

10. Descifrá qué hacen los siguientes algoritmos, explicar cómo lo hacen y reescribirlos asignando nombres adecuados a todos los identificadores

```
proc q (in/out a: array[1..n] of T)
  for i:= n-1 downto 1 do
    r(a,i)
  od
end proc
```

```
proc r (in/out a: array[1..n] of T, in i: nat)
  var j: nat
  j:= i
  do j < n ∧ a[j] > a[j+1] → swap(a,j+1,j)
    j:= j+1
  od
end proc
```

- **¿Qué hace?:** el algoritmo ordena los elementos del arreglo de menor a mayor.
- **¿Cómo funciona?:** funciona como la ordenación por inserción pero de manera inversa, comparando los elementos de derecha a izquierda y formando un segmento final ordenado de  $[i,n]$  junto a otro de  $[1,i)$  sin insertar todavía

- **Nombres de variables:**

★  $q \rightarrow \text{insertion\_sort\_inverso}$

★  $r \rightarrow \text{insert\_inverso}$

- **Reescritura de los algoritmos con sus nuevas variables:**

```
proc insertion_sort_inverso(in/out a: array[1..n] of T)
  for i := n-1 downto 1 do
    insert_inverso(a,i)
  od
end proc

proc insert_inverso(in/out a: array[1..n] of T, in i: nat)
  var j: nat
  j := i
  do j < n ∧ a[j] > a[j+1] → swap(a,j+1,j)
    j := j+1
  od
end proc
```