



## Inside information (servers)

Why did nobody tell you that computer science books can be so *boring*?! You really do not make progress with them, and your BOI medal is getting out of reach. But wait—who says you have to win it fair and square?\*

BOI officials already mentioned that task statements are kept in a secret vault that can only be opened by the chair of the Scientific Committee. So the statements are beyond your reach. However, getting access to test data beforehand sounds manageable and should be enough to give you an advantage. Unfortunately for you, the Scientific Committee has taken measures against possible unfairness. They split up the test data into  $N$  chunks and distributed them among  $N$  servers numbered from 1 to  $N$ . Initially, server  $i$  holds data chunk  $i$ . The  $N$  servers are connected by  $N - 1$  wires in such a way that any two servers are connected to each other either directly or indirectly. From time to time, two servers *share* their data, meaning that afterwards both store the same chunks of data, namely precisely those chunks that were stored by at least one of them before. *Any two servers directly connected to each other, and only these servers, share their data precisely once.*

You are given the whole sequence of share operations. To coordinate your hacking attempts, you want to know at several points, in between two share operations, how test data is currently distributed among the servers. More precisely, you query *whether a given server currently stores a given data chunk or how many servers currently store a given data chunk.*

Write a program that allows you to answer such questions given (a) the sequence of share operations and (b) when each query is made.

### Input

The input describes the sequence of share operations and queries. The first line of input contains two integers  $N$  and  $K$ . Each of the following  $N + K - 1$  lines can have one of the following forms and meanings:

**S a b** means servers  $a$  and  $b$  Share all their data.

**Q a d** means you Query whether server  $a$  currently stores data chunk  $d$ .

**C d** means you query the Count (number) of servers that currently store data chunk  $d$ .

$N - 1$  lines start with an S, and  $K$  lines start with either Q or C.

### Output

For each query Q  $a d$  in the input, your program should output a single line containing the string `yes` if the  $a$ -th server stored the  $d$ -th piece of data at the time of the query, or `no` otherwise. For each query C  $d$  your program should output a single line containing a single integer: the number of servers storing data chunk  $d$  at the time of the query. Of course, the answer to each query only depends on the share operations that happened before the query.

### Constraints

We always have  $1 \leq N, K \leq 120\,000$ .

\* Well, we do, and your team leaders do too, but never mind.



**Subtask 1 (5 points).**  $N \leq 4\,000$

**Subtask 2 (5 points).** Server 1 is directly connected to servers 2, 3, ...,  $N$ .

**Subtask 3 (10 points).** Servers  $A$  and  $B$  are directly connected to each other if and only if  $|A - B| = 1$ .

**Subtask 4 (20 points).** Servers  $A$  and  $B$ , with  $A < B$ , are directly connected to each other if and only if  $2A = B$  or  $2A + 1 = B$ .

**Subtask 5 (25 points).** Any server is directly connected to at most 5 other servers.

**Subtask 6 (35 points).** No further constraints.

Moreover, the following holds: In each subtask you receive 50% of the points awarded for the respective subtask if you solve all testcases in which you never query how many servers store a certain data chunk (i.e. no input line starts with C). Inside CMS this is shown as “Group 1” of the corresponding subtask.

### Samples

Input	Output
6 9	no
S 1 2	yes
S 1 3	no
S 3 4	6
Q 5 1	6
S 4 5	5
S 1 6	3
Q 5 1	2
Q 1 5	2
C 1	
C 2	
C 3	
C 4	
C 5	
C 6	
4 4	yes
S 1 2	yes
S 1 3	no
S 3 4	no
Q 2 1	
Q 2 2	
Q 2 3	
Q 2 4	



**BOI 2021**  
Lübeck, Germany (online only)  
April 23–26, 2021

**Day 1**  
Task: **servers**  
Language: **en**

---

### Limits

Time: 2 s

Memory: 512 MiB