

12746 Reconstructing the Grid

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 $R * C$ exactly once in some arbitrary order.

For each integer n in the range 1 to $R * C$ (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 G_1 is lexicographically smaller than some other grid G_2 , 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 $G_1[x][y] < G_2[x][y]$, then $G_1[x][y] < G_2[x][y]$. (Here, (x, y) denotes the cell at row x and column y).

Note: Any cell (x_1, y_1) comes before (x_2, y_2) in a row major order, if and only if either $(x_1 < x_2)$ or $(x_1 == x_2$ and $y_1 < y_2)$ 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 $R * C$ lines contains a list of numbers. The i -th line starts with an integer k_i and then k_i distinct space-separated integers follow. All these integers will be in the range 1 to $R * C$ (inclusive). Here k_i is the number of integers adjacent to the number i in the stolen grid. The numbers following k_i 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.

Constraints:

- $1 \leq T \leq 40$
- $1 \leq R, C \leq 100$
- $0 \leq k_i \leq 4$

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).

Sample Input

```
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
```

Sample Output

Case 1:

```
1 3
```

```
4 2
```

Case 2:

```
NO SUCH GRID
```