

Les Tableaux

1) Définition et déclaration

Un Tableau est un regroupement d'éléments (ensemble, collection) de même type, occupant des espaces mémoires contiguës (les uns à côté des autres) et sont repérés par des indices (rangs de l'élément dans le tableau).

Un tableau est défini par un nom, une taille et une dimension

Le nom : est un identificateur composé par une suite de lettres, chiffres et le caractère “_”.

La taille : est le nombre maximum d'éléments que peut contenir un tableau.

La dimension : on distingue des tableaux vecteurs ou unidimensionnels (une seule dimension) composé d'une seule ligne et plusieurs colonnes, et des tableaux matrices ou bidimensionnels (plusieurs lignes, plusieurs colonnes).

Les Tableaux vecteurs :

- Déclaration en Algo : nomTableau : Tableau [1..Taille] de type
- Déclaration en C : type nomTableau [taille] ;
- Déclaration en PHP : \$nomTableau = array();
- L'accès à un élément se fait par son rang dans le tableau nomTableau[indice].
- Les tableaux en C et en PHP commencent à l'indice 0 et se terminent à l'indice taille-1.

Les Tableaux Matrices :

- Déclaration en Algo : nomTableau : Tableau [1..nbLignes][1..nbColonnes] de type
- Déclaration en C : type nomTableau [nbLignes][nbColonnes] ;
- Déclaration en PHP : \$nomTableau = array();
- L'accès à un élément se fait par son indice sur la ligne et son indice sur la colonne.
- Le premier élément en langages C et PHP se trouve à la position (0,0).

2) Les Algorithmes usuels sur les vecteurs

a- Remplissage d'un tableau

Principe : pour remplir un tableau, on réalise un parcours élément par élément et pour chaque case du tableau visitée, on stocke la valeur saisie par l'utilisateur.

Exemple: Remplir un tableau de 10 cases de type entier.

Algo : remplissage

Déclaration

tab : tableau [1.. 10] de entier

i : entier

Début

Pour i allant 1 à 10 faire

afficher (“Donner un élément :”)

saisir (tab[i])

fin pour

fin remplissage

Traduction en C :

```
#include <stdio.h>

int main () {
    int tab[10];
    int i ;
    for( i = 0; i < 10 ; i++){
        printf ("Donner un élément :");
        scanf ("%d", & tab[i]);
    }
    return 0;
}
```

b- Affichage des éléments

Principe : Pour afficher les éléments d'un tableau, on réalise un parcours case par case et pour chaque case, et pour chaque case visitée, on affiche son contenu.

Exemple : Afficher les éléments d'un tableau de 10 entiers déjà rempli.

Algo : Affichage

Déclaration

tab : tableau [1.. 10] de entier

i : entier

Début

// on suppose le tableau rempli de valeurs

pour i allant 1 à 10 faire

afficher (" Elément :", tab[i])

fin pour

fin affichage

Traduction en C :

```
#include <stdio.h>

int main () {
    int tab[10];
    int i ;
    //on suppose le tableau rempli de valeurs
    for( i = 0; i < 10 ; i++){
        printf ("Elément : %d", tab[i]);
    }
    return 0;
}
```

c- La recherche séquentielles

Principe : Pour rechercher une valeur dans un tableau, on parcourt le tableau case par case et pour chaque case visitée, on compare son contenu avec la valeur à rechercher. S'il y a égalité, on arrête la recherche, la valeur a été trouvée. Dans le cas contraire, on continue la recherche jusqu'à la fin du tableau.

Exemple: Rechercher une valeur dans un tableau de dix entiers.

