Re-Transmission

Rar the Cat is building a new system, dubbed the Cats Transmission System (CTS). This horribly inefficient system utilises cats to help transfer information. In this system, there are **N** cats lined up in a line from cat **0** to cat **N-1**. If a message needs to be transferred from, say, cat 2 to cat 7, cat 2 will pass on the message to cat 3, to cat 4... and so on, until it reaches cat 7. Sounds simple, right?

However, there is a problem. As everyone knows, cats *LOVE* sleeping. Some of these cats tend to fall asleep on the job. Say, if cat 3 falls asleep, the message from cat 2 to cat 7 will not be able to be transmitted. As such, given a list of "SLEEP" and "WAKE" events, as well as "TRANSMIT" requests in between, Rar wants you to check if each of these "TRANSMIT" requests will pass. **However, it is currently night time and all cats start out asleep.**

There will be a total of **Q** events. The format of the events will be as follows:

|  |  |
| --- | --- |
| **Event** | **Description** |
| WAKE **[x]** | Cat **[x]** wakes up.  It is guaranteed cat **[x]** is currently asleep. |
| SLEEP **[x]** | Cat **[x]** falls sleep.  It is guaranteed that cat **[x]** is currently awake. |
| TRANSMIT **[x] [y]** | Attempt to transmit information from cat **[x]** to cat **[y]** (**[x]** ≤ **[y]**).  If it is successful (all cats from **[x]** to **[y]** inclusive are awake), then output “YES”. Otherwise, output “NO”. |

**Input**

The first line of input will contain two integers, **N** and **Q**.

The next **Q** lines of input will each contain one event as stated above.

**Output**

There should be one line of output for every "TRANSMIT" operation, either stating a "YES" or a "NO".

**Limits**

* 0 < **N** ≤ 2,000,000,000
* 0 < **Q** ≤ 300,000.
* All values of **x** and **y** are guaranteed to be between **0** and **N-1**.

**Sample Testcase**

|  |  |
| --- | --- |
| Sample Input (**retransmission1.in**) | Sample Output (**retransmission1.out**) |
| 8 13  TRANSMIT 3 4  WAKE 4  WAKE 3  WAKE 6  WAKE 7  TRANSMIT 4 6  WAKE 5  TRANSMIT 4 6  TRANSMIT 3 7  SLEEP 6  TRANSMIT 3 7  TRANSMIT 3 6  TRANSMIT 3 5 | NO  NO  YES  YES  NO  NO  YES |

**Explanation**

There is a total of 8 cats, labelled from 0 to 7, and 13 events that follows.

**Initially, all the 8 cats are asleep.** Hence, the first transmission request (in fact any transmission request) will fail since none of the cats have woken up.

After that, cats 3, 4, 6, 7 wake up. Since cat 5 is still asleep, the transmission request from 4 to 6 fails.

Then, cat 5 wakes up to the transmission from 4 to 6 succeeds. Since all the cats from 3 to 7 are awake, the transmission from 3 to 7 also succeeds.

Then, cat 6 falls asleep. Subsequently, the transmission request from cats 3 to 7 and cats 3 to 6 both fail since cat 6 is asleep.

However, the transmission request from cats 3 to 5 will still succeed as every cat in the range is still awake.