## The Goal

Skynet is responding to your intrusion and has activated additional defenses:

* Most of the links have been reinforced: your virus no longer has the possibility to destroy a link between two ordinary nodes, it can now only sever links leading to gateways. If it attempts to cut a link between two ordinary nodes it will be detected and deleted.
* Nodes may now be connected to up to two gateways!

There ar **six new tests** to pass in this final confrontation.  
  
Do not hesitate to copy/paste your code from level 1 to level 2.  
  
***Recap:***  
  
Your virus has caused a backdoor to open on the Skynet network enabling you to send new instructions in real time.  
  
You decide to take action by stopping Skynet from communicating on its own internal network.  
  
Skynet's network is divided into several smaller networks, in each sub-network is a Skynet agent tasked with transferring information by moving from node to node along links and accessing gateways leading to other sub-networks.  
  
Your mission is to reprogram the virus so it will sever links in such a way that the Skynet Agent is unable to access another sub-network thus preventing information concerning the presence of our virus to reach Skynet's central hub.

## Rules

For each test you are given:

* A map of the network.
* The position of the exit gateways.
* The starting position of the Skynet agent.

Each game turn:

* First off, you sever one of the given links in the network.
* Then the Skynet agent moves from one Node to another accessible Node.

|  |  |
| --- | --- |
| *Skynet agent* | *Gateway* |

Victory Conditions

The Skynet agent cannot reach an exit gateway

Lose Conditions

The Skynet agent has reached a gateway

## Example

4 4 1

0 1

0 2

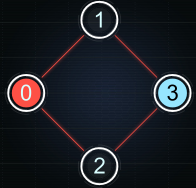
1 3

2 3

3

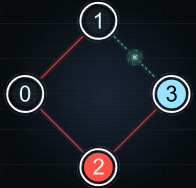
Initialization input

Text representation of the network used in this example. There are 4 nodes, 4 links and 1 gateway. The next 4 lines describe the links. The last integer is the index of the exit node.



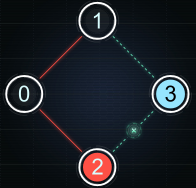
Turn 1

The Skynet agent starts at node 0 (SI = 0). Our virus cut the link between the nodes 1 and 3.



Turn 2

The Skynet agent moves to node 2 (SI = 2). Our virus cut the link between the nodes 2 and 3.



Turn 3

The Skynet agent has been cut off from the exit, you have won !

## Note

The tests provided are similar to the validation tests used to compute the final score but remain different.

## Game Input

The program must first read the initialization data from standard input. Then, within an infinite loop, read the data from the standard input related to the current state of the Skynet agent and provide to the standard output the next instruction.

Initialization input

Line 1: 3 integers N L E  
- N, the total number of nodes in the level, including the gateways.  
- L, the number of links in the level.  
- E, the number of exit gateways in the level.

Next L lines: 2 integers per line (N1, N2), indicating a link between the nodes indexed N1 and N2 in the network.

Next E lines: 1 integer EI representing the index of a gateway node.

Input for one game turn

Line 1: 1 integer SI, which is the index of the node on which the Skynet agent is positioned this turn.

Output for one game turn

A single line comprised of two integers C1 and C2 separated by a space. C1 and C2 are the indices of the nodes you wish to sever the link between.

Constraints

2 ≤ N ≤ 500  
1 ≤ L ≤ 1000  
1 ≤ E ≤ 20  
0 ≤ N1, N2 < N  
0 ≤ SI < N  
0 ≤ C1, C2 < N  
Response time per turn ≤ 150ms