

B. Multi-core Processor

time limit per test: 1 second

memory limit per test: 256 megabytes

input: standard input

output: standard output

The research center Q has developed a new multi-core processor. The processor consists of n cores and has k cells of cache memory. Consider the work of this processor.

At each cycle each core of the processor gets one instruction: either do nothing, or the number of the memory cell (the core will write an information to the cell). After receiving the command, the core executes it immediately. Sometimes it happens that at one cycle, multiple cores try to write the information into a single cell. Unfortunately, the developers did not foresee the possibility of resolving conflicts between cores, so in this case there is a *deadlock*: all these cores and the corresponding memory cell are locked forever. Each of the locked cores ignores all further commands, and no core in the future will be able to record an information into the locked cell. If any of the cores tries to write an information into some locked cell, it is immediately locked.

The development team wants to explore the deadlock situation. Therefore, they need a program that will simulate the processor for a given set of instructions for each core within m cycles. You're lucky, this interesting work is entrusted to you. According to the instructions, during the m cycles define for each core the number of the cycle, during which it will become locked. It is believed that initially all cores and all memory cells are not locked.

Input

The first line contains three integers n, m, k ($1 \leq n, m, k \leq 100$). Then follow n lines describing instructions. The i -th line contains m integers: $x_{i1}, x_{i2}, \dots, x_{im}$ ($0 \leq x_{ij} \leq k$), where x_{ij} is the instruction that must be executed by the i -th core at the j -th cycle. If x_{ij} equals 0, then the corresponding instruction is «do nothing». But if x_{ij} is a number from 1 to k , then the corresponding instruction is «write information to the memory cell number x_{ij} ».

We assume that the cores are numbered from 1 to n , the work cycles are numbered from 1 to m and the memory cells are numbered from 1 to k .

Output

Print n lines. In the i -th line print integer t_i . This number should be equal to 0 if the i -th core won't be locked, or it should be equal to the number of the cycle when this core will be locked.

Examples

input
4 3 5 1 0 0 1 0 2 2 3 1 3 2 0
output
1 1 3 0

input
3 2 2 1 2 1 2 2 2
output

1
1
0

input
1 1 1
0
output
0