H Harmonic Matrix

Input: Standard Input
Output: Standard Output

Matrix is a collection of data, i.e. **[3, 4, 5, 1, 2, 0, 6]** is an example of **1D** matrix of integers. Matrix can be of any dimension. Phase of a matrix is another matrix, defines the comparison of matrix elements with respect to the adjacent elements. Here **3<4, 4<5, 5>1, 1<2, 2>0, 0<6**. If **1** resembles a **'<'** and **0** resembles a **'>'**, phase of the above mentioned matrix is **[1, 1, 0, 1, 0, 1]**.

Now a **2D** matrix can be visualized as a collection of **1D** row matrices placed vertically one after another or as a collection of **1D** column matrices placed horizontally one beside another. The phase of a **2D** matrix is a combination of row phase matrices and column phase matrices. Every single row(/column) of the row(/column) phase matrix is generated from the corresponding row(/column) of the original **2D** matrix. For example, the phase of the **2D** matrix on the left is the combination of two boolean **2D** matrices on the right in the following picture.

83 85 87 15 93 35 84 92 49 21 62 27 90 59 63 26 Original 2D Matrix	1 1 0 0 1 1 0 1 0 0 1 0 Row phase matrix	1 0 0 1 0 0 0 0 1 1 1 0 Column phase matrix

A 2D matrix is harmonic if,

- All the rows of the row phase matrix are same and
- All the columns of the column phase matrix are same.

Following matrix on the left is the shuffled version of the previous one. But it satisfies the above mentioned two criteria. So it's a harmonic matrix.

83 85 87 15 84 92 93 35 21 49 62 26 59 63 90 27 Permuted 2D Matrix	1 1 0 1 1 0 1 1 0 1 1 0 Row phase matrix	1 1 1 1 0 0 0 0 1 1 1 1 Column phase matrix

Given a **2D** matrix of **distinct** integers, your task is to shuffle the elements of the matrix so that it becomes harmonic. For shuffling, you can perform only one kind of operation, take two adjacent elements vertically or horizontally and swap them. You have to sequentially output all the swap operations you need to do to shuffle the given matrix. But the number of swaps you are performing can't be infinite, right?

Input

First line of the input is an integer **T (1 ≤ T ≤ 15)**, the number of test cases. Following lines contain **T** test cases. A case starts with a line containing space separated integers **R** and **C** (**1 ≤ R, C ≤ 1003**) representing the number of row and column of the input matrix. Each of the following **R** lines contains **C** space separated integers which constitutes the input matrix **A**. You can assume that all the elements of matrix A are distinct, strictly positive and do not exceed **1000000009.**

Output

For each test case output contains the test case id in the first line. Next line contains an integer **n**, the number of swap operations you are performing to make the array harmonic. Each of the following **n** lines contains the swap operation formatted with four integers **r1, c1, r2, c2 (1 ≤ r1, r2 ≤ R and 1 ≤ c1, c2 ≤ C)**. Condition **(|r1-r2| + |c1-c2|) = 1** should hold, this means you are swapping two adjacent elements **A(r1, c1)** and **A(r2, c2)**. **The number of swap operations has to be bounded by 2.5*R*C, that means n can be at most 2.5*R*C**. See the sample for exact format. Any valid answer that satisfies the constraints will be accepted.

Sample Input	Output for Sample Input
1 3 3 3 4 6 7 5 8 1 9 2	Case 1: 3 2 1 2 2 3 2 3 3 2 3 3 3

Explanation:

3 4 6 7 5 8 1 9 2 Input Matrix	3 4 6 5 7 8 1 9 2 After swap 2 1 2 2	3 4 6 5 7 8 1 2 9 After swap 3 2 3 3	3 4 6 5 7 9 1 2 8 After swap 2 3 3 3