4112 Scotland Yard

This game of Scotland Yard consists of K cops chasing one thief through a network of interconnected cities. All players' initial positions are known. On every move, all players must choose a road to move on. All players move simultaneously. The roads that the thief takes are not known to the cops and the roads that the cops take are not known to the thief. However after every five moves they reveal their positions to each other.

Two cops can stay in the same city, while the thief is caught as soon as he reaches a city where a cop is. The cops want to catch the thief as soon as possible, no matter how the thief plays. Can you tell me if this is possible, if the cops plan well ahead?

Input

Input consists of multiple testcases.

The first line of each testcase contains three integers N, M and K, representing the number of cities, the number of roads, and the number of cops, respectively. Cities are numbered between 1 and N. $(1 \leq K \leq 2, K < N \leq 10)$.

The second line contains $K+1$ integers, the first giving the initial location of the thief, and the rest giving the initial locations of the cops. All initial locations will be distinct.

Subsequent M lines contain two integers, first being the source city and the second being the destination city that the one-way road connects.

It is also possible that the source city and destination city are the same. You may assume there exists at least one road leaving from any given city.

Input ends with a line containing three zeros separated by spaces.

Output

Output to each test case must begin with a line containing '`Case #`$< case - number >$`:`'.

Then on the next line, print '`The thief is caught in` x `moves.`', if it is possible to catch the thief in x moves.

Print such minimal $x \leq 20$. If it is not possible to catch the thief in 20 moves, print '`The thief cannot be caught within 20 moves.`'.

Sample IO explanation:

Test case `#1` consists of a straight line with 4 cities. After every move the thief stays at a city with a number different from the cop modulo 2. Hence he can never be caught.

Test case `#2` consists of the same straight line with 5 cities now. Here is what the cop does to catch the thief:

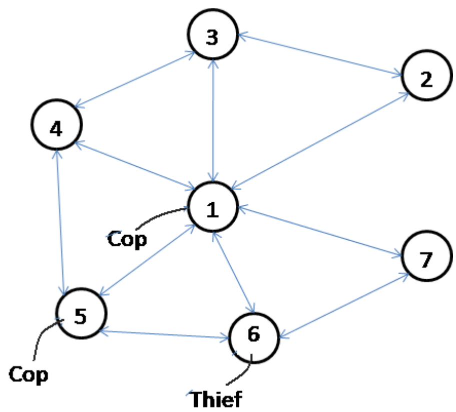
move#1: move to 4

move#2: move to 3

move#3: move to 2

At move #1 the thief can only be at 2. At move #2 he can be at 1 or 3. If he's at 3, he's caught immediately. If he's at #1, he has to move to #2 at the next move and hence caught at move#3.

Test case #3 looks like the figure on the right:



Situation after 5 moves.

The best-strategy here is to force the thief into the above situation (or the symmetric dual of the situation), after 5 moves. From here, in the next move, force the thief to the corner node 7 by having the cops exchange places. And the next move (7 th) he's caught, if the cop at 1 goes to 6, and the cop at 5 goes to 1.

Sample Input

```
4 6 1
1 4
1 2
2 1
2 3
3 2
3 4
4 3
5 8 1
1 5
1 2
2 1
2 3
3 2
3 4
4 3
4 5
5 4
7 22 2
1 2 3
1 2
2 1
1 3
3 1
1 4
4 1
1 5
5 1
1 6
6 1
1 7
7 1
2 3
3 4
4 5
5 6
6 7
7 6
6 5
5 4
4 3
3 2
0 0 0
```

Sample Output

```
Case #1:
The thief cannot be caught within 20 moves.
Case #2:
The thief is caught in 3 moves.
Case #3:
The thief is caught in 7 moves.
```