K – Keys

Time limit: 4 s *Memory limit: 256 MiB*

Alice and Bob live in a massive mansion with n rooms (one of them denoting outdoors, where they play the Moon game) and m doors between them. Each door connects two rooms or a room with the outdoors and has a single unique key that opens this door only. Every door closes behind you and locks automatically, so one always needs a key to pass through. The building is very large, but Alice and Bob only use one room – their bedroom. Other rooms are only there to make the house look bigger and make the neighbors jealous.

This unusual way of building their house is now causing some trouble for Alice and Bob. Bob is leaving for a two-week-long trip. In a week, Alice is also going abroad for a month and when she leaves, she needs the right keys to leave the house. However, Bob also needs keys to get back in since Alice will not be there at the time to let him in. Alice and Bob are now trying to figure out how to split the keys to the doors for Alice to be able to get from room 0 (their bedroom) to 1 (outdoors) and Bob to go from room 1 (outdoors) to 0 (bedroom) one week later.

Fortunately, Alice remembered that she could drop some of the keys on her way out for Bob to pick up on his way back. This way, they can both pass through the same doors. She, of course, cannot drop any keys in room 1 (outdoors) since the neighbors could find them and break into their house.

Can you help Alice and Bob split their keys and plan their trip through the house?

Task

You are given the description of Alice and Bob's mansion: m doors between n rooms numbered 0 to $n-1$, where 1 is outdoors and 0 is the bedroom. The i-th door can be opened with the key number i (0-indexed).

You should first print two lines containing space-separated numbers of keys for Alice and Bob, respectively. It is fine if they do not use all the keys, but it is not allowed for them to both have a copy of the same key (or for one of them to have multiple copies of a key).

You should then print instructions that Alice and Bob will follow. First, print Alice's movements from room 0 to 1 by printing commands of one of the two types:

- "`MOVE x`" to move to room x (assuming that there is a door between Alice's current location and x and that Alice has the key),
- "`DROP k1 k2 ...`" to drop keys $k_1, k_2, \ldots$ (printed as space-separated integers) in the room where Alice currently stands. This means that Alice no longer carries these keys.

Once Alice is done with her movements, print "`DONE`" in a new line. She should finish her movement in room 1. It is fine if Alice passes through room 0 or 1 multiple times while following the printed instructions.

Second, print Bob's movements from room 1 to 0 by printing commands of one of the two types:

- "`MOVE x`" to move to room x (assuming that there is a door between Bob's current location and x and that Bob has the key),
- "`GRAB`" to grab any keys in the room where Bob is currently standing. Bob always grabs all the keys that Alice left in the room. If Alice left none, he does not grab any.

Once Bob is done with his movements, print "`DONE`" in a new line. He should finish his movement in room 0. It is fine if Bob passes through room 0 or 1 multiple times while following the printed instructions.

Remarks: It is considered acceptable, albeit useless, to `DROP` an empty list of keys or to grab keys in a room that has no keys, or grab keys in room 1 (i.e. outside).

Input data

The first line contains integers n and m, the number of rooms (outdoors included) and number of doors. This is followed by m lines that describe the doors. The i-th line (count starting from 0) is described by integers a_i, b_i, indicating that there is a door between rooms a_i and b_i, opened with key i.

Input and output limits

- $2 \leq n, m \leq 10^5$
- $0 \leq a_i, b_i < n$
- It is guaranteed that it's always possible to reach any room from any other room if you have all the keys.
- Each pair of rooms is connected with at most one door.
- No room is connected to itself.
- Your program may print at most $4 \cdot 10^5$ instructions.

Output data

First, print two lines, describing the split of keys. Then, print all the instructions for Alice and Bob, as described in the task description, one per line. If there is no solution, print "`No solution`" (without the quotes). If there are multiple valid solutions, any of them is accepted.

Examples

Input

```
5 5
0 1
1 2
2 3
3 4
4 1
```

Output

```
0 1 2
3 4
MOVE 1
MOVE 2
MOVE 3
DROP 0
MOVE 2
MOVE 1
DONE
MOVE 4
MOVE 3
GRAB
MOVE 4
MOVE 1
MOVE 0
DONE
```

Input

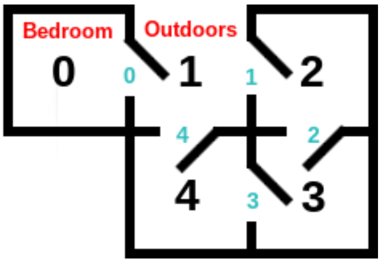
```
3 2
0 2
1 2
```

Output

```
No solution
```

Comment

The first example represents the following floor plan, where blue numbers correspond to key IDs required for each door:



Alice takes keys 0, 1, and 2 while Bob takes keys 3 and 4. Alice walks from 0 to 1, then to 2, and then to 3. There, she drops key 0. She retraces her way back to 1. Bob starts in 1, walks to 4, then to 3, where he picks up key 0. He retraces his way back to 1, and with the newly picked-up key 0, opens up the door to 0.

In the second example, there is no way for both Alice and Bob to reach their destination. Note that Alice cannot drop keys in room 1.