

Tri par fusion/éclatement

const MAX = 10

type vecteur = **tableau**[1..MAX] **de type** entier

procedure eclatement(V : vecteur, **var** V1 : vecteur, **var** V2 : vecteur, **var** i1 : entier, **var** i2 : entier)

/ les éléments de V sont répartis dans V1 et V2 : V1 contient les monotonies de rang impair, V2 celles de rang pair*/*

var
rangerdansV1 : **booleen**
i, precedent : **entier**

debut
 precedent := V[1]
 rangerdansV1 := **vrai**
 i1 := 0
 i2 := 0
 pour i **allant de** 1 **à** MAX **faire**
 si V[i] < precedent **alors**
 rangerdansV1 := **non**(rangerdansV1)
 fsi
 si rangerdansV1 **alors**
 i1 := i1 + 1
 V[i1] := V[i]
 sinon
 i2 := i2 + 1
 V[i2] := V[i]
 fsi
 precedent := V[i]
 fpour
fin

procedure fusion(**var** V : vecteur, V1 : vecteur, V2 : vecteur, nb1 : entier, nb2 : entier)

/ on parcourt simultanément V1 et V2 : le plus petit des deux éléments de tête de V1 et V2 est placé dans V : on avance dans le tableau qui a fourni cet élément. On copie ensuite les éléments du tableau restant non vide*/*

var
 i, i1, i2 : **entier**
debut
 i := 1
 i1 := 1
 i2 := 1
 tq (i1 ≤ nb1) et (i2 ≤ nb2) **faire**
 si V1[i1] ≤ V2[i2] **alors**
 V[i] := V1[i1]
 i1 := i1 + 1
 sinon
 V[i] := V2[i2]
 i2 := i2 + 1
 fsi
 i := i + 1
 ftq
 tq (i1 ≤ nb1) **faire**
 V[i] := V1[i1]
 i1 := i1 + 1
 i := i + 1
 ftq
 tq (i2 ≤ nb2) **faire**
 V[i] := V2[i2]
 i2 := i2 + 1
 i := i + 1
 ftq
fin

procedure tri_fusion_eclatement(**var** v : vecteur)

/ tri fusion/éclatement : un premier éclatement suivi d'un « couple » fusion/éclatement. Le tri s'arrête quand il ne reste plus d'éléments dans le second tableau*/*

var
 n1, n2 : **entier**
 v1, v2 : vecteur
debut
 eclatement(v, v1, v2, n1, n2)
 tq (n2 <> 0) **faire**
 fusion(v, v1, v2, n1, n2)
 eclatement(v, v1, v2, n1, n2)
 ftq
fin