# Name Game

## Objective

Give practice with Tries in C.

## Story

The employees have decided to kill some time by creating some unique backstories for some of the critters in your park. Partly due to their solitary nature there is not a more rich, diverse, and mysterious set of characters than that of your orangutans. To help distinguish between the different orangutans each one gets a name. Due to the escapades that the orangutans get into their names can change from day to day. This dynamic element makes keeping track of the storylines tricky. The employees like to feed their favorite orangutans mangos as a treat.

Due to certain great ape weight issues, you are concerned that your employees may be over feeding some orangutans and underfeeding others. Luckily, your employees track the orangutans that get fed. Unluckily, your employees write the name of the orangutan at the time of the feeding.

You have decided to embrace the nonsense that is an inconsistent and changing naming convention, but you have forced your employees to write down when an orangutan changes its name. You will focus on asking at particular times how many mangos have been eaten by the orangutans whose name begins with a particular letter sequence.

## Problem

Given a list of name changes and feedings, determine the number of mangos eaten by the orangutans with a particular name prefix.

## Input

Input will begin with a line containing 2 integers, ***n*** and ***e*** ($1 \leq n \leq 500{,}000$; $1 \leq e \leq 500{,}000$), representing the number of orangutans and the number of events. The following ***e*** lines each contain a single event description. An event description will be one of the following three,

- 1 ***n a*** - which represents a feeding, where ***n*** is the name of the orangutan at the time of the feeding and ***a*** represents the amount of mangos given to the orangutan. ($1 \leq a \leq 100$)
- 2 ***o n*** - which represents a name change, where ***o*** represents the old name of the orangutan and ***n*** represents the new name.
- 3 ***p*** - which represents an inquiry as to the number of mangos eaten by orangutans whose name starts with the sequence of characters ***p***.

Each name and character sequence will contain at most 20 characters. All names will be strictly uppercase Latin characters (‘A’ through ‘Z’). No name will contain whitespace. No orangutan will change their name to an already existing name.

Assume that no orangutan has eaten prior to the given events

## Output

For each input event that represents an inquiry print the number of mangos eaten by orangutans whose name starts with the given input character sequence.

| Sample Input | Sample Output |
|---|---|
| 10 6<br>1 BOB 5<br>1 BETTY 3<br>3 B<br>3 ALICE<br>2 BETTY ALICE<br>3 B | 8<br>0<br>5 |
| 4 8<br>1 WILLIAM 4<br>1 WILL 6<br>3 WILLI<br>1 WILLIAN 9<br>1 WILLY 10<br>2 WILL MATT<br>1 WILLIAN 2<br>3 WILL | 4<br>25 |

## Explanation

**Case 1**

There are 10 orangutans, but we only ever hear about 2 of them (initially BOB and BETTY). Additionally there is

First Event

BOB gets fed 5 mangos.

Second Event

BETTY gets fed 3 mangos.

Third Event

We want to know how many mangos have been given to orangutans whose name starts with B. Both BOB who received 5 mangos and BETTY who received 3 mangos have a name that starts with the letter B. Because of this the total mangos given is 8.

Fourth Event

We want to know how many mangos have been given to orangutans whose name starts with ALICE. We do not know of any orangutans with names that start with ALICE that have been given a mango. For this reason the answer is 0.

Fifth Event

BETTY is changing their name to ALICE. This means that “BETTY” has eaten 0 mangos and ALICE has eaten 3.

Sixth Event

We want to know how many mangos have been given to orangutans whose name starts with B. BOB who received 5 mangos has a name that starts with the letter B, but since BETTY is no longer BETTY we don’t count them. Because of this the total mangos given is 5.

**Case 2**

There are 4 orangutans and 8 events.

First Event

WILLIAM gets 4 mangos.

| Name | WILLIAM | ??? | ??? | ??? |
|---|---|---|---|---|
| Mangos | 4 | 0 | 0 | 0 |

Second Event

WILL gets 6 mangos

| Name | WILLIAM | WILL | ??? | ??? |
|---|---|---|---|---|
| Mangos | 4 | 6 | 0 | 0 |

Third Event

We want to know how many mangos orangutans whose names start with WILLI got. In this case only WILLIAM meets the criteria. They got 4 mangos.

Fourth Event

We feed WILLIAN 9 mangos.

| Name | WILLIAM | WILL | WILLIAN | ??? |
|---|---|---|---|---|
| Mangos | 4 | 6 | 9 | 0 |

Fifth Event

We feed WILLY 10 mangos.

| Name | WILLIAM | WILL | WILLIAN | WILLY |
|---|---|---|---|---|
| Mangos | 4 | 6 | 9 | 10 |

Sixth Event

WILL changes their name to MATT

| Name | WILLIAM | MATT | WILLIAN | WILLY |
|---|---|---|---|---|
| Mangos | 4 | 6 | 11 | 10 |

Seventh Event

WILLIAN gets 2 more mangos.

| Name | WILLIAM | MATT | WILLIAN | WILLY |
|---|---|---|---|---|
| Mangos | 4 | 6 | 11 | 10 |

Eighth Event

We want to know how many mangos have been eaten by orangutans whose name starts with WILL. The valid orangutans are WILLIAM, WILLIAN, and WILLY.

The total mangos eaten is 4+ 11 +10 = 25.

## Hints

**Graph:** Use a Trie. The Trie should index based on the name of the orangutan.

**Recommended Function:** The functionality of the Trie will help if the following is implemented

- Compute the sum for a subtree
- Increment a node's value, if it exists / create a node if it does not.

**Node Data:** I recommend also storing each trie Node the sum of all descendants in the subtree. Storing this value makes computing the output for event type 3 *fast*. When you update a node you need to update all the ancestors nodes if you store the subtree sum.

**Struct Definition Recommendation:** I used the following struct

```
struct TrieNode {
   TrieNode * children[26];
   int subtreeSum;
   int nodeTotal;
};
```

**Name Change:** Make sure when an orangutan changes its name you remove its sum from the old subtree. You may want to consider doing this by adding negative its original value to the old name and positive its original value to the new name.

**AVL:** An AVL could be used, but it is very challenging to get the AVL to work 🙁.

## Grading Criteria

- Good comments, whitespace, and variable names
  - 15 points
- No extra input output (e.g. input prompts, “Please enter the number of friends:”)
  - 10 points
- Change how many tokens you read in based on the event type
  - 5 points
- If using a Trie, offset the letters by ‘A’ to map to an index in the range of 0 to 25 inclusive.
  - 5 points
- Insert only names you read.
  - 5 points
- “Remove” the old value when an orangutan changes its name.
  - 10 points
- Programs will be tested on 10 cases
  - 5 points each

*No points will be awarded to programs that do not compile using “gcc -std=gnu11 -lm”.*

*Sometimes a requested technique will be given, and solutions without the requested technique will have their maximum points total reduced. For this problem use a trie.* ***Without this programs will earn at most 50 points!***

*Any case that causes a program to return a non-zero return code will be treated as wrong. Additionally, any case that takes longer than the maximum allowed time (the max of {5 times my solutions time, 10 seconds}) will also be treated as wrong.*

***No partial credit will be awarded for an incorrect case.***