O - Longest Nap

As you may already know, computer teachers have busy schedules! Your teacher, let's call him *Professor C*, is a bit lazy and wants to take a nap during the day, but he doesn't have a lot of chances to nap. He wants to take one nap every day, and it should be the longest nap possible, given his schedule. He needs a program to help him in this task but, as we said, Professor C is lazy, so he wants you must write the program.

The Input

The first line of input will indicate the number of test cases to follow. For each test case that follows, the first line contains a positive integer *s* (not greater than 100) representing the number of scheduled appointments during that day. In the next *s* lines there are the appointments in the following format:

```
time1 time2 appointment
```

Where *time1* represents the time when the appointment starts, and *time2* is the time it ends. All times are in hh:mm format. *time1* is always strictly less than *time2* and they are separated by a single space. All times are greater than or equal to 10:00 and less than or equal to 18:00. Your response must be in this interval as well (no nap can start before 10:00 and last after 18:00). The appointment can be any sequence of characters, but will always be in the same line. You can assume that no line will be longer than 255 characters, that $10 \leq hh \leq 18$ and $0 \leq mm < 60$. The input will be in no specific order.

The Output

Write to standard output. For each test case, you must print the following line:

```
Day #d: the longest nap starts at hh:mm and lasts for [H hours and] M minutes.
```

...where d stands for the number of the test case (starting from 1) and hh:mm is the time when the nap can start.

To display the duration of the nap, follow these simple rules:

1. If the total duration X in minutes is less than 60, just print "M minutes", where M = X.
2. If the total duration X in minutes is greater or equal to 60, print "H hours and M minutes", where H = X div 60 (integer division) and M = X mod 60.

Notes:

- You do not have to worry about singular versus plural, so you can print "1 minutes" or "1 hours" if the values you print are equal to 1.
- The duration of a nap is calculated by the difference between the ending time free and the begining time free. That is, if an appointment ends at 14:00 and the next one starts at 14:47, then you have (14:47) − (14:00) = 47 minutes of possible nap time.
- If there is more than one longest nap with the same duration, print the earliest one.
- You can assume that there won't be a completely busy day (i.e. you can assume that there will be at least one possible nap).

Sample Input

```
4
4
10:00 12:00 Lectures
12:00 13:00 Lunch, like always.
13:00 15:00 Boring lectures...
15:30 17:45 Reading
4
10:00 12:00 Lectures
12:00 13:00 Lunch, just lunch.
13:00 15:00 Lectures, lectures... oh, no!
16:45 17:45 Reading (to be or not to be?)
4
10:00 12:00 Lectures, as everyday.
12:00 13:00 Lunch, again!!!
13:00 15:00 Lectures, more lectures!
15:30 17:15 Reading (I love reading, but should I schedule it?)
1
12:00 13:00 I love lunch! Have you ever noticed it? :)
```

Sample Output

```
Day #1: the longest nap starts at 15:00 and lasts for 30 minutes.
Day #2: the longest nap starts at 15:00 and lasts for 1 hours and 45 minutes.
Day #3: the longest nap starts at 17:15 and lasts for 45 minutes.
Day #4: the longest nap starts at 13:00 and lasts for 5 hours and 0 minutes.
```