

Python

Les tableaux à 2 dimensions

TD 01

Série d'exercices en Python : exercices initiaux :

1) Rédiger les programmes permettant d'initialiser un tableau à 2 dimensions :

- Créez un tableau a 2 dimensions dont toute les cases sont initialisées à 0.

0	0	0
0	0	0
0	0	0

- Obtenir les résultats suivant en utilisant uniquement la structure de boucles conventionnelle vu au cours (2 « for » croissants)

for i in range(len(m)):

for j in range(len(m[i])):

a. Initialiser tout à 7

7	7	7
7	7	7
7	7	7

b. Initialiser Ligne : 123, Ligne :123, Ligne :123

1	2	3
1	2	3
1	2	3

c. Initialiser Ligne :111, Ligne :222, Ligne :333

1	1	1
2	2	2
3	3	3

d. Initialiser Ligne :123, Ligne :456, Ligne :789

1	2	3
4	5	6
7	8	9

e. Initialiser Ligne :987, Ligne :654, Ligne :321

9	8	7
6	5	4
3	2	1

- f. Initialiser Ligne :321, Ligne :654, Ligne :987

3	2	1
6	5	4
9	8	7

- g. Initialiser Ligne :123, Ligne :654, Ligne :789

1	2	3
6	5	4
7	8	9

- 2) Soit un tableau à 2 dimensions dont toute les cases sont initialisées à 9 :

9	9	9
9	9	9
9	9	9

- a. Initialiser la première ligne à 3,

3	3	3
9	9	9
9	9	9

- b. Initialiser la dernière colonne à 7

3	3	7
9	9	7
9	9	7

- c. Initialiser la diagonale à 0,1,2,...

0	3	7
9	1	7
9	9	2

- d. Initialiser l'autre diagonale à 5 (en commençant en bas à gauche).

0	3	5
9	5	7
5	9	2

- e. Initialiser l'autre diagonale à 10, 20,30,... (en commençant en haut à droite).

0	3	10
9	20	7
30	9	2

3) Soit le tableau à 2 dimensions suivant :

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

Rédiger les programmes permettant d'afficher ce tableau dans l'ordre :

- a. 4,3,2,1,8,7,6,5,12,11,10,9,16,15,14,13
 - b. 16,15,14,13,12,11,10,9,8,7,6,5,4,3,2,1
 - c. 1,5,9,13,2,6,10,14,3,7,11,15,4,8,12,16
 - d. 16,12,8,4,15,11,7,3,14,10,6,2,13,9,5,1
 - e. 1,2,3,4,9,10,11,12,5,6,7,8,13,14,15,16
 - f. 1,3,5,7,9,11,13,15
(2 possibilités : affichage des colonnes paires OU affichage du genre "damier")
 - g. 1,2,3,4,6,7,8,11,12,16
 - h. 13,14,10,15,11,7,16,12,8,4
 - i. 1,5,2,9,6,3,13,10,7,4,14,11,8,15,12,16 (très difficile à faire)
- 4) Rédiger un programme permettant d'afficher un tableau à 2 dimensions en utilisant seulement une seule boucle de type « for ».
- 5) Rédiger un programme permettant de remplir un tableau à 1 dimension de taille N*M depuis un tableau à 2 dimensions de N lignes et M colonnes avec et sans utiliser de variable supplémentaire (CPT).
- 6) Rédiger un programme permettant de remplir un tableau à 2 dimensions de N lignes et M colonnes depuis un tableau à 1 dimension de taille N*M. Fournir une solution avec 2 for et une avec 1 seul for.
- 7) Rédiger un programme qui remplit un tableau à 2 dimensions avec des valeurs aléatoires comprises en 0 et 20. Ensuite, on effectuera une recherche. On recherchera la valeur X (rencontré pour la première fois) dans ce tableau à 2 dimensions avec X (étant aussi un nombre aléatoire compris en 0 et 20). Si on trouve, on affiche la position (numéro de ligne et de colonne) de l'élément dans le tableau. On affiche « pas trouvé » si on n'a pas trouvé. La recherche dans le tableau doit s'arrêter dès que l'on rencontre le nombre recherché.

Série d'exercices en Python :

Exercice 1

Rédiger un programme qui détermine la $N^{\text{ème}}$ ligne du triangle de Pascal, avec $N = \text{randint}(3,8)$

Exemple :

```
1      1
1      2      1
1      3      3      1
1      4      6      4      1
1      5      10     10     5      1
```

Pour toutes les lignes (sauf la première), les éléments de la 2^{ème} colonne à l'avant dernière sont calculés en faisant la somme des éléments de la colonne précédente et de la même colonne de la ligne précédente.

