Question 7

[16 marks in total]

Consider the following procedure:

```
procedure sort (var A : array 1 .. * of int)
    const n : int := upper(A)      % Note that n >= 1
    var i : int := 1

    loop
        % Loop invariants would go here
        exit when i = n

        var candidate : int := i
        for j : i + 1 .. n
            if A (j) < A (candidate) then
                candidate := j
            end if
        end for

        const temp : int := A (candidate)
        A (candidate) := A (i)
        A (i) := temp

        i := i + 1
    end loop

end sort
```

a. [3 marks]

Trace the procedure with $A = (17, 13, 23, 4, 62)$ and $n = 5$. Show the value of i and contents of the array A at the start of every iteration of the outer loop.

i	contents of A at the start of the ith iteration
1	(17, 13, 23, 4, 62)
2	
3	
4	
5	

b. [5 marks]
Let $A_0[j]$ denote the *initial* value of $A(j)$ and let $A[j]$ denote the value of $A(j)$ just before an arbitrary iteration of the loop. Which of the following are loop invariants for the outer loop in procedure `sort`? Circle the appropriate response.

	Is it a loop invariant?
1) $A_0[k] \leq A[k]$ for all $1 \leq k \leq n$	Yes No
2) If $i > 1$ then $A[1] \leq A[2] \leq \cdots \leq A[i-1]$	Yes No
3) If $i > 1$ then $A[i-1] \leq A[k]$ for all k such that $i \leq k \leq n$	Yes No
4) $A[1..n]$ contains exactly the same elements as $A_0[1..n]$, possibly rearranged.	Yes No
5) If $A_0[k] \neq A_0[k+1]$, then $A[k] \neq A[k+1]$ for all $1 \leq k \leq n-1$	Yes No

c. [8 marks]
Assume that the appropriate loop invariants from above have been proven, and that the procedure terminates successfully. Prove that the procedure returns the array A rearranged so that A is sorted in non-decreasing order.