'''

Students are asked to stand in the ground in rows and cols with their roll numbers.

Given this you have to sort each matrix diagonal in ascending order of their

roll numbers

A Matrix diagonal is a diagonal line of cells starting from some cell in

either the topmost row or leftmost column and going in the bottom-right

direction until reaching the matrix's end.

For example, the matrix diagonal starting from mat[2][0], where mat is

a 6 x 3 matrix, includes cells mat[2][0], mat[3][1], and mat[4][2].

Input Format

-------------

Line1: M, N integers representing row and col size

Line2: space separated integers of size M\*N

Output Format

-------------

space separated integers of size M\*N

Sample Input1:

--------------

3 4

3 3 1 1

2 2 1 2

1 1 1 2

Sample Output1:

---------------

1 1 1 1

1 2 2 2

1 2 3 3

Sample Input2:

--------------

5 6

11 25 66 1 69 7

23 55 17 45 15 52

75 31 36 44 58 8

22 27 33 25 68 4

84 28 14 11 5 50

Sample Output2:

---------------

5 17 4 1 52 7

11 11 25 45 8 69

14 23 25 44 58 15

22 27 31 36 50 66

84 28 75 33 55 68

'''

r,c=list(map(int,input().split()))

l=[]

for i in range(r):

l.append(list(map(int,input().split())))

for j in range(r):

k=j

i=0

ls=[]

while(i<c and k<r):

# print(l[k][i])

# if(abs(k-i)==k):

ls.append(l[k][i])

# print(ls)

i+=1

k+=1

ls.sort()

# print(ls)

k=j

i=0

while(i<c and k<r):

l[k][i]=ls[i]

k+=1

i+=1

for j in range(c-1,-1,-1):

k=j

i=0

ls=[]

while(k<c and i<r):

ls.append(l[i][k])

i+=1

k+=1

ls.sort()

# print(ls)

k=j

i=0

while(k<c and i<r):

l[i][k]=ls[i]

i+=1

k+=1

# print(l)

for i in l:

print(\*i)

For X-Mas, santa claus is preparing a X-Mas Tree with set of Bulbs.

The bulbs are of different voltages, and preparation of tree as follows:

- The bulbs are arranged in level-wise, levels are numbered from 0,1,2,3..

so on.

- At level-0: There will be only one bulb as root bulb.,

- From next level onwards, we have to attach two bulbs to left side,

and right side of every bulb in previous level.

- At each level, there must be two bulbs attached to its previous level bulbs.

- After the last level, there will be noe more bulbs attached.

You will be given the root of the X-Mas Tree (i.e.,perfect binary tree),

Your task is to return the root of the structure, after you reverse the bulbs

arrangement at ODD levels.

Implement the class Solution:

1.public Node reverseValuesInOddLevel(Node root): returns the list of integers.

Input Format:

-------------

An integer, number of bulbs.

A single line of space separated integers, voltages of the set of N bulbs.

Output Format:

--------------

Print the list of voltages of the bulbs from the root.

7

Sample Input-1:

---------------

7

1 2 3 4 5 6 7

Sample Output-1:

----------------

1 3 2 4 5 6 7

Sample Input-2:

---------------

15

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

Sample Output-2:

----------------

1 3 2 4 5 6 7 15 14 13 12 11 10 9 8

/\*

//TreeNode Structure for Your Reference..

class Node{

public int data;

public Node left, right;

public Node(int data){

this.data = data;

left = null;

right = null;

}

}

\*/

import java.util.\*;

class Solution {

public Node reverseValuesInOddLevel(Node root) {

// Implement Your Code here..

// Stack<Integer> st=new Stack<>();

Node current=root;

Queue<Node> q=new LinkedList<>();

q.offer(current);

// qq.offer(current);

int count=1;

while(!q.isEmpty()){

int n=q.size();

Queue<Node> qq=new LinkedList<>();

// q.remove()

// int cc=1;

Stack<Integer> st=new Stack<>();

for(int i=0;i<n;i++){

Node curr=q.poll();

qq.offer(curr);

if(curr.left!=null){

q.add(curr.left);

// qq.add(curr.left);

if(count%2!=0){

st.push(curr.left.data);

}

}

if(curr.right!=null){

q.add(curr.right);

// qq.add(curr.right);

if(count%2!=0){

st.push(curr.right.data);

}

}

}

if(count%2 !=0 ){

for(int i=0;i<n;i++){

Node curr=qq.poll();

if(curr.left!=null){

curr.left.data=st.pop();

}

if(curr.right!=null){

curr.right.data=st.pop();

}

}

}

count+=1;

}

return current;

}

}

Mr Ajay Sharma is working woth words.

He found that few words in the langugage have same meaning.

Such words are given as a list of pairs as mappedpairs[],

where mappedpairs[i] = [word1, word2] indicates that word1 and word2 are

the words with same meaning.

He is given phrase, and he wants to apply all the mappedpairs[] on the phrase.

Your task is to help Mr.Ajay Sharma to find and return all possible

Mapped Phrases generated after applying all the mapped words,

and print them in lexicographical order.

Input Format:

-------------

Line-1: An integer N, number of mapped pairs.

Next N lines: Two space separated words, mappedpair[].

Last Line: A line of words, the phrase.

Output Format:

--------------

Print the list of mapped phrases in sorted order.

Sample Input-1:

---------------

3

hi hello

sweet sugar

candy chocolate

hi sam! ram has a sugar candy

Sample Output-1:

----------------

[hello sam! he has sugar candy, hello sam! he has sugar chocolate,

hello sam! he has sweet candy, hello sam! he has sweet chocolate,

hi sam! he has sugar candy, hi sam! he has sugar chocolate,

hi sam! he has sweet candy, hi sam! he has sweet chocolate]

Sample Input-2:

---------------

2

hi hey

hey hello

hi how are you

Sample Output-2:

----------------

[hello how are you, hey how are you, hi how are you]