

Algorithmes Séance 6

Les tableaux

Définition

Un tableau est une structure de données. Il est utilisé pour stocker plusieurs valeurs de même type. Les éléments d'un tableau sont accessibles grâce à leurs indices, commençant du zéro.

Exemple

Soit le tableau suivant :

tn	Amine	Israa	Sirine	Eya
	0	1	2	3

1. Quel est le nombre d'éléments du tableau **tn** ?

4 éléments

2. Quel est l'indice du premier élément ? Et quel est l'indice du dernier élément ?

Le premier élément possède l'indice 0 et le dernier élément possède l'indice 3.

3. Est-ce qu'on peut insérer des valeurs numériques (de type entier ou réel) dans le tableau **tn** ?

Tous les éléments d'un tableau doivent avoir le même type. Bien qu'en Python les listes peuvent des valeurs de n'importe quel type.

Déclaration

La déclaration d'un tableau se fait au niveau du T.D.O. de la façon suivante :

Objet	Type
...	...
nom_tableau	tableau de nb_éléments type
...	...

Exemple

Pour déclarer le tableau **tn** utilisé précédemment on écrit :

Objet	Type
tn	tableau de 4 chaîne

Activité 1

Dessiner le TDO relatif aux tableaux suivants :

moy	9.75	16.5	14.25	12.75	6.5
	0	1	2	3	4

Tableau **moy** contenant les moyennes, des réels, de 5 matières (0 : Arabe, 1 : Mathématiques, 2 : Techniques).

pieces	20	30	25	55	40	...	88
	0	1	2	3	4	...	29

Tableau **pieces** qui indique le nombres de pièces vendues par un marchand, pendant un mois.

absences	Faux	Faux	Vrai	Faux	Vrai	Faux
	0	1	2	3	4	5

Tableau **absences** qui indique les jours d'absences d'un élève pendant une semaine (0 : Lundi, 1 : Mardi, etc.). Vrai signifie que l'élève est absent.

Remplissage

Un tableau peut être rempli, explicitement, case par case par des constantes.

Pour remplir le tableau **tn** on peut écrire :

Exemple

```
tn[0] ← "Amine"  
tn[1] ← "Israa"  
tn[2] ← "Sirine"  
tn[3] ← "Eya"
```

Un tableau peut être, aussi, rempli par des valeurs saisies par l'utilisateur.

Exemple

```
Pour i de 0 à 3 Faire  
  Ecrire("Donner le nom de la ", i, "ème personne ? ")  
  Lire(tn[i])  
Fin Pour
```

Le tableau peut, encore, être rempli d'une façon aléatoire.

Exemple

Le code suivant remplit le tableau `tn` par des mots de passes composés de chaînes aléatoires.

```
Pour i de 0 à 3 Faire  
  tn[i] ← ""  
  Pour j de 0 à 5 Faire  
    tn[i] ← tn[i] + chr(aléa(48, 90))  
  Fin Pour  
  Ecrire("Mot de passe", i, ":", tn[i])  
Fin Pour
```

Affichage

On veut souvent afficher le contenu d'un tableau. L'opération se fait de la façon suivante :

Exemple

Afficher les noms de tous les participants d'un Tombola :

```
Pour i de 0 à 3 Faire  
  Ecrire("Participant", i, ":", tn[i])  
Fin Pour
```

D'autres fois, on veut juste afficher quelques éléments du tableau. Par exemple ceux qui vérifient une condition.

Exemple

Afficher les noms des participants d'indices impaires :

```
Pour i de 0 à 3 Faire  
  Si (i mod 2 = 1) Alors  
    Ecrire("Participant", i, ":", tn[i])  
  Fin Si  
Fin Pour
```

Activités

Parking Vertical

Dans les villes on préfère les parking verticaux parce qu'il permettent de gagner de l'espace.

Un parking possède une capacité comptée en nombre de places n , avec $4 \leq n \leq 200$

En supposant qu'un parking peut être représenté par un tableau qui contient les états suivants :

- 0 Emplacement vide
- 1 Emplacement occupé
- 2 Emplacement réservé

Travail demandé

On demande d'écrire un programme qui :

1. Remplit le parking.
2. Calcule le nombre d'étages occupés.
3. Déterminer l'indice du premier étage vide.
4. Insère un nouveau véhicule.
5. Affiche l'état du parking.

