LES TABLEAUX

1. Introduction

Les variables que nous avons vues sont jusqu'à présent élémentaires. Elles ne contiennent qu'une seule valeur d'un type simple.
Cependant, il arrive que nous soyons obligés de traiter plusieurs données de même type appartenant à la même entité .

Exemple :

Imaginons que dans un programme, nous ayons besoin simultanément de 10 valeurs (par exemple, des notes pour calculer une moyenne). Evidemment, la seule solution dont nous disposons à l'heure actuelle consiste à déclarer 10 variables (N1, N2, …N10, etc). Mais cela ne change pas fondamentalement notre problème, car arrivé au calcul, cela donnera obligatoirement:

Moy = (N1 + N2 + N3 + N4 + N5 + N6 + N7 + N8 + N9 + N10) / 10

C'est tout de même ennuyeux, surtout si nous sommes face à un programme de gestion avec quelques centaines ou quelques milliers de valeurs à traiter.
Heureusement, la programmation nous permet de rassembler toutes ces variables en une seule, appelée **tableau (vecteur).**

2. Définitions

- Un vecteur (tableau unidimensionnel) est une variable indicée permettant de stocker n valeurs de même type. Le nombre maximal d'éléments, précisé à la déclaration, s'appelle la capacité du tableau. Le type du tableau est le type de ses éléments.
- La position d'un élément s'appelle indice ou rang de l'élément. Un tableau possède un ensemble d'indices. A chaque valeur de l'indice ne correspond qu'une et une seule case du tableau, donc un élément.

Remarque :

La définition indique que :

- Tous les éléments d'un tableau portent le même nom (celui du tableau).
- Tous les éléments d'un tableau ont le même type ; on parlera d'un tableau d'entiers, d'un tableau de caractères, etc.
- Un tableau peut ne pas être entièrement rempli mais il ne pourra jamais contenir plus d'éléments que le nombre prévu lors de la déclaration.
- L'indice est forcément un entier positif

3. Déclaration d'un tableau et Accès aux éléments

3.1. Déclaration :

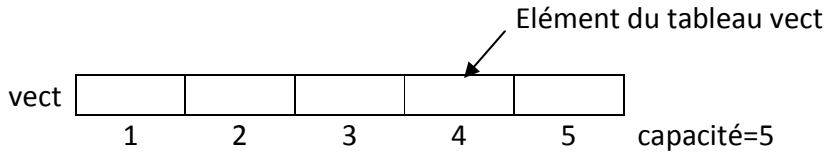
Syntaxe

nomVariable : tableau [1.. capacité] de Type

Où :

nomVariable : indique le nom du tableau
tableau : indique que c'est un tableau (et non pas une variable simple)
capacité : indique la capacité du tableau.

Type : indique le type des éléments du tableau.



Vect est un tableau de 5 éléments de type entier

Exemples de déclarations:

Variable
Vect : tableau [1..5] de entier
T1 : tableau [1..10] de caractère (T1 est un tableau de 10 cases de type caractère)
Tab : tableau [1..100] de réel (Tab est un tableau de 100 cases de type réel)
En langage Pascal, les déclarations précédentes sont traduites comme suit :
Vect : array [1..5] of integer ;
T1 : array [1..10] of char ;
Tab ; array [1..100] of real ;

