1115 – Filling the Regions

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) 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 report the updated map after filling the regions.

Input

Input starts with an integer **T (≤ 100)**, denoting the number of test cases.

Each case starts with two integers **m** and **n (5 ≤ m, n ≤ 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 first. Print **m** lines, each line with **n** characters showing the final grid after erasing sub-regions and filling the surrounding cells.

Sample Input	Output for Sample Input
2 5 5 AAAEE ABAHE A.AHF AAAHF .G... 20 20 ....B................ ..BBBB.............. ..B..BBB............ .BB....B............ BBB..RRBBBBBBBBBB... BBB..RR....SS.BBB... .BB..RR....SSBB..... ..B..RR.DD.BBB...... .BB..KMDDD.B........ ..BB.KDD...B..BBB... ...B.KK....BBBBBB... ..BB.KK....BBBCBB... ...B.KK...BB.BCB.... ...B.K..BBB..BCBB... ...BBBBB....BBBB... ..BBBBBBB....BBB... ....BB... ....BB.. ....B..	Case 1: AAAEE AAAHE AAAHF AAAHF .G... Case 2: ....B................ ..BBBB.............. ..BBBBBB............ .BBBBBBB............ BBBBBBBBBBBBBBBBB... BBBBBBBBBBBBBBBBB... .BBBBBBBBBBBBBB..... ..BBBBBBBBBBBB...... .BBBBBBBBBB........ ..BBBBBBBBBB..BBB... ...BBBBBBBBBBBBBB... ..BBBBBBBBBBBBBBB... ...BBBBBBBB.BBB.... ...BBBBBBB..BBBB... ...BBBBB....BBBB... ...BBBBBB....BBB... ....BB... ....BB.. ....B..