Problem J. Lowest Unique

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	512 mebibytes

This is an interactive problem.

The *lowest unique positive integer* game is often used as an example of a game with very simple rules that is quite tricky for the computers to master.

n players are playing this game, and it consists of m rounds. In each round, each player chooses a positive integer in secret. The chosen integers are then revealed at the same time, and the player which has the lowest integer that was not chosen by any other player wins the round. In case all integers are repeated, there is no winner in that round.

In this problem you control k players, $k > \frac{n}{2}$ (strictly more than half of all players), and your goal is for one of your players (not necessarily the same one) to win at least 90% of the rounds.

The other $n - k$ players will each play using a predetermined strategy that does not depend on your moves.

Interaction Protocol

First, your program needs to read three integers n, k and m ($3 \le n \le 10$, $\frac{n}{2} < k < n$, m is either 2 or 1000, and m is 1000 in all cases except the sample case.

Then you need to repeat the following process m times: you need to print k integers between 1 and 100, inclusive, denoting the moves of the players you control. Remember to flush the output after doing that. Then you need to read $n - k$ integers, which will also be between 1 and 100, inclusive, denoting the moves of the remaining players in the round.

Your program must exit gracefully after completing m rounds.

Your solution must win at least $0.9 \cdot m$ rounds, rounded up.

Example

standard input	standard output
5 3 2 1 1 2 3	2 3 3 100 50 1

Note

In the sample case, the other players will always choose 1 and 1 in the first round, and 2 and 3 in the second round.

There are 100 non-sample test cases in this problem.