SCPE Spring 2025

CycleGuard

Jason Feng

CycleGuard's co-op system allows users to join **packs**, teams of users. When a user finishes a bike ride, the system records the user's distance traveled. Each pack features metrics such as distance and time traveled.

You must design a system for fast computation of **distance traveled** by all users in a pack. This value only takes into account the distance traveled by users **after joining the pack**. That is to say, a user joins the pack with 0 distance traveled, and this value increases every time that user completes a ride. If a user leaves and rejoins, the distance is reset to 0.

Objective

You are given a list of u **updates**, one per line. Each query starts with one letter, which determines what action to take. Usernames are strings of 1 to 10 lowercase letters.

- Queries starting with `j` represent a user **j**oining the pack. These queries take the form `j username`. The user is guaranteed to not be in the pack.
- Queries starting with `r` represent a user finishing a **r**ide. These queries take the form `r username distance`, where `distance` is the total distance traveled by that user $(0 \leq \text{distance} \leq 10^4)$. The user is guaranteed to already be in the pack.
- Queries starting with `l` represent a user **l**eaving the pack. These queries take the form `l username`. The user is guaranteed to already be in the pack.

For each of the q updates, output the distance traveled by all users in the pack, on a separate line.

Input Specification

The first line of input is integer u: the number of updates.
Following this are u updates following one of the three above formats.

Constraints

- $1 \leq u \leq 10^5$

Output Specification

For each of the q updates, output the distance traveled by all users in the pack, on a separate line.

Sample Input	Sample Output
7 j sskota j braxa r sskota 5 r braxa 2 l sskota r braxa 4 l braxa	0 0 5 7 2 6 0
7 j jhfeng r jhfeng 4 r jhfeng 2 j ahhoang l ahhoang j ahhoang r ahhoang 4 l jhfeng	0 4 6 6 6 10 4