

1. (a) Dado el siguiente arreglo, muestre cómo queda el mismo luego de cada modificación que realiza el algoritmo **insertion sort**.

[3, 17, 2, 10, 5, 8, 3]

- (b) Escriba un algoritmo que, dado un arreglo y un elemento e que no pertenece al mismo, ubique los menores a e al comienzo del arreglo y devuelva la cantidad de elementos menores a e . Utilice para ello el siguiente encabezado:

proc p (in/out a: array[1..n] of Int, in e: Int, out k: Nat)

2. Dado el siguiente procedimiento

```

proc p (in/out l: list)
  var a, b: pointer to node
  a := l
  while a ≠ null do
    b := a → next
    if b ≠ null then
      a → next := b → next
      free(b)
    fi
    a := a → next
  od
end proc

```

donde los tipos **node** y **list** se definen como sigue

```

type node = tuple
  value: elem
  next: pointer to node
end
type list = pointer to node

```

- (a) Explique qué hace el procedimiento p.
- (b) ¿Cuál es el orden del procedimiento p? Justificá la respuesta.
- (c) Si se llama al procedimiento p con una lista que tiene los valores 1, 2, 3, 4, 5, 6 y 7, en ese orden, ¿qué valores tendrá la lista luego de la llamada?

3. Para el siguiente algoritmo, plantee la recurrencia en función de n y calcule el orden contando la cantidad de asignaciones a la variable t :

```

fun f (n: nat) ret t: nat
  if n ≤ 1 then t := 1
  else
    t := f(n/2)
    t := t + f(n/2)
    for i := 1 to n do
      for j := 1 to n do
        t := t + 1
      od
    od
  fi
end fun

```

4. Un **multiconjunto** M con elementos en S , es un subconjunto de S donde cada elemento tiene asociado un número natural que indica cuántas veces el elemento ocurre. En otras palabras, un multiconjunto puede pensarse como un conjunto en el cual los elementos pueden estar incluidos más de una vez, indicando cuántas veces lo hace.
- (a) Especificar el TAD **Multiset of T** de multiconjuntos con elementos de tipo T , incluyendo constructores para crear un multiconjunto vacío y otro para agregar un elemento a un multiconjunto. Y operaciones para saber si un multiconjunto es vacío o no, para saber si un elemento dado existe en el multiconjunto, para obtener cuántas veces un elemento dado ocurre en un multiconjunto, y para eliminar una ocurrencia de un elemento en un multiconjunto.
 - (b) Implementar el TAD **Multiset of T** especificado en el punto anterior utilizando como representación interna una lista donde cada elemento es un par consistente de un elemento de tipo T y un número natural que indica la cantidad de ocurrencias.
 - (c) Implementar una función que reciba un arreglo de N naturales y devuelva el multiconjunto que contiene todos los elementos pares que ocurren en el arreglo. Todo tipo de datos utilizado en la función debe utilizarse de manera **abstracta**.