D. Reconstructing the Grid

Score: 1  
  
CPU: 1s  
Memory: 1024MB

Fox Shial loves to collect grids of numbers. One of his favorite grids had been stolen recently. It had R rows and C columns and the grid had every integers in the range 1 to RxC exactly once in some arbitrary order.

For each integer n in the range 1 to RxC (inclusive), Fox Shial remembers the numbers that were adjacent to n in the stolen grid. A cell (x, y) is adjacent to at most four other cells (x-1,y), (x+1,y), (x,y-1), (x,y+1).

Your task is to reconstruct the grid for Shial. If there are multiple possible grids, find the one that is lexicographically smallest.

A grid G1 is lexicographically smaller than some other grid G2, if the following condition holds true: If we traverse both of the grids in the row major order and if (x,y) is the first cell where the G1[x][y] ≠ G2[x][y], then G1[x][y] < G2[x][y]. (Here, (x,y) denotes the cell at row x and column y). Note: Any cell (x1, y1) comes before (x2, y2) in a row major order, if and only if either (x1 < x2) or (x1 == x2 and y1 < y2) holds true.

Input

The first line contains an integer T denoting the number of test cases. Each test case begins with a line containing 2 integers R and C where R is the number of rows and C is the number of columns in the stolen grid. Each of the next RxC lines contains a list of numbers. The ith line starts with an integer ki and then ki distinct space-separated integers follow. All these integers will be in the range 1 to RxC (inclusive). Here ki is the number of integers adjacent to the number i in the stolen grid. The numbers following ki are all of those adjacent integers in an arbitrary order. It is guaranteed that, if some integer u is adjacent to some other integer v, then v is also adjacent to u. No integer is adjacent to itself.

Output

For the output of each input case, print the serial of the input on a single line and then print the grid in the following format.

Each row should be printed on a different line. Every number of a row should be printed with exactly 1 space between the numbers. There should be no space at the end of a row. (See the sample input output).

If the given input is invalid (i.e. there is no grid that satisfies the given adjacency information) print “NO SUCH GRID” (without quote).

Constraints

1 ≤ T ≤ 40 1 ≤ R, C ≤ 100 0 ≤ ki ≤ 4

Sample

| **Input** | **Output** |
| --- | --- |
| 2  2 2  2 3 4  2 3 4  2 1 2  2 1 2  1 3  2 2 3  2 1 3  2 1 2 | Case 1:  1 3  4 2  Case 2:  NO SUCH GRID |