# CSE 241 Assignment #7 - Schedule for Nap Spring 2015

**Due date: May 29 (Friday) - 20:00**

## Description

Professors lead very busy lives with full schedules of work and appointments. Professor P likes to nap during the day, but his schedule is so busy that he doesnt have many chances to do so.

He *really* wants to take one nap every day, however. Naturally, he wants to take the longest nap possible given his schedule. Write a program to help him with the task.

## Input

The input consists of an arbitrary number of test cases, where each test case represents one day.

The first line of each case contains a positive integer $s \leq 100$, representing the number of scheduled appointments for that day. The next $s$ lines contain the appointments in the format *time1 time2 appointment*, where *time1* represents the time which the appointment starts and *time2* the time it ends. All times will be in the **hh:mm** format; the ending time will always be strictly after the starting time, and separated by a single space.

All times will be greater than or equal to 10:00 and less than or equal to 18:00. Thus your response must be in this interval as well; i.e., no nap can start before 10:00 and last after 18:00.

The appointment can be any sequence of characters, but will always be on the same line. You can assume that no line is be longer than 255 characters, that $10 \leq hh \leq 18$ and that $0 \leq mm < 60$. You *cannot* assume, however, that the input will be in any specific order, and must read the input until you reach the end of file.

## Output

For each test case, you must print the following line:

**Day #*d*: the longest nap starts at *hh*:*mm* and will last 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 length of the nap, follow these rules:

1. If the total time $X$ is less than 60 minutes, just print ”$X$ **minutes**.”
2. If the total duration $X$ is at least 60 minutes, print ”$H$ **hours and M minutes**,” where

   $$H = X \div 60 \text{ (integer division, of course) and } M = X \bmod 60.$$

   You don't have to worry about correct pluralization; i.e., you must print ”**1 minutes**” or ”**1 hours**” if that is the case.

The duration of the nap is calculated by the difference between the ending time and the beginning time. 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 napping.

If there is more than one longest nap with the same duration, print the earliest one. You can assume the professor won't be busy all day, so there is always time for at least one possible nap.

## Sample Input

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

## Submitting Your Code and Report

Your file name must be “Nap.java”, and it is the only file you should submit. Please do not submit the executable file of your program. Turn in your project using the “oopsubmit” command as follows:

```
$ oopsubmit assign7 Nap.java
```

Late submissions will lose 20 points per day.

If you do not follow this submission guideline, you will lose 10 points out of 100.

You should also submit a hard copy of your code to TA. Your report must have a cover page with your student ID and name. In the report, your code must be well commented to explain your algorithm. Also, the sample input and the output of your program must be included in the report.