

Rosen, Discrete Mathematics and Its Applications, 7th edition  
Extra Examples  
Section 3.1—Algorithms



— Page references correspond to locations of Extra Examples icons in the textbook.

---

**p.192, icon at Example 1**

**#1.**

(a) Describe an algorithm that determines the location of the last even integer in a nonempty list  $a_1, a_2, \dots, a_n$  of integers. (If no integer in the list is even, the output should be that the location is 0.)

(b) Describe the algorithm, with “last” replaced by “first”.

**Solution:**

(a) We need to find the last subscript,  $i$ , such that  $a_i$  is even, that is,  $a_i \bmod 2 = 0$ . We use *location* to keep track of the subscript. Initially we set *location* to 0 (because an even integer has not yet been found), and then proceed to examine each element of the list by advancing the subscript  $i$  one step at a time, until the end of the list is reached. Here is the pseudocode:

```
location := 0           {location is initially set to 0}
for i := 1 to n         {examine, in order, each entry a_i in the list}
if a_i mod 2 = 0 then location := i {change location to i if a_i is even, otherwise keep old location}
```

(b) Suppose we seek the location of the *first* even integer in the list. In this case the loop should end once an even integer  $a_i$  is encountered or else all entries in the list have been examined and no even integer has been encountered. We can use a while-loop

```
location := 0           {location is initially set to 0}
i := 1                  {begin by examining first element in the list}
while (location = 0 and i ≤ n) {as long as no even element has been found and there are
                                more elements in the list yet to be examined}
begin
  if a_i mod 2 = 0 then location := i {examine element a_i; if it is even, update the location}
  i := i + 1                       {advance counter to examine next element}
end
```

---

**p.192, icon at Example 1**

**#2.** Describe an algorithm that takes as input a sequence of distinct integers  $a_1, a_2, \dots, a_n$  ( $n \geq 2$ ) and determines if the integers are in increasing order.

**Solution:**

One way to do this is to examine each pair of consecutive integers,  $a_{i-1}$  and  $a_i$ , to see if  $a_i < a_{i-1}$ . If this happens, the integers are not in increasing order, and we stop and output FALSE. If this never happens, then the output remains TRUE.

```
output := TRUE
i := 2
while (i ≤ n and output = TRUE)
begin
```



```
         $k := k + 1$   
    end  
end  
end
```

---