

There is a maze in a massive dune. In the maze, there are  $n$  ( $2 \leq n \leq 25$ ) caves and  $m$  ( $1 \leq m \leq 300$ ) bidirectional corridors connecting them. You can drop a robot into the maze (through a small hole from above), move it around and collect information about the maze.

When the robot is inside a cave, you can see how many corridors are connecting it. But it's too dark to distinguish two caves with the same number of corridors connecting it. Further more, the corridors are too dark, so the robot cannot know which corridors it is walking along.

The only thing that can help you is a token. Initially, the token is in the robot's pocket. You can ask the robot to leave the token in a cave, or to pick it up with it. Note that you cannot leave the token in a corridor because corridors are too dark.

Your task is to find out the number of caves and corridors. Note that you have no way to take out the robot from the cave. Don't worry, it's just a robot :)

It is guaranteed that the maze is connected, no two corridors are connecting the same pair of caves, and no corridor is connecting a cave and itself. Note that the corridors can be twisted in 3D, so the maze is not necessarily a planar graph.

## Interaction Protocol

Your program should read from standard input, and write to standard output. After printing each line to the standard output, you should flush the output, by calling `fflush(stdout)` or `cout << flush` in C/C++, `flush(output)` in Pascal and `System.out.flush()` in Java. Please read general instructions for interactive problems for more information.

First, read the number of test cases  $T$  ( $1 \leq T \leq 25$ ). For each test case, issue one or more 'Look', 'Walk', 'Put' and 'Take' commands, then one 'Answer' command.

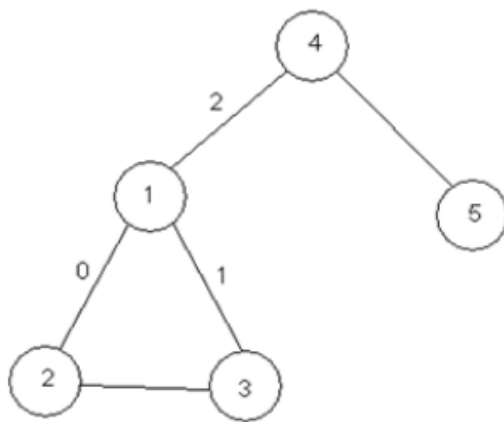
Command	Description
Look	Returns $d$ and $s$ , where $d$ is the number of corridors connecting the current cave, and $s = 0$ means there is no token in the cave, $s = 1$ means there is a token.
Walk $i$	Walk along the corridors numbered $i$ . Initially the corridors connecting the starting cave are counter-clockwise numbered 0, 1, 2, ..., starting from an arbitrary corridor. Later, the corridors connecting the current cave are always counter-clockwise numbered, starting from the corridor that you used to enter current cave. That means, 'Walk 0' always means 'go back'. This command does not return anything.
Put	Put the token in current cave. Please make sure the token is in robot's pocket. This command does not return anything.
Take	Take the token from the current cave. Please make sure the token is on the ground of the current cave. This command does not return anything.
Answer $n$ $m$	You found out the maze contains $n$ caves and $m$ corridors. This command does not return anything.

If your program violated any of these rules (bad format, invalid arguments etc), the server will exit immediately, and you will receive **Protocol Violation (PV)**.

## Protocol Limit

For each test case, you can issue at most 65536 'Walk' commands, **otherwise you'll get Protocol Limit Exceeded (PLE)**.

**Sample Explanation:** The maze graph is



Initially, you're at node 1, corridor 0, 1, 2 lead to cave 2, 3, 4, respectively.

## Sample Interaction

```

1
    Look
3 0
    Put
    Walk 0
    Look
2 0
    Walk 1
    Look
2 0
    Walk 1
    Look
3 1
    Take
    Walk 1
    Look
2 0
    Walk 1
    Look
1 0
    Answer 5 5
  
```