Les Tableaux : l'essentiel

a) Définition :

Un tableau (ou vecteur) est un groupe de plusieurs variables de même types (groupe de données). Dans lequel, chaque variable est repérée par un numéro appelé index. La plupart des langages commencent à numéroter à partir de 0.

NB : il existe d'autres façons de regrouper des variables de même type :

- liste (ou collection) : FIFO, FILO, LILO, FINO,...
- arbre,
- graphe,
- fichier,...).

b) Déclaration : (dans ces exemples N est une constante)

- en langage algorithmique :

Tableau Entier **composé de** N **entier; // entier est un tableau regroupant N entier**

- En C : **int** tEntier **[**N**];** // tEntier est un pointeur sur un entier
- En C# : int [] tEntier = new int[N];

le tableau « tEntier » est une référence sur un objet de base abstrait « Array »

Tous ces exemples déclarent des tableaux regroupant N entiers numérotés de 0 à N-1.

c) Accès au tableau : on donne le nom du tableau et l'index de la variable à laquelle on veut accéder souvent entre crochets.

Si on veux affecter la valeur 0 à la position 5 d'un tableau.

- En algo : tEntier[5] <- 0;
- En C : tEntier[5] = 0; ou *(tEntier + 5) = 0;
- En C# : tEntier[5] = 0;

d) Passage d'un tableau en paramètre d'une fonction :

- Passage par valeur : rarement possible (possible en Pascal). Pas en C (car tableau est un pointeur), ni en C# car le passage par valeur est interdit.

Les éléments du tableau ne peuvent pas être modifié.

- Passage par référence : dans tout les langages

Les éléments du tableau peuvent être modifié. (référence ressemble à une adresse)

- Passage par pointeur : en C , C++ car un tableau est un pointeur.

Les éléments du tableau peuvent être modifié.

- Passage pour modifier la référence sur un tableau :

La référence du tableau peut-être modifiée.
En C il faut utiliser un pointeur sur un pointeur càd un pointeur sur un tableau.
En C# : on utilise « ref » (le tableau doit-être initialisé avant l'appel) ou « out » (le tableau n'a pas besoin d'être initialisé avant l'appel).

e) Initialisation : Elle n'est pas toujours possible lors de la déclaration.

- En C : int jours[7] = { 1, 1, 1, 1, 1, 1, 1 };
- En C# : int [] tEntier = new int[] {10, 15, 20, 25, 30, 35};

f) Exploration d'un tableau (càd le fait de parcourir tous les éléments d'un groupe de donnée):

- En algo : initialisation de tout les éléments du tableau.

```
i : entier;
pour tout i index du tableau t faire
    t[i] <- 0;
```

- en C, C++ :

```
for (int i = 0; i<N; i++)        // exploration du tableau
    t[i] = 0;
```

- en C# :

```
foreach ( int Elt in t)        // exploration du tableau
        Console.Write(" {0:d4}", Elt);

                ou

for (int i = 0; i < t.Length; i++)        // exploration du tableau
            t[i] = 0;
```

g) Tableau à plusieurs dimensions :

Il s'agit d'un tableau de tableau.

Exemple :

- Algo : Tableau MotsCroisés composé de 10 Tableau Ligne composé de 10 carcatères;
  ou MotsCroisés : Car[10][10];
- C : char MotCroises[10][10];
- C# : char [,] MotsCroise = new char[3,3];

Autre exemple : on peut définir une matrice 3 D comme un tableau à 2 dimensions de 3 Réels par 3 Réels;

En Algo : Matrice : tableau de 3 Tableau de 3 entier;

NB : si on veut implémenter un type abstrait (objet) Matrice avec toutes les opérations y afférentes il vaut mieux créer une classe Matrice.

h) Parcours d'un tableau à plusieurs dimensions : en général en 2 D, on utilise 2 variables d'indexation, une pour explorer les lignes une pour explorer les colonnes.

- En Algo :

```
Pour tout Eléments du tableau faire
        Ecrire(  Elément);
```

- En C# :

```
for (int i = 0; i<t.GetLength(0); i++)        // exploration des lignes
        {for (int j = 0; j < t.GetLength(1); j++)        // exploration des colonnes
            Console.Write(" " + t[i, j ]);  // attention au formatage des données en sortie
        Console.Write(Environment.NewLine);
        }
```