Problem C. ngfam the Navigator

Input file:	`standard input`
Output file:	`standard output`
Time limit:	4 seconds
Memory limit:	256 megabytes

This is an interactive problem

After purchasing tables and monitors, *TrungNotChung* needs to bring them to the location of the VNOI Cup. Ha Long City, where the VNOI Cup takes place, has n intersections numbered from 1 to n, and there are $n-1$ roads. From one intersection, it is possible to reach any other intersection using some roads in the city. Thus, the roads in Ha Long City form a *tree* structure.

Currently, *TrungNotChung* is standing at intersection 1. Unfortunately, *TrungNotChung*'s phone cannot receive any internet signal, so *TrungNotChung* cannot use the GPS feature. *TrungNotChung* does not know what the map of Ha Long city looks like, but only remembers that the VNOI Cup is held at Ha Long High School for Gifted Students – a location with a prime position as a **centroid** of the city.

Not knowing the destination, *TrungNotChung* immediately calls *ngfam* and asks for directions. *ngfam* also does not know the place where *TrungNotChung* is at looks like, so *ngfam* does not know how to give straight directions to Ha Long High School for Gifted Students. But after some thinking, the two of them agreed to communicate with each other according to the following description.

Let the position where *TrungNotChung* is standing be r, initially $r = 1$. Considering r as the root of the tree, *TrungNotChung* will use the following commands to ask *ngfam*.

Command		Explanation
`adj`	v	Find an intersection adjacent to intersection r on the path from r to v.
`subtree`	v	Find the number of vertices in the subtree rooted at intersection v.
`move`	v	*TrungNotChung* moves to position v with the condition that v is a **neighbor** of r. Then assign $r \leftarrow v$.

Of course, time is precious, so *TrungNotChung* needs to move to Ha Long Specialized High School as soon as possible to install the equipment. With only `15 555` commands, help *TrungNotChung* find a **centroid** of the city.

The city may have multiple centroids, but *TrungNotChung* only needs to find one centroid. When *TrungNotChung* arrives at a centroid that is not the competition venue, at that time *ngfam* also knows where *TrungNotChung* is and will go to pick him up.

An intersection is called a **centroid** of a city with a tree structure consisting of n vertices if, when removing this intersection (and the edges connected to this intersection) from the city, the resulting connected components all have a number of vertices not exceeding $\frac{n}{2}$.

Interaction Protocol

In this problem, the jury program will play the role of *ngfam* – executing the commands of *TrungNotChung*, and your program will play the role of *TrungNotChung* – giving commands to find the centroid of the city.

First, you need to read an integer n ($1 \le n \le 10\,000$) - the number of vertices in the tree.

Next, you need to interact with the jury program using the operations mentioned in the problem statement as follows:

Command	Interaction	
`adj`	Output	`adj` v $(1 \le v \le n)$
	Input	• Integer t when $v \neq r$ – the vertex adjacent to vertex r on the path from r to v. • -1 when $v = r$. You should terminate the program immediately in this case.
`subtree`	Output	`subtree` v $(1 \le v \le n)$
	Input	Integer s - the number of vertices in the subtree rooted at v.
`move`	Output	`move` v $(1 \le v \le n)$
	Input	• 1 when v is adjacent to r. Then r will be assigned to v. • -1 when v is not adjacent to r. You should terminate the program immediately in this case.

When you are certain that vertex r is a centroid of the city, output `found` and terminate the program. This command will not count towards the given operation limit.

If the tree has multiple centroids, move r to any centroid.

The tree is **fixed** before the interaction process and will **not change** during the interaction.

After outputting a command, remember to go to a new line and flush the standard output, otherwise you may receive a `Time limit exceeded` verdict. To do this, you can use:

- `fflush(stdout)` or `cout.flush()` in C++;
- `System.out.flush()` in Java;
- `flush(output)` in Pascal;
- `stdout.flush()` in Python;
- consult the documentation for other languages.

Scoring

Subtask	Score	Constraints
1	500	$n \le 100$
2	750	$n \le 1000$
3	1000	No additional constraints
Total	2250	

Examples

standard input	standard output
2 1	move 2 found
3 1 3 1	subtree 2 adj 2 move 3 found

Note

In the first example, the city of Ha Long consists of only 2 intersections connected to each other. Both of these intersections are centroid of the city. Therefore, besides moving to intersection 2, it is possible to immediately respond to the jury while being at intersection 1.

In the second example, the city consists of 3 intersections and 2 connecting roads 1–3 and 2–3.

- With the first query, we can conclude that 2 is a leaf intersction.
- With the second query, we can conclude that intersection 1 is not adjacent to intersection 2.

From here, it can be concluded that 3 is the centroid of the city.