Network

Our engineers have designed a communication network that consists of nodes and unidirectional direct communication channels (links) between some pairs of nodes. We say that a node ***q*** is reachable from node ***p***, if there is a sequence of links that does not meet any node more than once, and through which data can be transmitted from ***p*** to ***q***. This network has a central node ***r***, such that any other node ***p*** can be reached from ***r*** via a *unique* sequence of links. The maintainers plan to improve on the network, but have not yet decided how. One idea they are considering is to reassign the central node, therefore they want to know for each node how many nodes are reachable from it. Another idea is to just decentralize the network, so they also want to know how they could introduce new links so that for any pair of nodes ***p*** and ***q***, the node ***p*** is reachable from ***q*** in a unique way, and vice versa.

Task

You are to write a program that computes the number of reachable nodes for every node (Subtask A), and also computes the minimum number of new links needed to make every node reachable in a unique way from every other node. Your program must give the list of new links, too (Subtask B).

Input

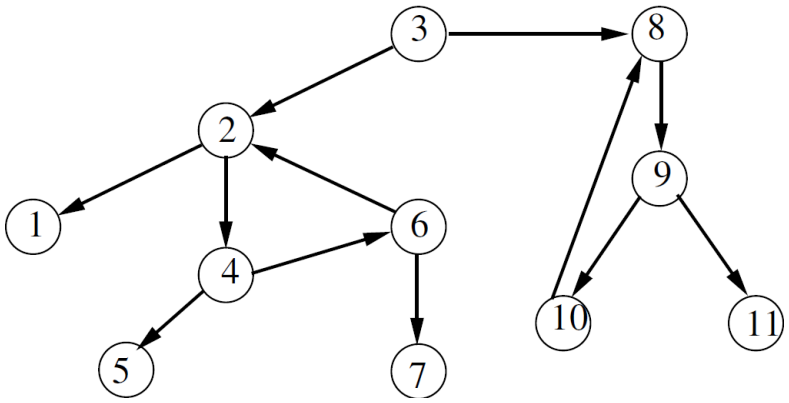
The first line of the input contains three integers, ***N*** (***1 ≤ N ≤ 100 000***) the number of nodes, ***M*** (***1 ≤ M ≤ 500 000***) the number of links, and ***r*** (***1 ≤ r ≤ N***) the central node. Nodes are numbered from ***1*** to ***N***. The next ***M*** lines contain the description of the links. Each line contains two integers ***p*** and ***q*** separated by space, that corresponds to a link, which can transmit data from ***p*** to ***q***.

Output

The first line of the output contains the solution to Subtask A: ***N*** integers separated by space, where the ***i***th number is the number of reachable nodes from node ***i*** (including ***i*** itself). The next ***K+1*** lines contain the solution for Subtask B: The second line of the output contains one integer ***K***, the minimum number of new links needed to achieve the above property of the network. The next ***K*** lines list the new links: each of lines contains two integers ***u*** and ***v*** separated by space, that corresponds to a new link transmitting data from node ***u*** to node ***v***. If there are multiple solutions, your program should output only one; it does not matter which one.

Example input and output

input	output
11 12 3 3 2 2 1 2 4 4 5 4 6 6 2 6 7 3 8 8 9 9 10 9 11 10 8	1 6 11 6 1 6 1 4 4 4 1 5 1 3 5 4 7 6 11 9 8 3



Limits

Memory limit: 32 MB
Time limit: 0.1 sec

Grading

In 50% of the test cases ***N*** is at most ***10 000***.
Subtask A is worth 40% of the points, Subtask B is worth the other 60% of the points.
If you only solve Subtask B, you must write ***N*** integers in the first line.