

# Chapitre 6 : Les Tableaux ## Définitions et citations : Un tableau est une collection, un ensemble, un regroupement d'objets de données de même type occupant des espaces mémoires contigus et repérés par des indices (rang de l'élément d'un tableau) Un tableau est défini par un nom, une taille et une dimension. Le Nom : est un identificateur composé d'une suite de lettres, chiffres et du caractère `_`. La taille est le nombre maximum que peut contenir un tableau. La dimension : on peut déclarer des tableaux uni-dimensionnel ou vecteur, bidimensionnel ou matrice.

1. Un tableau vecteur est composé d'une seule ligne et de plusieurs colonnes. - Schéma : - nomTableau : Tableau [1...Taille] de type - Exemple : Déclarer un tableau de 10 entiers - `tab : Tableau [1..10] de entier` - Accès à un élément du tableau : pour accéder à un élément du tableau, on doit fournir son rang dans le tableau. - `nomTableau[indice] <-- valeur` - Exemple : affecter la valeur 15 à la case numéro 3 du tableau `tab`. - `tab[3]<--15`

Les vecteurs en C et PHP : 1. En C `""C type nomTableau[Taille];` 2. En PHP `""PHP $nomTableau = array();` Les vecteurs en C et PHP démarrent à l'indice 0 et se terminent à l'indice `taille-1`. Exemple En C : `""C int tab[10]; tab[2]=15;` En PHP: `""PHP $tab=array(); $tab[2]=15;`

Les Matrices : tableaux de plusieurs lignes et plusieurs colonnes: déclaration d'une matrice: `nomMatrice : Tableau [1..nbLignes][1..nbColonnes]` de type Exemple : déclarer une matrice de 3 lignes, 4 colonnes de réel. `mat : Tableau[1..3][1..4] de réel` Accès à un élément : on doit fournir son rang dans la ligne et son rang dans la colonne. Affecter la valeur 15.25 à l'élément se trouvant sur la 2ème ligne et la 3ème colonne : `mat[2][3]<-- 15.25`

Déclaration de la matrice en C et en PHP : Langage C : `""C type nomMatrice[nbLigne][nbColone];` Langage PHP: `""PHP $nomMatrice = array();` Exemple : en C : `""C float mat[3][4]; mat[1][2] = 15.25;` en PHP `""PHP $mat = array(); $mat[1][2]=15.25;`

## Les Algorithmes élémentaires sur les vecteurs : ### Le Remplissage 1. Principe On parcourt le tableau case par case et pour chaque case visitée, on lui affecte la valeur saisie par l'utilisateur. Exemple : remplir un tableau vecteur de 10 entiers `""Algo Algo : remplissage Déclaration tab : Tableau[1..10]de entier i : entier Début pour i allant de 1 à 10 faire, Afficher ("Donner la valeur de la case numero ",i,":") saisir(tab[i]) fin pour fin remplissage`

Traduction en C : `""C int main() { int tab[10]; int i; for(i=0; i<10;i++) { printf("Donner la valeur de la case numero %d :",i); scanf("%d",& tab[i]); } return 0; }`

### L'affichage 1. Principe : Pour afficher les éléments du tableau, on parcourt le tableau case par case et pour chaque case visitée, on affiche son contenu. `""Algorithmes Algo : affichage Déclaration tab:Tableau[1..10] de entier i : entier Début pour i allant de 1 à 10 faire afficher(tab[i]) fin Pour affichage`

Traduction en C : `""C int main() { int tab[10]; int i; for(i=0; i<10; i++) { printf("%d", tab[i]); } return(0) }`

### La recherche séquentielle 1. Le Principe On parcourt le tableau case par case et à chaque case visitée, on compare la valeur recherchée au contenu de la case. S'il y a égalité, on arrête la recherche sinon on continue les comparaisons jusqu'à la fin du tableau. Exemple : Rechercher une valeur dans un tableau de 10 entiers `""Algo Algo : recherche Déclaration tab : Tableau [1..10] de entier i, valeur : entier trouve : boolean Début pour i allant de 1 à 10 faire afficher("Donner la valeur de la case numéro ", i, ":") saisir (tab[i]) Fin Pour afficher ("Donner la valeur à rechercher :") saisir(valeur) i<--1 trouve <--faux tant que i < 10 et trouve = faux faire si valeur = tab[i] Alors trouve <--vrai sinon i<--i+1 fin si fin tant que si trouve = faux afficher ("la valeur n'existe pas dans le tableau") sinon afficher ("la valeur existe dans le tableau à l'indice ", i) fin si fin recherche`

Traduction en C : `""C int main() { int tab[10]; int i, valeur, trouve; for(i=0; i<10; i++) { printf("Donner la valeur de la case %d :",i); scanf("%d", &tab[i]); }`

```
printf("Donner la valeur à rechercher :"); scanf("%d", &valeur ); i = 0; trouve=0; while (i<10 &&
trouve == 0) { if(valeur == tab[i]) { trouve = 1; } else { i++; } } if(trouve == 0) { printf("La valeur
n'existe pas dans le tableau"); } else { printf("La valeur existe dans le tableau à l'indice %d", i); }
return 0; } ``
```