

1. Escribí algoritmos para resolver cada uno de los siguientes problemas sobre un arreglo a de posiciones 1 a n , utilizando **do**. Elegí en cada caso entre estos dos encabezados el que sea más adecuado:

<pre>proc nombre (in/out a:array[1..n] of nat) ... end proc</pre>	<pre>proc nombre (out a:array[1..n] of nat) ... end proc</pre>
---	--

- (a) Inicializar cada componente del arreglo con el valor 0.

```
proc todos_0(out a: array[1..n] of nat)
  var i: nat
  i := 1

  while i ≤ n do
    a[i] := 0
    i := i+1
  od
end proc
```

- (b) Inicializar el arreglo con los primeros n números naturales positivos.

```
proc primeros_naturales(out a: array[1..n] of nat)
  var i: nat
  i := 1

  while i ≤ n do
    a[i] := i
    i := i+1
  od
end proc
```

- (c) Inicializar el arreglo con los primeros n números naturales impares.

```
proc primeros_naturales_impares(out a: array[1..n] of nat)
  var i: nat
  i := 1

  while i ≤ n do
    a[i] := i+(i-1)
    i := i+1
  od
end proc
```

- (d) Incrementar las posiciones impares del arreglo y dejar intactas las posiciones pares.

```
proc incrementar_pos_impares(in/out a: array[1..n] of nat)
  var i: nat
  i := 1

  while i ≤ n do
    a[i] := a[i]+1
    i := i+2
  od
end proc
```