Problem J. Redundant Edges

Input file:	`redundant.in`
Output file:	`redundant.out`
Time limit:	2 seconds
Memory limit:	256 megabytes

Consider a directed graph G consisting of N nodes and M directed edges. Nodes are numbered from 1 to N and edges are numbered from 1 to M. Let's select some node R as the starting one and find all nodes that are reachable from R by moving along edges (denote this set as C_0). Define $C(e)$ as the set of nodes which are reachable from R using any edges except the one with number e. Edge e is called *redundant* if $C(e)$ is equal to C_0.

You are given the graph G and the starting node R. Your goal is to find all *redundant* edges.

Input

The first line of input contains three integers N, M and R ($1 \le N \le 100\,000$, $1 \le M \le 200\,000$, $1 \le R \le N$): the number of nodes, the number of edges and the starting node, respectively. Next M lines describe the edges of the graph: i-th of them contains two integers a_i and b_i ($1 \le a_i, b_i \le N$) which introduce a directed edge from node a_i to node b_i. It is guaranteed that there are no loops, and for any two nodes u and v, there is at most one edge from u to v.

Output

On the first line, print the number of *redundant* edges. On the second line, print the numbers of all *redundant* edges in any order. Edges are numbered starting from 1 according to their order in the input.

Examples

redundant.in	redundant.out
3 3 1 1 2 2 3 3 1	1 3
4 6 2 2 1 2 3 3 1 3 4 1 4 4 2	5 1 3 4 5 6