D. Logging

The main server of Gomble company received a log of one top-secret process, the name of which can't be revealed. The log was written in the following format: «`[date:time]: message`», where for each «`[date:time]`» value existed not more than 10 lines. All the files were encoded in a very complicated manner, and only one programmer — Alex — managed to decode them. The code was so complicated that Alex needed four weeks to decode it. Right after the decoding process was finished, all the files were deleted. But after the files deletion, Alex noticed that he saved the recordings in format «`[time]: message`». So, information about the dates was lost. However, as the lines were added into the log in chronological order, it's not difficult to say if the recordings could appear during one day or not. It is possible also to find the minimum amount of days during which the log was written.

So, to make up for his mistake Alex has to find the minimum amount of days covered by the log. Note that Alex doesn't have to find the minimum amount of days between the beginning and the end of the logging, he has to find the minimum amount of dates in which records could be done. (See Sample test 2 for further clarifications).

We should remind you that the process made not more than 10 recordings in a minute. Consider that a midnight belongs to coming day.

Input

The first input line contains number n ($1 \le n \le 100$). The following n lines contain recordings in format «`[time]: message`», where time is given in format «`hh:mm x.m.`». For `hh` two-digit numbers from `01` to `12` are used, for `mm` two-digit numbers from `00` to `59` are used, and `x` is either character «`a`» or character «`p`». A message is a non-empty sequence of Latin letters and/or spaces, it doesn't start or end with a space. The length of each message doesn't exceed 20.

Output

Output one number — the minimum amount of days covered by the log.

Examples

input

```
5
[05:00 a.m.]: Server is started
[05:00 a.m.]: Rescan initialized
[01:13 p.m.]: Request processed
[01:10 p.m.]: Request processed
[11:40 p.m.]: Rescan completed
```

output

```
2
```

input

```
3
[09:00 a.m.]: User logged in
[08:00 a.m.]: User logged in
[07:00 a.m.]: User logged in
```

output

```
3
```

Note

Formally the 12-hour time format is described at:

- `http://en.wikipedia.org/wiki/12-hour_clock.`

The problem authors recommend you to look through these descriptions before you start with the problem.

- `http://en.wikipedia.org/wiki/12-hour_clock.`