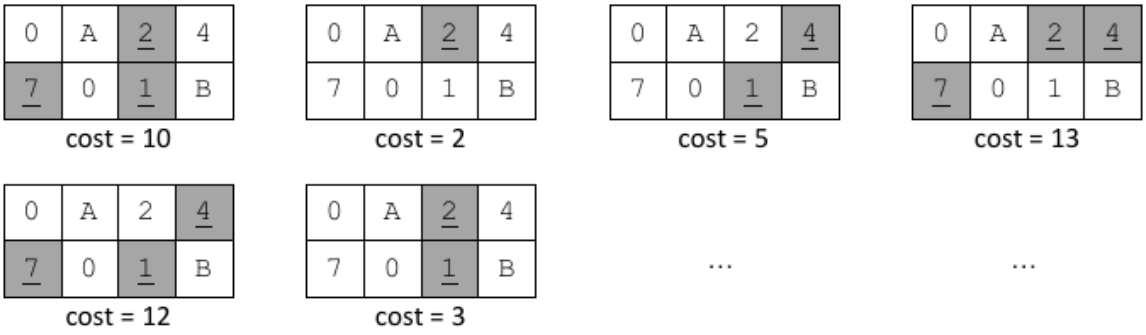
The game objective is to prevent an imp (the game's character) to reach goal point from the starting point. From each cell, the imp can move to an adjacent cell (sharing an edge), given the destination cell is not an obstacle. To prevent the imp reaching the goal point, the player can put additional obstacles on empty cells, i.e. by transforming an empty cell into a cell with obstacle. The cost of transforming an empty cell into an obstacle is represented by the empty cell's character; '`1`' means the cost is 1 token, '`2`' means 2 tokens, ..., '`9`' means 9 tokens. The player can continuously transform empty cells, given the puzzle is not yet solved. A puzzle is not considered as solved if and only if there exists at least one way for the imp to reach the goal point (oh, by the way, the imp is not actually moving in the game, it's supposed to be a puzzle).

Wijaya believes that high is the new low, and it is reflected in this game. The more tokens you spent in each puzzle, the lower your rank is (consequently, the worse player you are). For each puzzle, Wijaya wants to know the worse number of tokens that can be spent by a player.

For example, consider the following 2×4 map.

0	A	2	4
7	0	1	B

In this example, the puzzle can be solved by at most 13 tokens. These are several possible games that might be played by the player:



The darken/underlined cells are those which are transformed.

Note that the previous example plays are not exhaustive, there are other possible plays. The player cannot transform all empty cells into obstacle with 14 tokens without winning the game first (the best one can do is 13).

Before the game is published, Wijaya has to do a quality check on the game. He needs to ensure all puzzles are correct and challenging. In particular, he needs your help to determine the maximum number of tokens can be spent by a player in each puzzle.

Input

The first line of input contains T ($T \leq 100$) denoting the number of cases. Each case begins with two integers N ($2 \leq N \leq 100$) and M ($2 \leq M \leq 100$) denoting the size of the map (rows and columns, respectively). In the following N lines, each contains M characters describing the given map. Each character in the map is either: '`A`', '`B`', '`0`', '`1`', '`2`', ..., '`9`' which meaning has been defined in the above problem statement. It is guaranteed that '`A`' and '`B`' each appear exactly once and never adjacent in the given map.

Output

For each case, output '`Case #`X`:` Y' (without quotes) in a line where X is the case number (starts from 1) and Y is an integer representing the maximum number of tokens can be spent by a player for the respective case.

Note:

- Explanation for 1st sample case

  This is the example from the problem statement

- Explanation for 2nd sample case

  There's no need to transform any empty cell as there's already no way to reach '`B`' from '`A`'.

- Explanation for 3rd sample case

  In this puzzle, player needs to transforms all empty cells to win the game.

Sample Input

```
4
2 4
0A24
701B
2 3
A04
90B
2 2
3A
B5
3 3
37A
496
B52
```

Sample Output

```
Case #1: 13
Case #2: 0
Case #3: 8
Case #4: 29
```