

## TD6 : éléments de correction

### Exercice 4 : Tableau trié

Fonction : **VérifierTableauTrié**

Entrée : - tab : **tableau 1D** de taille **entiers**

- taille : **entier**

Local : - i : **entier**

Sortie : - trie : **booléen**

Début

```
Soit trie = vrai
Si taille > 1 Alors
  Soit i = 1
  Tant Que i < taille OU trie == faux Faire
    Si tab[i] < tab[i-1] Alors
      Soit trie = faux
    Fin Si
    i = i + 1
  Fin Tant Que
Fin Si
Retourner trie
```

Fin

Procédure : **Main**

Entrée : **void**

Local : - tableau1 : **tableau 1D** de tailleTab **entiers**

- tableau2 : **tableau 1D** de tailleTab **entiers**

- tailleTab1 : **entier**

- tailleTab2 : **entier**

- tabTrie1 : **booléen**

- tabTrie2 : **booléen**

Sortie : **void**

Début

```
Soit tailleTab = 10
Soit tableau1 = tableau 1D de tailleTab entiers initialisé à {1,2,3,4,5,6,7,8,9,10}
Soit tabTrie1 = VérifierTableauTrié(tableau1)
Si tabTrie1 == vrai Alors
  Afficher("le tableau 1 est trié.")
Sinon
  Afficher("le tableau 1 n'est pas trié.")
Fin Si
Soit tableau2 = tableau 1D de tailleTab entiers initialisé à {1,2,3,4,6,5,7,8,9,10}
Soit tabTrie2 = VérifierTableauTrié(tableau2)
Si tabTrie2 == vrai Alors
  Afficher("le tableau 2 est trié.")
Sinon
  Afficher("le tableau 2 n'est pas trié.")
Fin Si
```

Fin

## Exercice 4 : Recherche dans un tableau

Fonction : **CompterApparitionsEntierDansTableau1D**

Entrée :  
- tab : **tableau 1D** de taille **entiers**  
- taille : **entier**  
- n : **entier**

Local :  
- i : **entier**

Sortie :  
- nApparitions : **entier**

Début

```
Soit nApparitions = 0
Pour i de 0 à taille avec un pas de +1 Faire
|   Si (tab[i] == n) Alors
|   |   Soit nApparitions = nApparitions + 1
|   Fin Si
Fin Pour
Retourner nApparitions
```

Fin

Procédure : **Main**

Entrée :  
**void**

Local :  
- tableau : **tableau 1D** de tailleTab **entiers**  
- tailleTab : **entier**  
- nUtilisateur : **entier**  
- apparitions : **entier**

Sortie :  
**void**

Début

```
Soit tailleTab = 10
Soit tableau = tableau 1D de tailleTab entiers initialisé à {1,2,3,4,2,5,6,7,1,2}
Soit nUtilisateur = SaisirEntier()
Soit apparitions = CompterApparitionsEntierDansTableau1D(tableau, nUtilisateur)
Afficher("L'entier {nUtilisateur} apparaît {apparitions} fois dans le tableau.")
```

Fin

#### Exercice 4 : Recherche multiple dans un tableau trié

Fonction : **CompterApparitionsEntierDansTableau1D** (celle définie précédemment)

Entrée :  
- tab : **tableau 1D** de taille **entiers**  
- taille : **entier**  
- n : **entier**

Local :  
- i : **entier**

Sortie :  
- nApparitions : **entier**

Début

```
Soit nApparitions = 0
Pour i de 0 à taille avec un pas de +1 Faire
|   Si (tab[i] == n) Alors
|   |   Soit nApparitions = nApparitions + 1
|   Fin Si
Fin Pour
Retourner nApparitions
```

Fin

Procédure : **AfficherApparitionsEntiersEntreBornesDansTableau1D**

Entrée :  
- tab : **tableau 1D** de taille **entiers**  
- taille : **entier**  
- borneInf : **entier**  
- borneSup : **entier**

Local :  
- i : **entier**  
- nApparitions : **entier**

Sortie :  
**void**

Début

```
Soit nApparitions = 0
Pour i de borneInf à borneSup avec un pas de +1 Faire
|   Soit nApparitions = CompterApparitionsEntierDansTableau1D(tab, i)
|   Afficher("l'entier {i} apparaît {nApparitions} fois dans le tableau.")
Fin Pour
```

Fin

Procédure : **Main**

Entrée :  
**void**

Local :  
- tableau : **tableau 1D** de tailleTab **entiers**  
- tailleTab : **entier**  
- borneInfU : **entier**  
- borneSupU : **entier**

Sortie :  
**void**

Début

```
Soit tailleTab = 10
Soit tableau = tableau 1D de tailleTab entiers initialisé à {1,2,3,4,2,5,6,7,1,2}
Soit borneInfUtilisateur = SaisirEntier()
Soit borneSupUtilisateur = SaisirEntier()
AfficherApparitionsEntiersEntreBornesDansTableau1D(tableau, borneInfU, borneSupU)
```

Fin