

Reserved Seats (seats)

When commuting by train, Italian people tend to just sit in the first empty seat they find instead of spending time looking for their reserved seat. Specifically: when an Italian person boards a train, he/she will start looking for an empty seat (from the very first seat in the train) and they will stop only:

- When an empty seat is encountered: in this case, that person will occupy the seat. Or...
- When the actual seat, reserved to that person, is encountered: in this case, if the seat is occupied, the person will kindly ask the offending traveler to move. The offending traveler will apologize, get up, and go to their reserved seat (*not* just some empty seat, this time).

Note that, in the second case presented, a *chain reaction* of movings could fire up (A asks B to move, then B asks C to move, and so on).



Figure 1: A *Frecciarossa* train in the *Milano Centrale* station.

Train seats are numbered from 0 to $N - 1$. Every person who boards the train will start looking for an empty seat from the 0-th seat (the “head” of the train), and then walk up to the $(N - 1)$ -th seat (the “tail” of the train).

Giorgio loves travelling by train. Today, sitting in his reserved seat, he has seen a lot of travelers boarding/leaving, moving around the train, asking other travelers to get up. Since Giorgio hates when his reserved place is occupied, he wanted to measure how bad the Italian situation really is and started wondering: how many people, in total, had to get up and apologize to other travelers?

You have the list of “events” that Giorgio witnessed (from the train departure to its arrival), which can be either “a person, for which the k -th seat was reserved, *boards* the train” or “a person, for which the k -th seat was reserved, *leaves* the train”. Compute how many times a person had to get up and give their seat to another passenger.

📎 Among the attachments of this task you may find a template file `seats.*` with a sample incomplete implementation.



Input

The first line contains two integers N and Q , the number of seats in the train and the number of events, respectively.

Other Q lines follow, each containing a character T_i representing the type of event ('b' for "boards the train", 'l' for "leaves the train"), and an integer K_i representing the seat *reserved to the person* in this event (not necessarily the seat they actually sat on).

Output

You need to write a single line with an integer: how many times someone had to apologize and get up.

Constraints

- $1 \leq N, Q \leq 100\,000$.
- T_i is either 'b' or 'l' for each $i = 0 \dots Q - 1$.
- $0 \leq K_i \leq N - 1$ for each $i = 0 \dots Q - 1$.
- The events are consistent (e.g. the same person won't board the train twice, and it won't happen that a non-existent person leaves the train).
- After a person leaves the train, a new person with the same reserved seat could board the train.

Scoring

Your program will be tested against several test cases grouped in subtasks. In order to obtain the score of a subtask, your program needs to correctly solve all of its test cases.

- **Subtask 1** [5 points]: Examples.
- **Subtask 2** [25 points]: $N, Q \leq 10$.
- **Subtask 3** [40 points]: $N, Q \leq 1000$.
- **Subtask 4** [30 points]: No additional limitations.

Examples

input.txt	output.txt
3 3 b 1 b 2 b 0	2
4 6 b 1 b 2 b 0 l 0 b 3 b 0	3

Explanation

In the **first sample case** three people board the train in the following manner:

```
A >>   ___ ==> A__
B >>   A__ ==> AB_
C >>   AB_ ==> CB_ (A)
A ->   CB_ ==> CA_ (B)
B ->   CA_ ==> CAB
```

In the **second sample case**, five people board the the train and one leaves it:

```
A >>   ____ ==> A____
B >>   A____ ==> AB____
C ->   AB____ ==> CB____ (A)
A ->   CB____ ==> CA____ (B)
B ->   CA____ ==> CAB____
        CAB____ ==> _AB_   ~> C
D >>   _AB_ ==> DAB_
E ->   DAB_ ==> EAB_ (D)
D ->   EAB_ ==> EABD
```