In order to celebrate the 100-th anniversary of Tsinghua University, n girls are planning to hold a party. They're experts at singing and dancing, and they love to perform in groups. In their current design, there will be a stage and a row of seats. When a group of girls need to sing or dance on the stage, they stand up from their seats, and go to the stage. When they've finished performing, they return to their own seat (they don't exchange seats, because every girl has a lot of personal belongings on her seat).

They want this procedure look cool, so for each performance, the actresses' seats should be consecutive. For example, if there are 4 girls, and a performance is done by girl 1, 2 and 4, then they cannot seat in order of 1-2-3-4, since when girl 1, 2 and 4 stand up, it's strange to see a non-actress (girl 3) sitting between girl 2 and 4.

As I mentioned, they're too good at singing and dancing, so they managed to come up with a lot of combinations. Now they become a bit worried: is there a way to seat all the girls, such that the requirement above can be satisfied (i.e. for every combination, the actresses' seats are consecutive).

As a decent programmer, you decide (I know that actually you're being decided, but...) to write a program that can calculate the number of seat arrangements. Since the girls' are constantly thinking about new combinations, your program should be able to read new combinations and adjust the answer accordingly. When there are only few possible arrangement (i.e. at most k feasible solutions), your program should output all of them.

Input

There are several test cases. The first line contains three integers n, m, k ($1 \leq n, m, k \leq 200$), where n is the number of girls, m is the number of combinations, and k is the parameter described above. Each of the next m lines contains a set of integers, terminated by a zero. These integers are the IDs of the girls in the combination (girls are numbered 1 to n). The input is terminated by end-of-file (EOF).

Output

For each new combination, output the number of seat arrangements, after considering this combination. If there is no way, print '0' and **ignore the combination**. If there are at most k ways, print them one in a line, in lexicographical order.

Sample Input

```
4 4 10
1 2 3 0
2 3 4 0
1 4 0
2 4 0
```

Sample Output

```
12
4
1 2 3 4
1 3 2 4
4 2 3 1
4 3 2 1
0
2
1 3 2 4
4 2 3 1
```