➔ Pour le calcul de la ligne courante (n), on s'appuiera sur la ligne précédente

Exercice 2

Rédigez un programme qui détermine un nombre entier N par $\text{Randint}(1,6)$, puis construisez un tableau de N lignes et N^2 colonnes :

- les éléments de la dernière colonne sont tous égaux à N^3
- tout autre élément est obtenu en prenant l'élément de la même ligne mais de la colonne adjacente droite et en lui soustrayant 1 pour la ligne 1, 2 pour la ligne 2, 3 pour la ligne 3,

Une fois le tableau entièrement constitué, l'algorithme doit rechercher le nombre N^2 dans le tableau, ligne par ligne en commençant par la première ligne et dans chaque ligne en commençant par l'élément en dernière colonne. Si le nombre N^2 est trouvé, l'algorithme doit afficher dans quelle ligne et colonne il l'a été. Si N^2 ne se trouve pas dans le tableau, un message doit s'afficher pour l'indiquer. La recherche dans le tableau doit s'arrêter dès la rencontre d'un nombre égal à N^2 .

Exemple avec $N=3$

19	20	21	22	23	24	25	26	27
11	13	15	17	19	21	23	25	27
3	6	9	12	15	18	21	24	27

Exercice 3

Rédigez un programme qui détermine un nombre entier N par $\text{Randint}(1,8)$, puis construisez un tableau de N lignes et N^2 colonnes :

- les éléments de la dernière ligne sont tous égaux à N^3

- tout autre élément est obtenu en prenant l'élément de la même colonne mais de la ligne adjacente inférieure et en lui soustrayant 1 pour la colonne 1, 2 pour la colonne 2, 3 pour la colonne 3,

Une fois le tableau entièrement constitué, l'algorithme doit rechercher le nombre N^2 dans le tableau, colonne par colonne en commençant par la première colonne et dans chaque colonne en commençant par l'élément en dernière ligne. Si le nombre N^2 est trouvé, l'algorithme doit afficher dans quelle ligne et colonne il l'a été. Si N^2 ne se trouve pas dans le tableau, un message doit s'afficher pour l'indiquer. La recherche dans le tableau doit s'arrêter dès la rencontre d'un nombre égal à N^2 .

Exemple avec $N=3$

25	23	21	19	17	15	13	11	9
26	25	24	23	22	21	20	19	18
27	27	27	27	27	27	27	27	27

Exercice 4

Rédiger un programme qui détermine un nombre N entier impair aléatoire compris entre 3 et 19 (3 et 19 compris), puis construise le tableau carré $N \times N$

Exemple avec $N=9$:

	0	1	2	3	4	5	6	7	8
0	4	6	8	10	12	14	16	18	20
1	6	9	12	15	18	21	24	27	30
2	8	12	16	20	24	28	32	36	40
3	10	15	20	25	30	35	40	45	50
4	12	18	24	30	36	42	48	54	60
5	14	21	28	35	42	49	56	63	70
6	16	24	32	40	48	56	64	72	80
7	18	27	36	45	54	63	72	81	90
8	20	30	40	50	60	70	80	90	100

Chaque case du tableau est remplie avec la formule suivante (convention Python) :
 (Numéro de ligne + 2) * (Numéro de colonne + 2).

Une fois le tableau entièrement constitué, l'algorithme doit rechercher le nombre X dans le tableau. Avec X est un nombre entier positif aléatoire inférieur à N^2 . La recherche doit se faire uniquement de triangle supérieur (en jaune) situé au-dessus la diagonale principale du tableau (en orange) et doit s'arrêter dès que X a été trouvé.

Exercice 5

Rédigez un programme qui détermine un nombre N entier pair et (pseudo) aléatoire compris entre 2 et 20 puis remplit un tableau de N lignes et N colonnes de la manière suivante :

- les valeurs des éléments des 2 **lignes centrales** sont obtenues par $\text{Randint}(0,N)$
- la valeur de **tout autre élément** est obtenu :
 - s'il est au-dessus des lignes centrales, en sommant les valeurs des 2 éléments de la même colonne et des 2 lignes adjacentes en dessous de l'élément calculé ;
 - s'il est en dessous des lignes centrales, en multipliant les valeurs des 2 éléments de la même colonne et des 2 lignes adjacentes au-dessus de l'élément calculé.

Une fois le tableau entièrement constitué, l'algorithme doit parcourir le tableau à la recherche du nombre N^2 . La recherche doit commencer dans le coin inférieur droit du tableau et colonne par colonne. Lorsqu'un bord du tableau est atteint, la recherche se poursuit dans la colonne adjacente gauche de la même ligne. Dès que le nombre N^2 est trouvé, l'algorithme doit arrêter la recherche et afficher les numéros des ligne et colonne où N^2 a été trouvé. Si N^2 ne peut être trouvé, un message doit s'afficher pour l'indiquer.

Exemple

Avec $N=6$:

12	11	10	5	3	10
7	7	7	5	2	5
5	4	3	0	1	5
2	3	4	5	1	0
10	12	12	0	1	0
20	36	48	0	1	0

Le nombre 36 a été trouvé en ligne 6 et colonne 2.