**Graph Connectivity**

|  |  |  |
| --- | --- | --- |
| **Time Limit:** 8000MS |  | **Memory Limit:** 131072K |
|  |  |  |
| **Case Time Limit:** 3000MS | | |

**Description**

Let us consider an **undirected** graph G = < V, E >. At first there is no edge in the graph. You are to write a program to calculate the connectivity of two different vertices. Your program should maintain the functions inserting or deleting an edge.

**Input**

The first line of the input contains an integer numbers N (2 <= N <= 1000) -- the number of vertices in G. The second line contains the number of commands Q (1 <= Q <= 20000). Then the following Q lines describe each command, and there are three kinds of commands:   
  
I u v: Insert an edge (u, v). And we guarantee that there is no edge between nodes u and v, when you face this command.   
D u v: Delete an existed edge (u, v). And we guarantee that there is an edge between nodes u and v, when you face this command.   
Q u v: A querying command to ask the connectivity between nodes u and v.   
  
You should notice that the nodes are numbered from 1 to N.

**Output**

Output one line for each querying command. Print "Y" if two vertices are connected or print "N" otherwise.

**Sample Input**

3

7

Q 1 2

I 1 2

I 2 3

Q 1 3

D 1 2

Q 1 3

Q 1 1

**Sample Output**

N

Y

N

Y