Jealousy II

After a successful year, your business is relocating to a larger office. Your business has N employees (numbered from 1 to N), and the new office has M desks (numbered from 1 to M), where $M \geq N$.

Each employee will be assigned their own desk and, to maintain company morale, you want this assignment to be *jealousy-free*. Each employee has assigned an integer value to each desk: employee i values desk j with value $v_{i,j}$. An assignment of desks is considered *jealousy-free* if every employee believes that their desk is **strictly** better than every other employee's desk. More formally:

- Each employee i will be assigned a distinct desk a_i.
- For each pair of different employees i and j, it must hold that $v_{i,a_i} > v_{i,a_j}$. Note that employee i only considers their own values. Specifically, employee i is not concerned with how employee j values their assigned desk.

If $M > N$, then some desks will be unassigned. These desks do not cause any jealousy.

You must find a jealousy-free assignment, or determine that no such assignment exists.

Subtasks and Constraints

For all subtasks:

- $2 \leq N \leq 1\,500$.
- $N \leq M \leq 2\,500$.
- $0 \leq v_{i,j} \leq 10\,000\,000$ for all i and j.

Additional constraints for each subtask are given below.

Subtask	Points	Additional constraints
1	5	$N = 2$.
2	15	$N = M$ and all v-values are distinct. That is, $v_{i,j} \neq v_{i',j'}$ unless $i = i'$ and $j = j'$.
3	15	$v_{i,j} \leq 1$ for all i and j.
4	30	$N, M \leq 200$ and all v-values are distinct. That is, $v_{i,j} \neq v_{i',j'}$ unless $i = i'$ and $j = j'$.
5	20	All v-values are distinct. That is, $v_{i,j} \neq v_{i',j'}$ unless $i = i'$ and $j = j'$.
6	15	No additional constraints.

Input

- The first line of input contains the two integers N and M.
- The next N lines contains M integers. The jth integer on the ith line is $v_{i,j}$.

Note on Input

This problem has a large amount of input. To speed up input, those using `cin` in C++ are encouraged to include the following line at the beginning of their main function:

```
ios_base::sync_with_stdio(false);
```

If you are using `scanf` for some or all of the input, you should not include this line.

Output

The first line of output should be either `POSSIBLE` or `IMPOSSIBLE`, depending on whether a jealousy-free assignment exists. If your answer is `POSSIBLE`, then the next N lines should describe an assignment. In particular, the ith such line should contain the single integer a_i: the desk assigned to employee i. If there are multiple possible jealousy-free assignments, then you may output any of them.

Sample Input 1

```
2 3
4 3 2
5 1 5
```

Sample Output 1

```
POSSIBLE
2
3
```

Sample Input 2

```
2 2
0 1
1 1
```

Sample Output 2

```
IMPOSSIBLE
```

Explanation

In Sample Input 1, the only jealousy-free assignment has $a_1 = 2$ and $a_2 = 3$. In particular:

- Employee 1 believes that their desk is strictly better than employee 2's desk, since $v_{1,a_1} = 3$ and $v_{1,a_2} = 2$.
- Employee 2 believes that their desk is strictly better than employee 1's desk, since $v_{2,a_2} = 5$ and $v_{2,a_1} = 1$.

In Sample Input 2, there is no jealousy-free assignment. In particular:

- If $a_1 = 1$ and $a_2 = 2$, then $v_{1,a_1} < v_{1,a_2}$ and so this assignment is not jealousy-free.
- If $a_1 = 2$ and $a_2 = 1$, then $v_{2,a_2} = v_{2,a_1}$ and so this assignment is not jealousy-free.

Since there are no other possible assignments, the correct output is `IMPOSSIBLE`.