

Arena Freaks (arena)

William is a long-time fanatic of *OpenArena*, an open source FPS (First-Person Shooter) game. For this reason he decided to start a brand-new web portal gathering OpenArena addicts from all over the world: arenafreaks.net. This website addresses multiple different needs of a professional OA player through forums, dedicated social networks, tutorials, rankings, and much more. The feature which William is most proud of, however, is the game database: a huge collection of the most spectacular games ever played in OpenArena, transcribed in *arenaic notation* (an alphanumeric notation subtly inspired by the algebraic notation of chess games).



Figure 1: A typical OpenArena “capture the flag” match.

In this notation, a game between N players (numbered from 0 to $N - 1$) each starting with L lives is transcribed as a series of E events which can be (among many others) of the following types:

- *frags* (denoted as ‘ $P \text{ f } Q$ ’), where a player P “kills” a player Q thus reducing the number of Q ’s lives by 1;
- *explosions* (denoted as ‘ $P \text{ e}$ ’), where a player P frags simultaneously all the other players, thus reducing the number of their lives by 1.

Help William count the number of players still alive (with some lives left) at the end of the game!

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

Input

The input file contains the full transcription of a game in arenaic notation. The first line contains the three integers N , E , L . Other E line follow, each containing the transcription of an event (as described in the task description).

Output







You need to write a single line with an integer: the number of players with some lives left at the end of the game.

Constraints

- $1 \leq N, E, L \leq 100\,000$.
- $0 \leq P, Q \leq N - 1$ in each event.
- All events are valid, that is, players P and Q in both frags and explosions cannot have already lost all their lives.

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 (0 points) Examples

- Subtask 2 (20 points) $N \leq 5$ and there are no explosions.

- Subtask 3 (20 points) There are no explosions.

- Subtask 4 (20 points) There are no frags.

- Subtask 5 (20 points) $N, E \leq 100$.

- Subtask 6 (20 points) No additional limitations.


Examples

input.txt	output.txt
4 4 2 0 f 3 1 e 2 f 1 0 f 2	2
6 7 4 5 e 5 f 1 1 f 2 0 f 2 3 f 2 5 e 3 e	4

Explanation

In the **first sample case**, the number of player's lives changes as follows:

Player	0	1	2	3
	2	2	2	2
0 f 3	2	2	2	1
1 e	1	2	1	×
2 f 1	1	1	1	
0 f 2	1	1	×	

In the **second sample case**, the number of player's lives changes as follows:

Player	0	1	2	3	4	5
	4	4	4	4	4	4
5 e	3	3	3	3	3	4
5 f 1	3	2	3	3	3	4
1 f 2	3	2	2	3	3	4
0 f 2	3	2	1	3	3	4
3 f 2	3	2	×	3	3	4
5 e	2	1		2	2	4
3 e	1	×		2	1	3