Guess the Queue

Guess the Queue

Busland is a city where the only means of transportation are buses. To get on a bus in Busland, people have to wait in queue. Each person has a **unique** bus **ID**. The person at the front is considered the first person and the person at the back is considered the last person in the queue. Trivially, the first person will get on the bus first and the last person will get on last.

It is well known that you should enter at the back of the queue. However, some people are very late and even though it is considered very rude, they try to enter in an alternative way. Since the sides of the queue are barricaded, the only alternative way is to enter from the front of the queue. Sometimes there are also people at the back of the queue, who become tired of waiting. They exit the queue from the back and just start walking to their destination instead.

Your task is to deal with three types of operations as follows:

1. '**1 *x y***' The person with **ID *y*** enters from the back or front, if ***x*** is '**B**' or '**F**' respectively.

2. '**2 *x'*** The person in the back or front exits the queue, if ***x*** is '**B**' or '**F**' respectively.

3. '**3 *x y'*** Find the **ID** of the person in the ***y***th position of the queue or the position of the queue, where the person with **ID *y*** is currently situated, if ***x*** is '**D**' or '**P**' respectively.

Input:

The first line of input will contain the number of test cases, ***T* (1 <= *T* <= 5)**. Then ***T*** cases follow.

Each case starts with an integer ***N***, denoting the total number of operations. The following ***N*** lines will contain one of the three types described previously. The operations are in chronological order.

It is guaranteed that the given input is always valid. Thus, for the second type operation, the queue will always be non-empty and for the third type, the given position or **ID** will always exist in the queue. Also, a person who has already exited, will not enter the queue again.

Constraints:

For the **easy** version, **1 <= *N* <= 2000**

For the **hard** version, **1 <= *N* <= 200000**

In general,

1 <= each ID <= 10^9, IDs are **unique** for each person

1 <= each position <= current size of the queue

Output:

For each case, print the case number on a single line, in the format "**Case x:**", where ***x*** is the case number.

For each operation of the third type, output a single integer, which denotes the answer of the query.

Sample Input	Sample Output
1 7 1 B 1 1 B 2 1 F 3 3 D 3 2 B 1 F 4 3 D 2	Case 1: 2 3

Explanation of the Sample:

There is only 1 test case, in which we have to perform a total of 7 operations.

At first, the queue is empty. A person with **ID 1** and **2** enters from the back. After that, a person with **ID 3** enters from the front. Now there are **3** people in the queue and the **ID** of the **3rd** person in the queue is **2**.

The person in the back (who has **ID 2**) exits the queue. This leaves only two people remaining.

Finally, a person with **ID 4** enters from the front. Now there are again **3** people in the queue and the **ID** of the **2nd** person of the queue is **3**.

Problem Setter: Aninda Majumder