Algo : recherche

Déclaration

```
tab : tableau [1.. 10] de entier
i, valeur : entier
trouve : booléen
```

Début

```
pour i allant 1 à 10 faire
    afficher ("Donner un élément :")
    saisir (tab[i])
fin pour
afficher ("Donner une valeur à rechercher :")
saisir (valeur)
trouve ← faux
i ← 1
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
    alors afficher ("La valeur n'existe pas dans le tableau")
    sinon afficher ("la valeur existe dans le tableau à la case numéro ", i)
fin si
fin recherche
```

Traduction en C :

```
#include <stdio.h>

int main () {
    int tab [ 10 ];
    int i , trouve , valeur ;
    for( i = 0; i < 10 ; i++){
        printf ("Donner un élément :");
        scanf ("%d", & tab[i]);
    }
    printf ("Donner une valeur à rechercher :");
    scanf ("%d", & valeur) ;
    trouve = 0 ;
    i = 0 ;
    while ( i < 10 && trouve == 0) {
```

```

    if ( valeur == tab[ i ]){
        trouve = 1;
    }
    else {
        i =i + 1;
    }
}
if (trouve == 0){
    printf ("La valeur n'existe pas dans le tableau");
}
else {
    printf ("la valeur existe dans le tableau à la case numéro %d ", i);
}
return 0;
}

```

d- Série d'exercices

Exercice 1: Écrire un Algo, C et PHP qui permet de stocker dans un tableau dix entiers et affiche tous les nombres pairs puis tous les nombres impairs.

Algo : exo1

Déclaration

tab : tableau[1..10] de entier

i : entier

Début

pour i allant 1 à 10 faire

 afficher ("Donner un élément :")

 saisir (tab[i])

fin pour

afficher ("Voici la liste des nombres pairs :")

pour i allant 1 à 10 faire

 si tab[i] % 2 =0

 alors afficher ("Nombre pair : " , tab[i])

 finsi

finpour

afficher ("Voici la liste des nombres impairs :")

pour i allant 1 à 10 faire

 si tab[i] % 2 = 1

 alors afficher ("Nombre impair : " , tab[i])

 finsi

finpour

fin exo1

Traduction en C :

```
#include <stdio.h>
```

```
int main (){
```

```

int tab[10];
int i;
for (i=0; i<10; i++){
    printf("Donner une valeur : ");
    scanf("%d", &tab[i]);
}
printf("Voici la liste des nombres pairs :");
for (i=0; i<10; i++){
    if (tab[i]%2==0){
        printf(" \n Nombre pair : %d", tab[i]);
    }
}
printf("Voici la liste des nombres impairs :");
for (i=0; i<10; i++){
    if (tab[i]%2==1){
        printf(" \n Nombre impair : %d", tab[i]);
    }
}
return 0;
}

```

Traduction en PHP :

```

<!DOCTYPE html>
<html>
    <head>
        <title>Exo Tableau</title>
    </head>
    <body>
        <center>
            <h1> Exercice 1 : Pairs / Impairs</h1>
            <form method="post" action="">
                Liste des nombres (;) : <input type="text" name="tab">
                <input type="submit" name="Afficher" value="Afficher">
            </form>
            <?php
                if (isset($_POST['Afficher'])){
                    $tab = array();
                    $tab = explode(";", $_POST['tab']);
                    echo "<br/>Voici la liste des nombres pairs :";
                    for ($i=0; $i<count($tab); $i++){
                        if ($tab[$i]%2==0){
                            printf("<br/>Nombre pair : %d", $tab[$i]);
                        }
                    }
                    printf(" <br/>Voici la liste des nombres impairs :");
                    for ($i=0; $i<count($tab); $i++){
                        if ($tab[$i]%2==1){
                            printf("<br/>Nombre impair : %d", $tab[$i]);
                        }
                    }
                }
            }
        </center>
    </body>
</html>

```

```
    }  
    ?>  
</center>  
</body>  
</html>
```

Exercice 2 : Écrire un Algo, C et PHP qui permet de remplir un tableau de dix entiers, de demander une valeur et d'afficher le nombre d'occurrences de la valeur dans le tableau.

