

8 - Tableaux



I – Qu'est-ce qu' un tableau en informatique ?

.....

.....

.....

.....

.....

Ad 200 45	Ad 201 200	Ad 202 21	Ad 203 75	Ad 204 0	Ad 205 0	Ad 206 0	Ad 207 0	Ad 208 28	Ad209 255
Ad 210 200	Ad 211 150	Ad 212 0	Ad 213 18	Ad 214 16	Ad 215 42	Ad 216 72	Ad 217 250	Ad 218 250	Ad219 72
Ad 220 90	Ad 221 253	Ad 222 2	Ad 223 78	Ad 224 95	Ad 225 27	Ad 226 201	Ad 227 99	Ad 228 6	Ad229 1
Ad 230 157	Ad 231 1	Ad 232 254	Ad 233 18	Ad 234 9	Ad 235 73	Ad 236 65	Ad 237 143	Ad 238 218	Ad239 0

Exercice 01 : Que vaut l'octet stocké à l'adresse 213 ? Convertir ce nombre N_{10} en base 2 de façon à visualiser les 8 bits réellement stockés à cet endroit.

Remarque (02) : Peut-on savoir ce que cet octet encode ?

.....

.....

.....

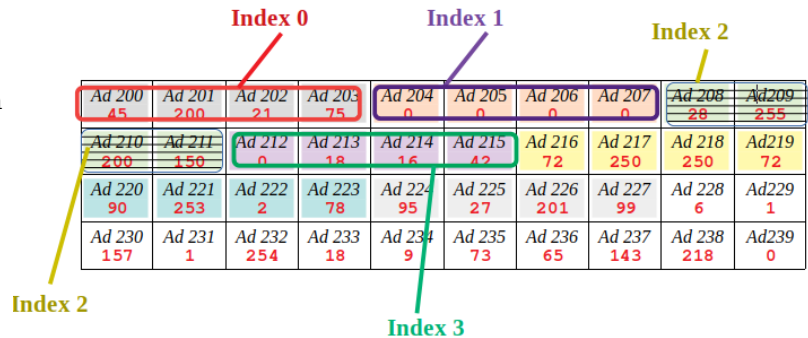
Exercice de cours : Le tableau est un tableau d'entiers encodés sur 4 octets.

Quels sont les 4 octets WXYZ de l'index 0 ?

Voici le contenu converti en binaire : 11001000 11001000 00010101 01001011

Trouver l'entier encodé dans ce premier emplacement mémoire.

Exercice 03 : Quelle est l'adresse de l'élément d'index 100 si on considère un tableau d'entiers sur 4 octets commençant à l'adresse 200 ?



Avantage des tableaux :

.....

.....

.....

Trois désavantages des tableaux :

.....

.....

.....

.....

.....

II – Tableaux en Python

1-Déclaration d'un tableau	<pre>>>> t = ['a', 'b', 'c']</pre>
2-Nombre d'éléments	<pre>>>> nbr = len(t) >>> nbr 3</pre>
3-Accès à l'un des éléments	<pre>>>> t[0] 'a'</pre>
4-Lecture avec boucle FOR d'index	<pre>>>> for i in range(len(t)) : print(t[index]) 'a' 'b' 'c'</pre>
5-Lecture avec boucle FOR nominative	<pre>>>> for e in t : print(e) 'a' 'b' 'c'</pre>
6-Modification d'un élément (le tableau est mutable)	<pre>>>> t[0] = 'z' >>> t ['z', 'b', 'c'] >>> for i in range(len(t)) : t[i] = 'toto' >>> t ['toto', 'toto', 'toto']</pre>

Exercice 04 : Donner les valeurs successives prises par **i** avec un tableau de 5 éléments si on utilise :

```
for i in range(len(t)):
```

ATTENTION : la variable contient la référence du tableau : on peut donc modifier un tableau depuis une fonction. On dit qu'on effectue une modification par **EFFET DE BORD**.

7-Effet de bord :	<pre>def multiplier(t, coef) : '''exemple >>> tab_test = [1,2,10] >>> multiplier(tab_test, 60) >>> tab_test [60, 120, 600] ''' for i in range(len(t)): t[i] = t[i] * coef</pre>
-------------------	--

Exercice 05 : En vous aidant du prototype fourni sur le site, écrire la fonction-pure **compter(t:list)** qui renvoie le nombre d'entier supérieur à 10 dans le paramètre-tableau t.

Exercice 06 : En vous aidant du prototype fourni sur le site, écrire la fonction-procédure **modifier(t:list)** qui modifie t par effet de bord en remplaçant ses éléments pour 0 si l'élément est inférieur à 10, ou 20 si l'élément est au moins égal à 10.

