A very simple algorithm called $\text{find}_\text{max}$ is designed for the following problem:

Problem: Given a list of positive numbers, return the largest on the list.

Inputs: A list $L$ of positive numbers. This list must contain at least one number. (Asking for the largest number in a list of no numbers is not a meaningful question.)

Outputs: A number $n$, which will be the largest number of the list.

1. Write an implementation of the following algorithm which solves the above problem. $(7 \text{ points})$
   
   1. Set $\text{max}$ to 0.
   2. For each number $x$ in the list $L$, compare it to $\text{max}$. If $x$ is larger, set $\text{max}$ to $x$.
   3. $\text{max}$ is now set to the largest number in the list.

2. Write an implementation of the following algorithm, which solves the same problem. $(7 \text{ points})$
   
   1. If $L$ is of length 1, return the first item of $L$.
   2. Set $v_1$ to the first item of $L$.
   3. Set $v_2$ to the output of performing $\text{find}_\text{max}()$ on the rest of $L$.
   4. If $v_1$ is larger than $v_2$, return $v_1$. Otherwise, return $v_2$.

3. For each of the above two algorithms in Q1 and Q2, answer the following questions by “Yes” or “No” with any necessary comments. $(6 \text{ points})$
   
   A) Does it have defined inputs and outputs?
   B) Is it guaranteed to terminate?
   C) Does it produce the correct result?