2.1-3.

Consider the ***searching problem*:**

Input. A sequence of n numbers $A = \langle a_1, a_2, ..., a_n \rangle$ and a value v.

Output. An index i such that $v = A[i]$ or the special value NIL if v does not appear in A.

Write pseudocode for ***linear search***, which scans through the sequence, looking for v. Using a loop invariant, prove that your algorithm is correct. Make sure that your loop invariant fulfills the three necessary properties.

Answer.

The pseudocode called LINEAR-SEARCH scans through the sequence, looking for the first element that equals to v, it returns NIL if no match occurs in this process.

```
LINEAR-SEARCH(A, v)
1   i=1
2   while i ≤ A.length
3         if A[i] == v
4               return i
5         else i = i + 1
6   return NIL
```

Observe that before entering each iteration of the **while** loop, which is indexecd by i, the subarray consisting of elements $A[1..i-1]$ consistute the set not containing v, and the remaining aubarray $A[i..n]$ corresponds to the sequence to be search. This reveal the following loop invariant:

> At the start of each iteration of the **while** loop of line 2–5, the element of value v does not belongs to the subarray $A[1..i-1]$.

We use this loop invariant to help us prove the correctness of LINEAR-SEARCH.

Initialization. Prior to the first iteration of the loop, we have $i = 1$, so that the subarray $A[1..i-1]$ is empty. This empty subarray does not contain the element of value v.

Maintenance. The loop continues as long as elements of value v has not been encountered. It goes by eliminating $A[1]$, $A[2]$, $A[3]$ and so on from the unexamined set $A[i..n]$ and add to the examined one, which is $A[1..i-1]$. The set without element v then expands from $A[1..i-1]$ to $A[1..i]$. Incrementing i for the next iteration of the **while** loop then preserves the loop invariant.

Email address: informlarry@gmail.com

Termination. The procedure terminates when it encounter an element of value v or run out of the array. In the first case, after passing by $i-1$ elements that are not equal to v, the element that terminate the loop is the ith one, and this index of i is returned. In second case, we must have $i = n+1$ at that time. Substituting $n+1$ for i in the statement of loop invariant, we have that the element of value v does not belongs to the subarray $A[1..n]$. Observing that the subarray $A[1..n]$ is the entire array, we conclude that v does not appear in A, and return the special value NIL.