**Chapitre 5**

**Les Tableaux**

I - Définition et Déclaration :

Un tableau est une collection, un ensemble, un regroupement d’objets de données de mêmes type. Qui occupent des espaces mémoires contigüs (les uns à côté des autres) et repérés par des indices (position ou rang de l’élément dans le tableau).

Un tableau est définie par un nom, une taille et une dimension.

*nom :* Ensembles de lettres, chiffres et de caractère “ \_ “. Le nom est un identificateur.

*la taille :* C’est le nombre maximum d’éléments que peut contenir un tableau

*la dimension :* On distingue des tableaux **vecteurs** ou **unidimensionnls** et des tableaux **matrices** ou **bidimentionnels**.

* vecteurs : composé d’une ligne et plusieurs colones  
  °*Déclaration :*

algo → nomtbleau : tableau[1...taille] de type.

C → type nomtableau [taille];

PHP → $nomtableau = array();  
 °*Accès à un élément :*

algo → tab[3] ← 15

C → tab[2] = 15;

PHP → $tab[2] = 15;

* matrices : composé de plusieurs lignes et plusieurs colones.

° *Déclaration :*

algo → nomtableau : tableau[1...nb lignes] [1...nb colones] de type

C → type nomtableau [nblignes] [nb colones];

PHP → $nomtableau = array();

° *Accès à un éléments :*

algo → mat [2] [3] ← 15

C → mat [1] [2] = 15;

PHP → $mat [1] [2] = 15;

II - Les algorithmes élémentaires sur le vecteur :

|  |  |
| --- | --- |
| Francais | Algo |
| remplissage | Algo : remplissage  Déclaration :  | tab : tableau[1...10] de entier  | i : entier  Début :  | Pour i allant de 1 à 10 faire  | | afficher(“donner une valeur”)  | | saisir(tab[i])  | Fin Pour Fin remplissage |
| Affichage | Algo : affichage  Déclaration :  | tab : tableau[1….20] de entier  | i : entier Début  | //on suppose le tableau rempli  | pour i allant de 1 à 10 faire  | | afficher(“Element numéros : ”,i,” = ”,tab[i])  | fin pour fin afficher |
| Recherche  séquentielle | Algo : recherche  Déclaration :  | tab : tableau[1...10] de entier  | i : entier  | trouve : booléen  | valeur : entier  Début :  | Pour i allant de 1 à 10 faire  | | affciher(“Donner un Element”)  | | saisir(tab[2])  | Fin Pour  | affciher (“Donner une valeur à rechercher”)  | saisir(valeur)  | trouve ← faux  | i ← 1  | tantque i <= 10 et trouve = faux faire  | | si tab[2]=valeur  | | | alors trouve ← vrai  | | | sinon i ← i+1  | | fin si  | fin tantque  | si trouve = vrai  | |alors afficher(“La valeur existe à la case”,i)  | | sinon afficher(“Le tableau ne contient pas cette valeur”)  | fin si  | fin recherche |

|  |  |
| --- | --- |
| Français | Algo |
| Tri par permutation | Algo : tripermutation  Déclaration :  | tab : tableau[1...10] de entier  | i, j, temp : entier  Début :  | // on suppose le tableau rempli  | Pour i allant de 1 à 9 faire  | | pour j allant de i+1 à 10 faire  | | | si tab[i] > tab[j] alors  | | | temp ← tab[i]  | | | tab[i] ← tab[j]  | | | tab[j] ← temp  | | | fin si  | | fin pour  | fin pour  | fin tripermutation |
| Tri par bulle | Algo : tribulle  Déclaration :  | tab : tableau[1...10] de entier  | i, temp : entier  | permute : boolean  Début :  | // on suppose le tableau rempli  | faire  | permute ← faux  | | pour i allant de 1 a 9 faire  | | | si tab[i] > tab[i+1] alors  | | | temp ← tab[i]  | | | tab[i] ← tab[j]  | | | tab[i+1] ← temp  | | | permute ← vrai  | | | fin si  | | fin pour  | tant que permute = vrai  | fin tribulle |
| Tri par sélection | Algo : triselection  Déclaration :  | tab : tableau[1...10] de entier  | i, min , indicemin,j: entier  Début :  | // on suppose le tableau rempli  | pour i allant de 1 à 10 faire  | | si tab[i] < min alors  | | min ← tab[j]  | | indicemin ← j  | | fin si  | fin pour  | tab[indicemin] ← tab[i]  | tab[i] ← min  | fin tribulle |

**Série D’exo**

*1 ⇒*

Ecrire un algo / c / php qui permet de stocker dans un tableau 10 entier et calcule :

* leur moyenne
* leur max
* leur min

Algo ExoTab1

Declaration :

tab : tableau[1...10] de entier

i ,max,min,somme : entier

moyenne : reel

Debut :

Pour i allant de 1 à 10 faire

afficher(“Donner une valeur”)

saisir(tab[i])

Fin pour

somme ← 0

pour i allant de 1 à 10 faire

somme ← somme + tab[i]

fin pour

moyenne ← somme.10

afficher(“La moyenne est de”,somme)

max ← tab[1]

min ← tab[1]

pour i allant de 2 à 10 faire

Si max<tab[i] alors

max ← tab[i]

Fin si

Si min>tab[i] alors

min ← tab[i]

Fin si

fin pour

afficher(“Le maximum est”,max)

afficher(“Le minimum est”,min)

Fin ExoTab1

*2 ⇒*

Ecrire un algo/c/php qui permet de stocker dans un tableau 10 prix et calculer la variance et l'écart type.

variance =(10 (pric)²-Moyenne(prix))/10

ecart=

Algo ExoTab2

Declaration :

tab : tableau[1...10] de reel

i ,somme : entier

variance,ecart,moyenne : reel

Debut :

Pour i allant de 1 à 10 faire

afficher(“Donner une valeur”)

saisir(tab[i])

Fin pour

somme ← 0

pour i allant de 1 à 10 faire

somme ← somme + tab[i]

fin pour

moyenne ← somme/10

variance ← 0

ecart← 0

pour i allant de 2 à 10 faire

variance ← variance +(tab[i])\*(tab[i])

fin pour

variance ← (variance - moyenne ) /10

ecart=sqrt(variance)

afficher(“La variance est”,variance)

afficher(“L'écart est”,ecart)

Fin ExoTab2

*3 ⇒*

Ecrire un prog en c qui permet de saisir un tableau de 20 entier et une position entre 1 et 20.

Le prog traite la première partie du tableau de 1 à position en ordre croissant et la dernière partie de position+1 à 20 en ordre décroissant

*4 ⇒*

Ecrire un prog en c qui stocke dans un tableau les éléments entiers puis le tri en ordre croissant. Le prog demande une valeur à retourner selon le procédé suivant :

* il compare la valeur au milieu du tableau.
* s’il y a égalité,on arrête la recherche.
* sinon selon que la valeur est inférieur (supérieur) à l’élément milieu,on la recherche dans la partie inférieur (supérieur) dans le tableau.