The University administration plans to build a new dinner hall, to replace the several small (and rather inadequate) dinner halls spread over the campus. To estimate the number of places needed in the new dinner hall, they performed an experiment to measure the maximum total number of clients inside the existing dinner halls at any time. They hired several students as *pollers*, and positioned one poller at each entrance and each exit of the existing dinner halls. The pollers' task was to note in small cards the time each client entered or exited the hall (one card for each event). In each card they wrote the time, in the format HH:MM:SS, and the associated event (letter '`E`' for an entry, letter '`X`' for and exit).

The experiment started in the morning, before breakfast, and ended in the evening, after dinner. The pollers had their watches synchronized, and the halls were empty both before and after the experiment (that is, no client was inside a hall before the experiment began, and no client remained in a hall after the experiment ended). The pollers wrote exactly one card for every client who entered a hall and for every client who exited a hall.

After the experiment, the cards were collected and sent to the administration for processing. The task, however, was not as easy as planned, because two problems were detected. Firstly, the cards were bunched together in no particular order and therefore needed sorting; that is fairly easy, but time-consuming to do by hand. But what is worse is that, although all cards had a valid time, some pollers forgot to write the letter specifying the event. The University administration decided they needed help from an expert!

Given a set of cards with times and the indication of the event (the indication of the event may be missing), write a program to determine the maximum number of clients that could possibly had been inside the dinner halls in a given instant of time.

Input

The input contains several test cases. The first line of a test case contains one integer N indicating the number of cards collected in the experiment ($2 \leq N \leq 64800$). Each of the next N lines contains the information written in a card, consisting of a *time specification*, followed by a single space, followed by an *event specification*. A time specification has the format $HH : MM : SS$, where HH represents hours ($06 \leq HH \leq 23$), MM represents minutes ($00 \leq MM \leq 59$) and SS represents seconds ($00 \leq SS \leq 59$). Within a test case, no two cards have the same time. An event specification is a single character: uppercase '`E`' for entry, uppercase '`X`' for exit and '`?`' for unknown. Information may be missing, but the information given is always correct. That is, the times noted in all cards are valid. Also, if a card describes an entry, then a client did enter a hall at the informed time; if a card describes an exit, then a client did leave a hall at the informed time; and if a card describes an unknown event, then a client did enter or leave a hall at the informed time.

The last test case is followed by a line containing a single zero.

Output

For each test case in the input, your program must print a single line, containing one single integer, the maximum total number of clients that could have been inside the dinner halls in a given instant of time.

Sample Input

```
4
07:22:03 X
07:13:22 E
08:30:51 E
21:59:02 X
4
09:00:00 E
20:00:01 X
09:05:00 ?
20:00:00 ?
8
10:21:00 E
10:25:00 X
10:23:00 E
10:24:00 X
10:26:00 X
10:27:00 ?
10:22:00 ?
10:20:00 ?
0
```

Sample Output

```
1
2
4
```