3.2. Accès aux éléments (Accès indiciel)

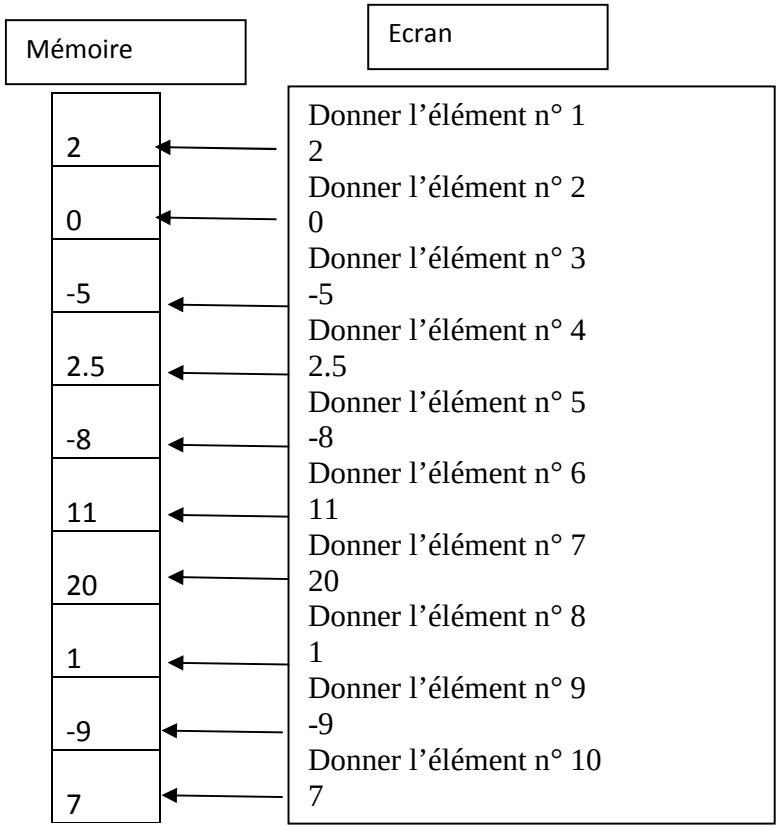
Pour repérer un élément parmi les autres, on utilise un indice qui représente un nombre entier qui permet d'accéder à un élément. Pour lire un élément ou le modifier, on indique la valeur de son indice.
Soit **vect** un tableau de capacité 5 de type réel, déclaré comme suit :
Vect : tableau [1..5] de réel.
Et soit a une variable de type réel et b une variable de type entier.
L'instruction a ←vect[1] exprime que la variable a reçoit la valeur de l'élément n°1
vect[5] ←34.56 exprime que l'élément n°5 du tableau vect reçoit la valeur 34.56
b ←3
a ←vect[b] exprime que a reçoit la valeur de l'élément n° 3

Remarque :
Attention au débordement : dans un tableau de capacité n, un indice i doit toujours être compris entre 1 et n ($1 \leq i \leq n$).

3.3. La lecture des éléments d'un tableau

Pour remplir la ***ième*** case d'un tableau ***vect*** par une valeur saisie par l'utilisateur, il faut utiliser l'instruction lire comme suit :
Lire(vect[i])
Pour remplir tous les éléments d'un tableau, il est nécessaire d'utiliser une boucle. Puisque la capacité du tableau est connue, alors la boucle « POUR » est la plus appropriée pour ce traitement mais rien n'interdit d'utiliser les autres boucles.
Soit **vect** un tableau de 10 éléments de type réel. L'algorithme qui permet de remplir ce tableau est le suivant :

```
Algorithme lecture_tableau ;
Variable
i :entier ;
vect : tableau[1..10] de réel ;
Début
pour i de 1 à 10 pas 1 faire
  écrire ('Donner l''élément n° ',i) ;
  lire (vect[i]) ;
finfaire
fin
```



3.4. Affichage des éléments d'un tableau

Pour afficher la ***ième*** case d'un tableau ***vect***, il faut utiliser l'instruction écrire comme suit :

Ecrire(vect[i])

Pour afficher tous les éléments d'un tableau, il est nécessaire d'utiliser une boucle. Puisque la dimension du tableau (le nombre d'éléments du tableau) est connue, alors la boucle « POUR » est la plus appropriée pour ce traitement mais rien n'interdit d'utiliser les autres boucles.

Soit vect un tableau de 10 éléments de type réel.

Vect

2	0	-5	2.5	-8	11	20	1	-9	7

L'algorithme qui permet d'afficher les éléments ce tableau est le suivant :

```
Algorithme affichage_tableau ;
Variable
i :entier ;
vect : tableau[1..10] de réel ;
début
pour i de 1 à 10 pas 1 faire
  écrire ('l''élément n° ', i , '=',vect[i]) ;
finfaire
fin
```

Ecran

```
l'élément n° 1=2
l'élément n° 2=0
l'élément n° 3=-5
l'élément n° 4=2.5
l'élément n° 5=-8
l'élément n° 6=11
l'élément n° 7=20
l'élément n° 8=1
l'élément n° 9=-9
l'élément n° 10=7
```

Exercice d'application

Ecrire un algorithme qui permet de :

1- Remplir un tableau « temp » par les températures quotidiennes d'une semaine.
2- Calculer et afficher la moyenne des températures.

```
Algorithme temperature ;
Variable
i :entier ;
temp : tableau [1..7] de réel ;
m,s : réel ;
début
pour i de 1 à 7 pas 1 faire
  écrire('Donner la température du jour n°', i) ;
  lire (temp[i]) ;
fin faire
s ←0 ;
pour i de 1 à 7 pas 1 faire
 s←s+temp[i] ;
fin faire
m←s/7;
écrire('La moyenne de la semaine=', m);
fin.
```