

Programming Lab: Aliens

RAM Memory: 4 MiB

Time: 0.5 seconds

Input: aliens.in

Output: aliens.out

Description

In 2020 humans sent an autonomous robot *Perseverance* to Mars. (*"The Mars 2020 mission is delivering the Perseverance rover to the Red Planet as part of NASA's Mars Exploration Program. The program's ongoing series of missions is helping us answer key questions about the potential for life on Mars. While previous missions have helped us look for signs of habitable conditions in ancient times, Perseverance will take it one step further by searching for signs of past microbial life itself."*)

Soon *Perseverance* noticed various signs of life on Mars. On their images there were green creatures who could change their location. After trying to hide for some time, the green creatures (subsequently called *aliens*) decided to communicate with our civilization. The humans on Earth (called *earthlings*) found that they are not dangerous, and decided to maintain good relations with aliens and get access to their technologies.

To address culture differences and avoid misunderstandings, earthlings became interested in the everyday life of aliens. As they found out, aliens have asexual reproduction: every alien can have no more than two children. Every child is either left-handed or right-handed. Additionally, if an alien has two children, then one of them is necessarily left-handed, and the other one is right-handed. Reproduction can happen soon after birth, many generations can live simultaneously. Since aliens have very large families, their most favoured relationships are with no more than 2 of their relatives.

Earthlings studied the question: How to identify the two (or less) favorite relatives for every alien. First, we pick some alien without living parent. Then draw the family tree (the tree of parent-child relationships with the given alien as the root), then every alien in this family tree has two favorite relatives immediately preceding him and immediately following him in the *inorder* DFS traversal of that tree. Left-handed aliens are shown to the left of their parent, the right-handed ones - to the right.

Your task is to build an efficient program that receives the genealogy data for a single family tree, and it should find the two favorite relative aliens for any given alien upon receiving a query.

It is known that a single extended family has no more than 10^5 aliens that are currently alive. Every alien has a unique number $[1..10^5]$.

An efficient solution for an input having just up to 100 aliens would get about 8 points. If it handles up to 1000 aliens, then it gets about 9 points.

Input:

The first line contains the number for the alien that is the *main living ancestor* of the given family tree. All the remaining lines of the input file contains one of these four commands that build the genealogy tree:

1. Specify the left-handed child for a parent.
2. Specify the right-handed child for a parent.
3. Query for the favorite relatives of a given alien.
4. Finish your work.

```

Ancestor
...
L Parent Child
...
R Parent Child
...
? Alien
...
F

```

- **Ancestor** is the number for an alien that is the *main living ancestor* of the whole family tree.
- **L** is the command to specify the left-handed child for a parent.
- **R** is the command to specify the right-handed child for a parent.
- **Parent** is the number for an alien that will get a new child (the identifier for an alien is within these boundaries: $[1..10^6)$).
- **Child** is the child to the parent **Parent**, (the identifier is in the interval $[1..10^6)$).
- **Alien** is the alien identifier in the request to find the favorite relatives (the identifier is in the interval $[1..10^6)$).

If the commands **L** or **R** succeed, new nodes are appended to the genealogy tree.

The input data is valid - regarding the format and the limitations defined above.

Output:

Depending on the input file the output file contains output for every query command and also every command that ends in failure:

```

Prev Next
...
error0
...
error1
...
error2
...
error3
...
error4
...
error5

```

- **Prev** is the immediately preceding alien (in *inorder* traversal order) for the given alien **Alien**, when we run the command **? Alien**. If there is no previous alien, output 0.
- **Next** is the immediately following alien (in *inorder* traversal order) for the given alien **Alien**, when we run the command **? Alien**. If there is no next alien, output 0.
- **error0** - alien with identification number **Alien** does not exist in the family tree, when we run the command **? Alien**.
- **error1** - **Parent** and **Child** are the same.
- **error2** - **Parent** does not exist in the family tree.
- **error3** - **Child** is already used in the family tree.

- error4 - Parent already has a left-handed child in the family tree.
- error5 - Parent already has a right-handed child in the family tree.

If a command causes multiple errors at once, print the one with the smallest number.

Example:

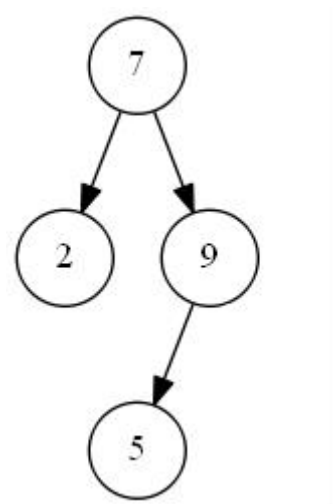
An input file `aliens.in`:

```
7
L 7 2
R 7 9
L 9 5
? 2
R 5 6
L 2 4
L 5 1
? 5
? 8
L 1 4
F
```

An output file `aliens.out`:

```
0 7
1 6
error0
error3
```

The family tree right before the command `? 2`



The family tree right before the command `? 5`

