Problem B. Browsing the Collection

Input file:	*standard input*
Output file:	*standard output*
Time limit:	4 seconds
Memory limit:	512 mebibytes

You are browsing an online collection of n items numbered from 1 to n arranged on a circle. The item to the right of each item i is item $i+1$, and the item to the right of item n is item 1. Similarly, the item to the left of each item i is item $i-1$, and the item to the left of item 1 is item n.

The items have m *parameters* numbered from 1 to m. The value of parameter j for item i is an integer $a_{i,j}$.

While you are browsing, at any moment, there is a pointer directed at some item, called the *current* item.

Moreover, you can manage a set of *filtering conditions*. Each condition is a pair (j, v), meaning that the j-th parameter of the item must be equal to v. The current item always satisfies all conditions in the set.

To browse through the collection, you can do *operations*. Each operation must have one of the following four kinds:

- Click "right". The pointer moves to the closest item to the right of the current item satisfying all filtering conditions. In particular, if the current item is the only such item, the pointer does not move.
- Click "left". Similarly, the pointer moves to the closest item to the left of the current item satisfying all filtering conditions, and stays at the same place if the current item is the only such item.
- Add a new filtering condition (j, v), for some integers j and v. If the current item satisfies this condition, the pointer does not move. Otherwise, the pointer moves to the closest item *to the right* of the current item satisfying all filtering conditions, including the new one. If there is no such item, the operation is illegal and can not be performed.
- Remove any filtering condition (j, v) from the set. The pointer does not move.

For each ordered pair of items (i, j), answer the following question:

- If you start browsing the collection with the pointer directed at item i and with no filtering conditions in the set, what is the smallest number of operations you need to move the pointer to item j? The set of filtering conditions may be arbitrary at the end.

Input

The first line contains two integers n and m, denoting the number of items in the collection and the number of parameters each item has ($2 \le n \le 500$; $1 \le m \le 5$).

The i-th of the next n lines contains m integers $a_{i,1}, a_{i,2}, \ldots, a_{i,m}$, denoting the parameter values of item i ($1 \le a_{i,j} \le n$).

Output

Print n lines containing n integers each.

In the i-th line, the j-th integer must be equal to the smallest number of operations required to move the pointer from item i to item j, starting with an empty set of filtering conditions.

Example

standard input	*standard output*
9 3 5 3 7 5 3 4 5 3 7 5 3 2 5 3 4 5 3 7 2 3 7 5 3 7 2 3 7	0 1 2 1 2 3 1 2 1 1 0 1 1 2 2 1 3 2 2 1 0 1 1 2 1 3 2 3 2 1 0 1 1 1 3 2 3 2 2 1 0 1 1 3 2 3 1 2 1 1 0 1 2 2 2 1 3 1 2 1 0 1 2 2 1 3 1 2 2 1 0 1 1 1 3 1 2 3 2 1 0

Note

In the example test, here is one possible fastest way to move from item 2 to item 5:

- Add a new filtering condition $(3, 4)$. Since item 2 indeed has the value of parameter 3 equal to 4, the pointer stays at item 2.
- Click "right". The pointer moves to the closest item to the right of item 2 satisfying the only active condition $(3, 4)$. This item is item 5. (Clicking "left" instead works as well.)

Here is one possible fastest way to move from item 8 to item 3:

- Add a new filtering condition $(3, 4)$. Since item 8 does not satisfy this condition, the pointer moves to the closest item to the right of item 8 with the value of parameter 3 equal to 4. This item is item 2.
- Remove the filtering condition $(3, 4)$. The pointer stays at item 2.
- Click "right". Since there are no filtering conditions in the set, the pointer moves to item 3.