2.1-3 Linear Search

```
1: procedure LINEAR-SEARCH(A, v)
2:     for i ← 1 to length[A] do
3:         if A[i] == v then
4:             return i
5:     return NILL
```

Proof of correctness To prove linear search is correct, I formulate a *loop invariant* that need to hold true at *initialization*, during *maintainance* and at *termination*.

Loop invariant At the start of loop iteration i, the value v is not in the sub-array $A[1..i-1]$. In formal terms, $v \notin \{A[j]|j \in \{1, ..., i-1\}\}$

Initialization At initialization $i = 0$, so the sub-array $A[1..i-1]$ is empty. So v is trivially not a member of that array, and the loop invariant holds.

Maintenance Assuming that v is not in the sub-array $A[1..i-1]$ at the start of iteration i, then two outcomes are possible during the iteration. Either, $A[i] == v$ and the loop terminates, or $A[i] \neq v$ and v is not an element of the array $A[1..i]$ which is the loop invariance condition for iteration $i+1$.

Termination The algoritm terminates in two different ways. The first happens when $A[i] == v$, then the loop invariant still holds from the initialazation and maintenance steps $v \notin A[1..i-1]$. The second happens when $i = n+1$, in that case $v \notin A[1..i-1] = A[1..n]$, which is the entire array, and the procedure returns NILL.