COP 3503 Honors – Homework 2 (Non-Collaborative)

Due Date: September 21, 2018

1. Write pseudocode for the brute force method of solving the maximum-subarray problem. Your procedure should run in $\Theta(n^2)$ time.

2. Implement both the brute force and recursive algorithms for the maximum-subarray problem (use any programming language you wish). What problem size n_0 gives the crossover point at which the recursive algorithm beats the brute force algorithm? Then change the base case of the recursive algorithm to use the brute force algorithm whenever the problem is less than n_0. Does that change the crossover point? Please provide a listing of your code when you hand in the assignment.

3. How would you modify the QUICKSORT algorithm so that it sorts the numbers in nonincreasing order? Provide pseudocode for your solution.

4. Optional Challenge Problem 10191 - Longest Nap

   Submit your solution to the following website for automatic grading
   https://uva.onlinejudge.org/index.php

   As you may already know, there are professors very busy with a filled schedule of work during the day. Your professor, let's call him Professor P, is a bit lazy and wants to take a nap during the day, but as his schedule is very busy, he doesn't have a lot of chances of doing this. He would REALLY like, however, to take one nap every day. Because he'll take just one nap, he wants to take the longest nap that is possible given his schedule. He decided to write a program to help him in this task but, as we said, Professor P is very lazy. So, he finally decided that YOU must write the program!

   Input
   The input will consist on an arbitrary number of test cases, each test case represents one day. The first line of each set 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:

   *time*1 *time*2 *appointment*

Where *time*1 represents the time which the appointment starts and *time*2 the time it ends. All times will be in the `hh:mm' format, *time*1 will always be strictly less than *time*2, they will be separated by a single space and all times will be greater than or equal to 10:00 and less than or equal to 18:00. So, 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 in the same line. You can assume that no line will be longer than 255 characters, that 10 <= *hh* <= 18 and that 0 <= *mm* < 60.
You CAN'T assume, however, that the input will be in any specific order. You 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 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 minutes', where

H = *X* / 60 (integer division, of course) and *M* = *X* mod 60*:*

Notice that you don't have to worry with concordance (i.e. you must print `1 minutes' or `1 hours' if it's the case).

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

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 day all busy (i.e. you may assume that there will be 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.
```