CIS 575. Introduction to Algorithm Analysis
Remarks on Assignment #1, Spring 2014

Question 1 The body of FindLast(A, n, x) can be written:

```
if A[n] = x
    return n
else
    return FindLast(A, n − 1, x)
```

Question 2 We shall prove that for all $n \geq 0$, FindLast meets its specification[1], and shall do so by induction in n. The base case is when $n = 0$ in which case the precondition does not hold so the specification is vacuously fulfilled.

We now consider the case where $n \geq 1$, assume that x occurs in $A[1..n]$, and split into two cases:

- if $A[n] = x$, then we return n which satisfies the postcondition since $1 \leq n \leq n$ and $A[n] = x$ and x does not occur in the empty $A[n + 1..n]$.
- If $A[n] \neq x$, we infer that x occurs in $A[1..n - 1]$; hence the precondition holds for the recursive call FindLast$(A, n - 1, x)$ and we can inductively assume that the postcondition holds. That is, the call returns p with $1 \leq p \leq n - 1$ such that $A[p] = x$ and x does not occur in $A[p + 1..n - 1]$. But this implies $1 \leq p \leq n$ and $A[p] = x$ and that x does not occur in $A[p+1..n]$, which amounts to the desired postcondition.

Question 3 The body of FindLast(A, n, x) can be written:

```
p ← n
while A[p] ≠ x
    p ← p − 1
return p
```

Question 4 As invariant for the while loop, we shall use:

$1 \leq p \leq n$ and x occurs in $A[1..p]$ and x does not occur in $A[p + 1..n]$.

That this invariant is properly **initialized** follows trivially from the precondition of FindLast, and the assignment before the while loop, since $A[n + 1..n]$ is empty.

To show that the invariant is **maintained**, we observe that if $A[p] \neq x$ then the invariant implies that x occurs in $A[1..p - 1]$, and thus $1 \leq p - 1$, and that x does not occur in $A[p..n]$.

To show that the while loop **terminates**, note that each iteration will decrease p which cannot go on forever since the invariant ensures $1 \leq p$.

Finally, to show **correctness**, observe that at loop exit we have $A[p] = x$ which together with the loop invariant yields the desired postcondition.

[1] It is OK to consider only $n \geq 1$.