Find maximum element in finite sequence

given input integers: $a_1, a_2, \ldots, a_n$

n, i

$max := a_1$

```
for i := 2 to n
    {if max < ai, then max := ai
     i := i+1
    }
return (max)
```

Given $a_1 = 2$, $a_2 = 1$, $a_3 = 5$, $a_4 = 3$, $a_5 = 2$:

$n = 5$, $max = a_1 = 2$

Step 1: $i = 2$ (to 5):
is $max = 2 < a_2 = 1$? NO

Step 2 $i = 3$: is $max = 2 < a_3 = 5$ YES: set $max = a_3 = 5$

Step 3 $i = 4$ is $max = 5 < a_4 = 3$ NO

Step 4 $i = 5$ is $max = 5 < a_5 = 2$ NO

↓ Exit for loop

return $max = 5$

Linear search

input: integers: n, x, a list $a_1, a_2, \ldots, a_n$ in increasing numeric order

find: the position of the integer x in the list. if x not present return 0.

local variables: integers $i := 1$, location $:= 0$

```
while (i ≤ n)
{ if (x = a_i), location := i
  i := i + 1
}
return(location)
```

Given: $a_1 = 1, a_2 = 3, a_4 = 6, a_5 = 8, a_6 = 9, a_7 = 10, a_8 = 12$
$x = 6$

$i = 1$ does $x = a_1$? NO
$i = 2$ does $x = a_2$? NO
$i = 3$ does $x = a_3$? NO
$i = 4$ does $x = a_4$? YES
location $= 4$
$i = 5$ does $x = a_5$? NO
$i = 6$ does $x = a_6$? NO
$i = 7$ does $x = a_7$? NO
$i = 8$ does $x = a_8$? NO

We now drop out of while loop with location $= 4$.

Binary Search

input:
integers: x, and $a_1, a_2, \ldots, a_n$ (in increasing order)
n

find where in the list $a_1, \ldots, a_n$ one finds integer x. If x not in list return 0.

local variables:
integers: $i := 1, j := n$; location $:= 0, m$

```
while (i < j)
{ m := ⌊ (i+j)/2 ⌋
   if (x > am), i := m+1
   else j := m
}
if (x = ai), location := i
return (location)
```

Given $a_1 = 1, a_2 = 3, a_3 = 4, a_4 = 6, a_5 = 8, a_6 = 9, a_7 = 10, a_8 = 12$
$x = 6$

Step 1: $i = 1$ and $j = 8$
$m = 4$, is $x > a_4$?, NO
thus $j = 4$.

Step 2: $i = 1$ and $j = 4$
$m = 2$, is $x > a_2$? YES
thus $i = 3$

Step 3: $i = 3$ and $j = 4$
$m = 3$ and is $x > a_3$. YES
thus $i = 4$

Loop is now finished, with $x = a_4 = 6$. Location number $i = 4$ is returned

Bubble Sort

Input: "real" numbers $a_1, a_2, \ldots, a_n$
integer n

Sort the numbers $a_1, \ldots, a_n$ so that they occur in increasing numeric order

---

```
{ for (i := 1 to n-1)
    { for (j := 1 to n-i
        { if a_j > a_{j+1}, interchange a_j and a_i
        }
    }
}
```

Insertion Sort

input: "real" numbers $a_1, a_2, \ldots, a_n$
n integer

Sort so that numbers occur in increasing numeric order

local variables: integers i, j, k

```
for (j := 2 to n)
{   i = 1
    while (a_j > a_i)
    { i := i+1
    }
    m := a_j
    for (k := 0 to j-i-1)
        a_{j-k} := a_{j-k-1}        } shuffle
    a_i := m
}
```

Example: $a_1 = 0,\ a_2 = 2,\ a_3 = 4,\ a_4 = 1,\ a_5 = 5$

$j = 2$ $i = 1$

Is $a_2 > a_1$? YES

Set $i = i+1 = 2$

drop out of loop

$m = a_2$

for ($k = 0$ to $2-2-1 = -1$)

do not execute

$a_2 = m$ · no change

$j = 3$ $i = 1$

Is $a_3 > a_1$? YES

Set $i = i+1 = 2$

Is $a_3 > a_2$? YES

Set $i = i+1 = 2+1 = 3$

$m = a_3$

do not execute "for" loop

$a_3 = m$ no change

$j = 4$ $i = 1$

Is $a_4 > a_1$? YES

Set $i = i+1 = 2$

Is $a_4 > a_2$? NO

$m = a_4 = 1$ ← *1

for ($k = 0$ to $4-2-1 = 1$)

$k = 0$ $a_4 = a_3$ giving

$a_1 = 0,\ a_2 = 2,\ a_3 = 4,\ a_4 = 4,\ a_5 = 5$

$k = 1$ $a_3 = a_2$ giving

$a_1 = 0,\ a_2 = 2,\ a_3 = 2,\ a_4 = 4,\ a_5 = 5$

$a_2 = m$ giving

$a_1 = 0,\ a_2 = 1,\ a_3 = 2,\ a_4 = 4,\ a_5 = 5$