

## Problem A. Dreissig

Input file: *standard input*  
Output file: *standard output*  
Time limit: 15 seconds  
Memory limit: 256 mebibytes

Two players are playing a game on the complete undirected graph with 100 vertices. Initially all its  $\frac{100 \times 99}{2} = 4950$  edges are uncolored. The players take turns in coloring the edges. The first player moves first, and at each move picks any 30 uncolored edges (or all uncolored edges if there are less than 30) and colors all of them black. The second player at each move picks any uncolored edge and colors it white. The game ends when all edges are colored.

The second player wins if they can point out a *Hamiltonian cycle* consisting only of white edges, otherwise the first player wins. A Hamiltonian cycle is a simple cycle that passes through each vertex exactly once.

In this problem you need to play this game as the second player. Moreover, you already know the strategy of the first player: they will pick 30 edges uniformly at random from all remaining edges. Can you win at least 95 out of 100 games as the second player?

### Interaction Protocol

Initially your program should read three integers  $n$ ,  $k$  and  $g$ : the number of vertices in the graph, the number of edges that the first player takes on each move, and the number of games to play. In all cases except the sample case  $n = 100$ ,  $k = 30$ , and  $g = 100$ .

Then your program should play exactly  $g$  games, and finish execution gracefully after the last game concludes.

The games are played as follows. If there's at least one uncolored edge and it's the first player's turn, your program should read the first player's move. It starts with the number  $m$  of edges colored black, followed by  $m$  pairs of integers — the numbers of vertices connected by edges to be colored black. It is guaranteed that the first player will play according to the problem statement:  $m$  will always be equal to  $k$  unless there are less than  $k$  uncolored edges remaining, in which case  $m$  will be equal to the number of uncolored edges. The edges will be picked uniformly at random from all uncolored edges.

If there's at least one uncolored edge and it's the second player's turn, your program should output the second player's move as two integers — the numbers of vertices connected by the edge to be colored white. The vertices are numbered with integers between 1 and  $n$ . The edge you output must be uncolored at this point. Remember to flush standard output after printing your move.

If there are no more uncolored edges, your program should print the Hamiltonian cycle it constructed as a permutation of numbers between 1 and  $n$  — the numbers of the vertices in the cycle order. If your program admits defeat for this game it should output a single number -1 instead of the Hamiltonian cycle. Note that it must still play the game to completion before doing that. You're allowed to admit defeat even if there is in fact a Hamiltonian cycle with white edges. Again, remember to flush standard output after printing the cycle or -1.

Your solution will be accepted if your program builds a Hamiltonian cycle in at least 95 out of 100 games in each non-sample case, and plays any valid games in the sample case (note that if you decide to output a Hamiltonian cycle in the sample case, it still has to be valid). There are 20 non-sample cases in this problem.

## Example

standard input	standard output
6 1 2	
1 4 2	5 2
1 3 1	5 4
1 4 1	6 3
1 6 2	4 3
1 6 4	3 2
1 6 1	5 1
1 6 5	5 3
1 2 1	-1
1 3 6	1 2
1 2 5	2 3
1 4 6	3 4
1 5 1	5 4
1 1 4	5 6
1 2 6	1 6
1 2 4	1 3
1 5 3	1 2 3 4 5 6