

## Corrigé examen - ASD -

### Exercice 1

```
Fonction Hailstone(n: entier): entier
Debut
  Ecrire(n)

  si(n = 1) retourner 1
  sinon
    si(n mod 2 = 0) retourner 1 + Hailstone(n div 2)
    sinon retourner 1 + Hailsonte(3 * n + 1)
  Fin si
Fin si
Fin
```

### Exercice 2

```
1. Procédure supp(var l:liste)
  var p:liste
  Debut
    tantque(l != NIL)faire
      p <-- l
      l <-- l^.suivant
      liberer(p)
    Fin tantque
  Fin
```

#### 2. (a) Version 1

```
Fonction fusion(l1:liste, l2:liste):liste
  var l3, p, q, r: liste
  i:entier
  Debut
    i <-- 0
    p <-- l1
    q <-- l2

    allouer(l3)
    r <-- l3

    tantque(p != NIL ou q != NIL)faire
      si((p != NIL) et (i mod 2 = 0))alors
        r^.val <-- p^.val
        p <-- p^.suivant
      sinon si(q != NIL)alors
```

```

        r^.val <-- q^.val
        q <-- q^.suivant
    sinon si(p != NIL)alors
        r^.val <-- p^.val
        p <-- p^.suivant
    Fin si

    si(p != NIL ou q != NIL)alors
        allouer(r^.suivant)
        r <-- r^.suivant
    sinon
        r^.suivant <-- NIL
    Fin si

    i <-- i + 1
Fin tantque

retourner l3
Fin

```

(b) **Version 2**

```

Fonction fusion(l1: liste, l2: liste): liste
    var p, q, r: liste
        i: entier
Debut
    i <-- 0
    p <-- l1
    q <-- l2

    tantque(p != NIL et q != NIL)faire
        si(i mod 2 = 0) alors
            r <-- p^.suivant
            p^.suivant <-- q
            p <-- r
        sinon
            r <-- q^.suivant
            q^.suivant <-- p
            q <-- r
        Fin si

        i <-- i + 1
    Fin tantque

    retourner l1
Fin

```

**Exercice 3**

1. Voir le cours
2. Fonction noeudsInterne(A: Arbre): Entier  
Debut

```
si(A = NIL)alors retourner 0
sinon si(A^.gauche = NIL et A^.droit = NIL) alors retourner 0
sinon retourner 1 + noeudsInterne(A^.gauche) + noeudsInterne(A^.droit)
fin si
Fin
```