III – Tableau créé en compréhension

On peut créer un tableau **par extension** en lui écrivant les éléments un à un : `t = ['a', 'b', 'c']`

On peut également créer un tableau **par compréhension** en utilisant une boucle FOR.

Valeur constante mais autant d'éléments qu'un tableau initial	<pre>>>> t1 = [1, 2, 3] >>> s = [None for v in t1] >>> s [None, None, None]</pre>
Calcul à partir d'un tableau initial	<pre>>>> t = [1, 2, 3] >>> s = [v*2 for v in t1] >>> s [2, 4, 6]</pre>
Valeur constante, nombre d'éléments défini par un range	<pre>>>> t = [0 for v in range(10)] >>> t [0, 0, 0, 0, 0, 0, 0, 0, 0, 0]</pre>
Valeur basée sur l'index de la case	<pre>>>> t = [v*10 for v in range(10)] >>> t [0, 10, 20, 30, 40, 50, 60, 70, 80, 90]</pre>
Valeur basée sur une fonction	<pre>>>> import random >>> t = [random.randint(1,100) for v in range(10)] >>> t [42, 41, 96, 35, 58, 94, 16, 21, 84, 51]</pre>
Utilisation d'une condition	<pre>>>> import random >>> t = [random.randint(0,20) for v in range(5)] >>> t2 = [note for note in t if note >= 10] >>> t [12, 8, 1, 3, 3] >>> ta2 [12]</pre>

Exercice 07 : Créer par compréhension un tableau de 20 éléments contenant de nombres aléatoires compris entre 20 et 50.

Exercice 08 : Créer par compréhension un tableau de 500 éléments contenant 0.

Utilisation d'une fonction	<pre>>>> import random >>> t = [f(x) for x in range(100)]</pre>
----------------------------	---

IV – Tableaux pour générer des graphiques

Le programme est fourni sur la page suivante

Exercice 10 : Que valent les valeurs minimale et maximale lorsqu'on utilise `for v in range(1000)` ?

Exercice 11 : Que valent les quatre premières valeurs et la valeur maximale du tableau suivant lorsqu'on utilise la création par compréhension suivante ?

```
tx = [ v/10 for v in range(1000) ]
```

Exercice 12 : Que valent les 4 premières valeurs du tableau suivant :

```
def f(x):
    return math.sqrt(x)
ty = [ f(x) for x in tx ]
```

Programme :

```
01 # 1 - Importation des modules nécessaires
02 import math
03 import matplotlib.pyplot as plt
04
05 # 2 - Constantes
06 # 3 - Déclaration des fonctions
07 # 4 - Programme principal
08
09 # Création des tableaux de données
10 tx = [ v/1000 for v in range(2000) ]
11 ty1 = [ math.sqrt(x) for x in tx ]
12 ty2 = [ x for x in tx ]
13 ty3 = [ x**2 for x in tx ]
14
15 xmax = max(tx)
16 ymax = max(ty3)
17
18 # Création des courbes
19 plt.plot(tx, ty1, label="f(x)", color='orange')
20 plt.plot(tx, ty2, label="g(x)", color='red')
21 plt.plot(tx, ty3, label="h(x)", color='green')
22 plt.xlabel("x")
23 plt.ylabel("Valeurs en ordonnée")
24 plt.title("Représentation des fonctions")
25 plt.legend(loc='upper center')
26 plt.axis([0, xmax, 0, ymax])
27 plt.grid()
28 plt.show() # Affichage à l'écran des courbes
```

V – Tableaux dans Javascript : pour informations, voir le site si vous en avez besoin

VI – Méthode append pour la création par extension et ajouts

Création d'un tableau
[0, 10, 20, ... jusqu'à 90]

```
t = []
for i in range(10) :
    t[i].append(i*10)
```

Attention : il ne faut PAS écrire de code de type **t = t.append(12)**. Avec ce code, tableau va contenir la réponse de append, à savoir None !

Tableau statique vs tableau dynamique :

→ Un **tableau statique** est un tableau qui possède un nombre de cases fixé à la création.

→ Un **tableau dynamique** est un tableau dont le nombre de cases peut varier. C'est le cas lorsqu'on utilise append. Attention, les tableaux dynamiques n'existent pas dans tous les langages et le coût des rajouts ou des suppressions n'est pas constant. Les utiliser est donc pratique mais cela rend vos algorithmiques moins portables d'un langage à l'autre et rend l'estimation du coût assez difficile.