Algo : occurrences

Déclaration

tab : tableau [1..10] de entier
i , valeur, compteur : entier

Début

```
pour i allant 1 à 10 faire  
    afficher ("Donner un élément :")  
    saisir (tab[i])  
fin pour  
afficher ("Donner une valeur à rechercher :")  
saisir (valeur)  
compteur ← 0  
pour i allant 1 à 10 faire  
    si valeur = tab[i]  
        alors compteur ← compteur + 1  
    fin si  
fin pour  
afficher("La valeur existe ", compteur, " fois ")  
fin occurrences
```

Traduction en C :

```
#include <stdio.h>  
  
int main () {  
    int tab[10];  
    int i, valeur, compteur ;  
    for (i=0; i<10; i++){  
        printf("Donner une valeur : ");  
        scanf("%d", &tab[i]);  
    }  
    printf("Donner la valeur à rechercher :");  
    scanf("%d", &valeur);  
    compteur = 0 ;  
    for (i=0; i<10; i++){  
        if (valeur == tab[ i ] ){  
            compteur = compteur + 1 ;  
        }  
    }  
    printf("La valeur apparaît %d fois ", compteur);  
}
```

```
return 0;
}
```

Traduction en PHP :

```
<!DOCTYPE html>
<html>
  <head>
    <title>Occurences</title>
  </head>
  <body>
    <center>
      <h1> Exercice 2 : Occurences</h1>
      <form method="post" action="">
        Liste des nombres (;) : <input type="text" name="tab"> <br/>
        Valeur à rechercher : <input type="text" name="valeur"> <br/>
        <input type="submit" name="Afficher" value="Afficher">
      </form>
      <?php
        if (isset($_POST['Afficher'])) {
          $tab = array();
          $tab = explode(";", $_POST['tab']);
          $valeur = $_POST['valeur'];
          $compteur = 0;
          for ($i = 0; $i < count($tab); $i++) {
            if ($valeur == $tab[$i]) {
              $compteur ++ ;
            }
          }
          printf("La valeur apparait %d fois", $compteur);
        }
      ?>
    </center>
  </body>
</html>
```

Exercice 3 : Écrire un algo, C, PHP qui permet de stocker dans un tableau un nombre de prix et afficher le prix moyen, le prix max et le prix min.

Algo : prix

Déclaration

tab : tableau [1..10] de réel
i : entier
max, min, moyen : réel

Début

pour i allant 1 à 10 faire
 afficher ("Donner un prix :")
 saisir (tab[i])
fin pour

```

moyen ← 0
min ← tab[1]
max ← tab[1]
pour i allant 1 à 10 faire
    si tab[i] < min
        alors min ← tab[i]
    fin si
    si tab[i] > max
        alors max ← tab[i]
    fin si
    moyen ← tab[i] + moyen
fin pour
moyen ← moyen / 10
afficher ("Le prix moyen est : ", moyen)
afficher ("Le prix max est : ", max)
afficher ("Le prix min est : ", min)
fin prix

```

Traduction en C :

```

#include <stdio.h>

int main (){
    float tab[10];
    int i;
    float min, max, moy;
    for (i=0; i<10; i++){
        printf("Donner un prix: ");
        scanf("%d", &tab[i]);
    }
    moy=0;
    min=tab[0];
    max=tab[0];
    for(i=0; i<10; i++){
        if(tab[i]<min){
            min=tab[i];
        }
        if(tab[i]>max){
            max=tab[i];
        }
        moy=moy+tab[i];
    }
    moy=moy/10;
    printf("Le prix moyen est: %f",moy);
    printf("Le prix min est: %f", min);
    printf("Le prix max est: %f", max);
    return 0;
}

```

Traduction en PHP :


```

<!DOCTYPE html>
<html>
  <head>
    <title>Prix</title>
  </head>
  <body>
    <center>
      <h1> Exercice 3 : Prix</h1>
      <form method="post" action="">
        Liste des prix (;) : <input type="text" name="tab"> <br/>
        <input type="submit" name="Afficher" value="Afficher">
      </form>
      <?php
        if (isset($_POST['Afficher'])) {
          $tab = array();
          $tab = explode(";", $_POST['tab']);
          $moy = 0;
          $min=$tab[0];
          $max=$tab[0];
          for ($i = 0; $i < count($tab); $i++) {
            if ($tab[$i] < $min){
              $min=$tab[$i];
            }
            if ($tab[$i] > $max){
              $max=$tab[$i];
            }
            $moy=$moy+$tab[$i];
          }
          $moy=$moy/count($tab);
          printf("Le prix moyen est: %f", $moy);
          printf("<br> Le prix min est: %f", $min);
          printf("<br> Le prix max est: %f", $max);
        }
      ?>
    </center>
  </body>
</html>

```