# J&J's

Lea and Bea have arranged for some friends to come over for a big LAN party lasting the whole weekend. Since healthy food is not appropriate for LAN parties, they bought tons of pizza, lasagna, and of course even more snacks like gummybears, chocolate cookies, and J&J's.

Unfortunately the moment they wanted to start their gaming session, the switch decided to retire, leaving them with no working local network to play on. After some hours of trying to get the switch to work again, they gave up and decided to play the infamous no-LAN-tonight-game called "Toss dem J&J's". The game works as follows: One person takes a few J&J's and tosses them towards a bunch of glasses that are standing in a line. The J&J's have to be tossed in such a way that every glass that gets some new J&J's gets the same number of new J&J's. Also if two glasses each get new J&J's all glasses in between also need to get new J&J's. For example, if some player messes up and tosses two J&J's into one glass and only one J&J into the other one, he/she is forced to remove the J&J's he/she tossed from the glasses. Obviously he/she loses the game instantly.

If the player did not violate any of the previous rules, he scores points equal to the total number of J&J's in all glasses he hit. Therefore it is crucial to know how many J&J's are in a given interval of glasses. Can you help Lea keep track of the number of J&J's in the glasses during the game?

## Input

The first line of the input contains an integer $t$. $t$ test cases follow, each of them separated by a blank line.

Each test case starts with two integers $n$ $k$, where $n$ is the number of glasses available and $k$ is the number of queries to follow. $k$ more lines follow. Each is of one of the following forms:

- `q` $a$: Return the number of J&J's in glass $a$.
- `i` $l$ $r$ $v$: Add $v$ J&J's to each glass in the interval $l$ to $r$ (including $l$ and $r$).

## Output

For each test case output one line containing "Case #$i$: $x$ where $i$ is its number, starting at 1, $x = (r_1 + \cdots + r_j) \mod 1000000007$" and $r_1 \ldots r_j$ are the answers to the glass queries.

## Constraints

- $1 \leq t \leq 5$
- $1 \leq k \leq 10^5$
- $1 \leq n \leq 10^6$
- $1 \leq v \leq 1000$
- $1 \leq a \leq n$
- $1 \leq l \leq r \leq n$
- There is always at least 1 sum-query

**Sample Input 1**

```
3
4 3
i 1 3 1
q 3
q 1

6 5
i 2 5 2
i 1 6 3
q 3
i 1 3 1
q 1

5 2
i 1 5 4
q 2
```

**Sample Output 1**

```
Case #1: 2
Case #2: 9
Case #3: 4
```

**Sample Input 2**

```
5
4 7
i 1 2 645
i 2 4 945
q 3
q 2
q 1
i 1 3 606
q 1

5 6
i 2 5 695
i 3 5 863
q 3
q 2
q 3
q 4

8 7
i 3 6 488
i 4 6 234
q 1
i 5 8 179
q 3
i 4 8 930
q 8

4 3
i 2 3 519
q 2
q 2

4 8
q 2
i 1 3 664
i 2 3 545
i 2 4 724
q 4
i 1 3 305
q 2
q 2
```

**Sample Output 2**

```
Case #1: 4431
Case #2: 5369
Case #3: 1597
Case #4: 1038
Case #5: 5200
```