Solution

Algorithme Parking

Début

```
// Question 1
```

Répéter

```
Ecrire("Nbre d'étages du parking ( $4 \leq n \leq 200$ ) ? ")
```

```
Lire(n)
```

```
Jusqu'à ( $4 \leq n \leq 200$ )
```

```
Pour i de 1 à n Faire
```

Répéter

```
Ecrire("L'état de l'étage n°", i, " ? ")
```

```
Lire(park[i])
```

```
Jusqu'à ( $0 \leq \text{park}[i] \leq 2$ )
```

```
Fin Pour
```

```
// Question 2
```

```
occ ← 0
```

```
Pour i de 1 à n Faire
```

```
Si park[i] ≠ 0 Alors
```

```
occ ← occ + 1
```

```
Fin Si
```

```
Fin Pour
```

```
Ecrire("Il y'a", occ, "places occupées et",  
n-occ, "places libres")
```

```
// Question 3
```

```
p ← -1
```

```
i ← 1
```

```
TantQue (p = -1) et (i ≤ n) Faire
```

```
Si park[i] = 0 Alors
```

```
p ← i
```

```
Sinon
```

```
i ← i + 1
```

```
Fin Si
```

```
Fin TantQue
```

Objet	Type
park	tableau de 201
n, i, occ,	entier
p	entier

```

Si p ≠ -1 Alors
  Ecrire("L'étage", p, "est vide")
Sinon
  Ecrire("Aucun étage vide")
Fin Si
// Question 4
Si p ≠ -1 Alors
  park[0] ← 1
  Pour i de p-1 à 0 Faire
    park[i+1] ← park[i]
  Fin Pour
  park[0] ← 0
Fin Si
// Question 5
Pour i de 1 à n Faire
  Si park[i] = 0 Alors etat ← "vide"
  Sinon Si park[i] = 1 Alors etat ← "occupé"
  Sinon etat ← "réservé"
  Fin Si
  Ecrire("L'étage", i, "est", etat)
Fin Pour
Fin

```

Activité 2

Ecrire un programme qui saisit un tableau de n entiers non nuls ($5 \leq n \leq 20$) puis supprime les duplications.

Solution

```

Algorithme Activité2
Début
  Répéter
    Ecrire("Donner n ( $5 \leq n \leq 20$ ) ? ")
    Lire(n)
  Jusqu'à ( $5 \leq n \leq 20$ )
  Pour i de 1 à n Faire
    Répéter
      Ecrire("t[", i, "] ? ")
      Lire(t[i])
    Jusqu'à (t[i] ≠ 0)
  Fin Pour
  // Remplacer les duplications par des 0
  Pour i de 1 à n-1 Faire
    p ← -1
    j ← i-1
    TantQue (p = -1) et (j ≥ 0) Faire
      Si t[i] = t[j] Alors
        t[i] ← 0
        p ← j
      Sinon
        j ← j - 1
      Fin Si
    Fin TantQue
  Fin Pour

```

Objet	Type
t	tableau de 20 entier
n, i, j, p	entier

```

// Supprimer les cases vides
j ← -1
Pour i de 0 à n-1 Faire
    Si t[i] ≠ 0 Alors
        j ← j + 1
        t[j] ← t[i]
    Fin Si
Fin Pour
// Afficher le nouveau tableau sans duplications
n ← j + 1
Pour i de 0 à n-1 Faire
    Ecrire(t[i], ", ")
Fin Pour
Fin

```

Tableau Symétrique

Partie 1

Le tableau [1,2,3,2,1] est symétrique alors que le tableau [1,2,3,1,2] n'est pas symétrique.

Ecrire un programme qui saisit un tableau de n entiers ($5 \leq n \leq 20$) puis détermine si le tableau est symétrique ou non.

Partie 2

Le tableau [1,2,3,8,2,1] n'est pas symétrique en supprimant la valeur 8 il devient symétrique.

Ecrire un programme qui détermine si le tableau saisit précédemment peut devenir symétrique en supprimant exactement un seul de ses éléments.

Top 5 Scores

Hachem a conçu un clone du jeu snake. Il veut maintenir un Top 5 des scores des joueurs. Pour cela on veut écrire un programme qui :

- Saisit le **score** d'un joueur, $\text{score} \geq 0$.
- Saisit le **nom** d'un joueur, composé uniquement de lettres majuscules.
- Insère les données dans deux tableaux **sj** et **nj**, le premier est utilisé pour stocker les scores des joueurs et le second est utilisé pour stocker leurs noms.
- Répète les opérations précédentes tant que le score est positif.
- Affiche le Top 5 Scores : Les noms des joueurs ainsi que leurs scores.

Montagne la plus culminante

Un chercheur veut déterminer les deux montagnes les plus hautes dans un relief montagneux. Pour cela il a pris des mesures de l'hauteur de la croute terrestre sur une ligne de n points, $3 \leq n \leq 15$ dans un tableau **haut**.

Ecrire un programme qui :

- Saisit les hauteurs de n points du relief étudié dans un tableau **haut**
- Détermine le point le plus haut de ce relief montagneux.
- Détermine le 2^{ème} point le plus haut.