

## TD10 : éléments de correction

### Exercice 3

Fonction : **FusionListesTriees**

Précondition : liste1 et liste2 triées dans l'ordre croissant

Entrée : - liste1 : liste d'entiers

- liste2 : liste d'entiers

Local : - indice1, indice2 : entiers

Sortie : - listeFusion : liste d'entiers

Début

Soit listeFusion = liste d'entiers

**Si** Longueur(liste1) == 0 ET Longueur(liste2) == 0 **Alors**

    Retourner listeFusion

**Fin Si**

**Si** Longueur(liste1) == 0 **Alors**

    Soit listeFusion = Copier(liste2)

    Retourner listeFusion

**Fin Si**

**Si** longueur(liste2) == 0 **Alors**

    Soit listeFusion = Copier(liste1)

    Retourner listeFusion

**Fin Si**

Soit indice1 = 0

Soit indice2 = 0

**Tant Que** indice1 < Longueur(liste1) ET indice2 < Longueur(liste2) **Faire**

**Si** liste1[indice1] < liste2[indice2] **Alors**

        Ajouter(listeFusion, liste1[indice1])

        Soit indice1 = indice1 + 1

**Sinon**

        Ajouter(listeFusion, liste2[indice2])

        Soit indice2 = indice2 + 1

**Fin Si**

**Fin Tant Que**

**Si** indice1 < Longueur(liste1) **Alors**

**Pour** i **de** indice1 **à** Longueur(liste1) - 1 **avec un pas de +1 Faire**

        Ajouter(listeFusion, liste1[i])

**Fin Pour**

**Fin Si**

**Si** indice2 < Longueur(liste2) **Alors**

**Pour** i **de** indice2 **à** Longueur(liste2) - 1 **avec un pas de +1 Faire**

        Ajouter(listeFusion, liste2[i])

**Fin Pour**

**Fin Si**

Retourner listeFusion

Fin

Fonction : **TriInsertionListe**

Entrée : - liste : liste d'entiers

Local : /

Sortie : - listeTrie : liste d'entiers

Début

Soit listeTrie = liste d'entiers

**Si** Longueur(liste) == 0 **Alors**

**Retourner** listeTrie

**Fin Si**

**Ajouter**(listeTrie, liste[0])

**Pour** i de 1 à Longueur(liste) - 1 **avec un pas de** +1 **Faire**

**Soit** valeur = liste[i]

**Soit** indice = 0

**Soit** stop = FAUX

**Tant Que** indice < Longueur(listeTrie) ET stop est FAUX **Faire**

**Si** listeTrie[indice] ≥ valeur **Alors**

**Insérer**(listeTrie, valeur, indice)

**Soit** stop = VRAI

**Fin Si**

**Soit** indice = indice + 1

**Fin Tant Que**

    // ajout en fin

**Si** stop est FAUX **Alors**

**Ajouter**(listeTrie, valeur)

**Fin Si**

**Retourner** listeTrie

**Fin**