22C:16 Quiz 11

The two problems in this quiz involve writing a little bit of code - at most 4-6 lines each. If you see yourself writing too much, it is time to stop and think. Turn the page for the second problem.

1. You are given a list `L` of numbers and your task is to write a *recursive* function to determine the minimum number in `L`. Use the following function header:

   ```
   def minimum(L):
   ```

   For example, if `L` is `[21, 3, 7, 67, 19, 210, 21]` then the function should return `3`.

   Of course this problem can be solved non-recursively, but you will not receive any credit for a non-recursive solution, even if it is correct. And, by the way, do not forget to specify the base cases.
   Hint: To find the minimum number in `L` first find the minimum number in the sublist of `L` that excludes the first element. Then you just have to compare this with the first element in `L` to determine the answer.

2. You are given a sorted list `L` of numbers and your task is to write a *recursive* function to determine two numbers in the list that are closest. Use the following function header:

   ```
   def closest(L):
   ```

   For example, if `L` is `[3, 7, 67, 68, 210, 215]` then the function should return `[67, 68]`. Note that the object that is returned is a size-2 list.

   Of course this problem can be solved non-recursively, but you will not receive any credit for a non-recursive solution, even if it is correct. And, by the way, do not forget to specify the base cases.

   Hint: To find the closest pair of numbers in `L` first find the closest pair in the sublist of `L` that excludes the first element. Then you just have to compare this with the pair consisting of the first and second elements in `L`.