

## A: Are we connected now?

Consider an emerging connectivity network. An example would be train stations at different locations. Say there are  $N$  cities, each independently building a train station. At first there aren't any connections between cities, but over time we add rails between them. Hopefully every station will be connected to every other station, but in the interim they may not be. At any particular point in time before the network is fully connected, we want to figure out if two cities are connected by a path in the system.

Each station can be represented as a node. Any given node  $X$  is represented by an index, which is in the range  $0 \leq X < N$ . There are two operations that can be performed on nodes.

The first operation is connection creation, which adds a two way connection between a pair of nodes. The second operation is a query, which determines whether or not a pair of nodes have a path between them.

You must write a program that performs these network operations and outputs the results of any queries.

### Input

Each case will start with two space separated integers,  $0 \leq N \leq 2000$ , and  $0 \leq K \leq 10^5$ . The integer  $N$  represents the number of nodes in the network, and  $K$  represents the number of operations to perform.

If line begins with  $+$  followed by two nodes indices each separated by a single space (e.g.  $+ x y$ ), then you should perform a connection creation between nodes  $x$  and  $y$ .

If line begins with  $?$  followed by two nodes indices each separated by a single space (e.g.  $? x y$ ), then you should perform a query operation on nodes  $x$  and  $y$ . Every case contains at least one query.

When  $N$  and  $K$  are both 0, the program should exit.  $N$  and  $K$  will never be 0 otherwise.

### Output

Cases should start by outputting a line of the form "Case X:" where  $X$  is the current case number.

On the following lines, you should output the results of any queries that happen during the execution of the case. Each query output should be on its own line.

### Sample Input

---

```
5 6
+ 0 2
+ 2 4
? 0 4
? 0 3
+ 3 4
? 0 3
4 4
+ 2 3
? 1 2
+ 1 2
? 1 3
0 0
```

---

### Sample Output

---

```
Case 1:
true
false
true
Case 2:
false
true
```

---