A. AVL Tree

time limit per test: 2 seconds

memory limit per test: 256 megabytes

You are given an initially empty set. Your task is to implement an **AVL tree** that supports the following two operations:

- **ADD k v** — add key k with value v to the BST OR print `KEY ALREADY EXISTS`
- **LOOKUP k** — print the value v associated with the key k in the BST OR print `KEY NOT FOUND`
- **DELETE k** — delete an item with key k from the BST OR print `KEY NOT FOUND`
- **PRINT_ROTATIONS** — print the total number of rotations

After processing each query, dependent to the type of the query either print the associated output or move to the next query.

Input

The first line contains a single integer N ($1 \leq N \leq 10^5$) — the number of queries.

Each of the next N lines contains a query in the format:

`OPERATION` M_i

where `OPERATION` is one of the operations mentioned above , and M_i ($-100 \leq M_i \leq 100$) is the set of argument(s) for the query.

Output

Print the output of each command separated by a new line.

Example

input Скопировать

```
10
ADD 7 5
ADD 3 2
ADD 10 100
ADD 2 3
ADD 6 6
ADD 4 4
ADD 8 7
DELETE 7
PRINT_ROTATIONS
LOOKUP 8
```

output Скопировать

```
4
7
```