Exercises 2.1

1 2.1-2

Rewrite the INSERTION-SORT procedure to srot into ninincreasing instead of non-decreasing order.

```
for j = 2 to A.length do
    key = A[j]
    i = j − 1
    while i > 0 and A[i] < key do
        A[i + 1] = A[i]
        i = i − 1
    end while
    A[i + 1] = key
end for
```

2 2.1-3

Consider the ***searching problem:***

Input: A sequence of n numbers $A = \langle a_1, a_2, \cdots, a_n \rangle$ 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.

```
for j = 1 to A.length do
    if A[j] = v then
        return j
    end if
end for
return NIL
```

Loop invariant:
At the start of each iteration of the **for** loop, the subarray $A[1 \dots j-1]$ doesn't contain the value of v.

Initialization:
When $j = 1$ the subarray $A[1 \dots j-1]$ contains no element and therefore doesn't contain the value v.

Maintenance:
If the algorithm reaches the iteration $n \leq A.length$ and that $A[n] = v$ then the algorithm returns n and the algorithm transfers back to the point of call in the calling procedure. Therefore it never reaches the iteration $n + 1$. So at the start of each iteration $n + 1$ the subarray $A[i \dots n]$ doesn't contains the value v.

Termination:
The **for** loop stops when $j > A.length = n$. The **for** loop increasing j by one at each iteration, $j = n + 1$ after the **for** loop. Substituting $n + 1$ in the loop invariant, we have that the subarray $A[1 \dots n]$ doesn't contains the value of v. The subarray $A[1 \dots n]$ being the whole array, the algorithm is correct.

3 2.1-4

Consider the problem of adding two n-bit binary integers, stored in two n-element arrays A and B. The sum of the two intergers shoiuld be stored in binary form in an $(n+1)$-element array C. State the problem formally and write pseudocode for adding the two integers.

Input: two n-element arrays A and B such that $\forall i \in [0 \cdots n],\ A[i] \in [0;1]$ *and* $B[i] \in [0;1]$.
Output: an array $(n+1)$-element array C containing the binary sum of A and B.

C an array of size $n+1$
$c = 0$
for i **to** n **do**
  $s = A[i] + B[i] + c$
  $C[i] = (s \mod 2)$
  $c = div(s, 2)$
end for
$C[n] = c$