

## 11905 Map Coloring

Given a model of a map in a 2D grid, you have to color the map satisfying certain constraints.

- 1) A map contains one or more regions. Each region is identified by an uppercase English letter  $L_i$ . In the grid, there will be some cells containing  $L_i$  to form the region. From any cell of that region, it's possible to go to all cells (in that region) using adjacent moves. Adjacent move means going to a cell which shares a side with the current cell and belongs to the same region.
- 2) A region **Q** may be surrounded by another region **P**. In such case, **Q** is called a sub-region of **P** and **Q** may be erased from the map. Mark all the cells that are surrounded by **P** with 'P's identifier letter.
- 3) All regions that are not surrounded by other regions should be colored, but two adjacent regions shouldn't be colored using the same color.
- 4) Two regions **A** and **B** are said to be adjacent if there is a cell of region **A**, and a cell of region **B**, share a side.
- 5) If two cells of a region share a corner, then there will always be at least one cell which shares sides with both the cells.

Now your job is to find the minimum number of different colors you need to color the map.

### Input

The first line of input will contain  $T$  ( $\leq 100$ ) denoting the number of cases.

Each case starts with two integers  $m$  and  $n$  ( $5 \leq m, n \leq 50$ ) denoting the number of rows and columns respectively. Each of the next  $m$  lines contains  $n$  characters each. Each character will be either a '.' or any uppercase English letter. '.' means empty place, and letters represent regions as described above. You can assume that the input data satisfies the above constraints.

### Output

For each case, print the case number and the minimum number of colors. Print  $m$  lines, each line with  $n$  characters showing the final grid after erasing sub-regions and filling the surrounding cells.

### Sample Input

```
2
5 5
AAAEE
ABAHE
A.AHF
AAAHF
.G...
20 20
.....
...B.....
..BBBB.....
```

```

..B..BBB.....
.BB....B.....
BBB....BBBBBBBBB...
BBB.....SS.BBB...
.BB.....SSBB.....
..B.....D..BBB.....
.BB....DDD.B.....
..BB.....B..BBB...
...B.....BBBBBB...
..BB.....BBBCBB...
...B.KK..BB.BCB.II.
...B.K..BBB..BCBBII.
...BBBBBB...BBBBMII
...BBBBBB...BBB...
.....BB...
.....BB..
.....B..
    
```

**Sample Output**

Case 1: 3

```

AAAEE
AAAHE
AAAHF
AAAHF
    
```

.G...

Case 2: 3

```

.....
...B.....
..BBBB.....
..BBBBBB.....
.BBBBBBB.....
BBBBBBBBBBBBBBBBB...
BBBBBBBBBBBBBBBBB...
.BBBBBBBBBBBBBBB.....
..BBBBBBBBBBBBBB.....
.BBBBBBBBBBBB.....
..BBBBBBBBBBB..BBB...
...BBBBBBBBBBBBBBB...
..BBBBBBBBBBBBBBBBB...
...BBBBBBBBBB.BBB.II.
...BBBBBBBBB..BBBBII.
...BBBBBB...BBBBMII
...BBBBBB...BBB...
.....BB...
.....BB..
.....B..
    
```