

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

```
proc p (in/out a: array[1..n] of T)
  var x: nat
  for i:= n downto 2 do
    x:= f(a,i)
    swap(a,i,x)
  od
end proc
```

```
fun f (a: array[1..n] of T, i: nat) ret x: nat
  x:= 1
  for j:= 2 to i do
    if a[j] > a[x] then x:= j fi
  od
end fun
```

- **¿Qué hace?:** toma un arreglo y permuta sus elementos, ordenándolos de menor a mayor.
- **¿Cómo lo hace?:** básicamente funciona como la ordenación por selección de manera inversa pues: Selecciona el MAYOR elemento del arreglo (no el menor) y lo intercambia con el elemento de la ÚLTIMA posición (no la primera). Luego toma el mayor elemento de los restantes, lo lleva a la penúltima posición y así sucesivamente hasta finalizar.
- **Nombres de variables:**
 - ★ **p** → *selection_sort_inverso*
 - ★ **f** → *may_pos_from*
 - ★ **x** → *mayor_posicion*
- **Reescritura de los algoritmos con sus nuevas variables:**

```
proc mayor_elemento_al_ultimo(in/out a: array[1..n] of T)
  var x: nat
  for i := n downto 2 do
    x := posicion_mayor_elemento(a,i)
    swap(a,i,x)
  od
end proc

fun posicion_mayor_elemento(a: array[1..n] of T, i: nat) ret mayor_posicion: nat
  mayor_posicion := 1
  for j := 2 to i do
    if a[j] > a[x] then mayor_posicion := j fi
  od
end fun
```