

Formation Python



LES STRUCTURES DE DONNEES AVANCEES EN PYTHON

Goupillés par
Najah DAOUES
Inspecteur d'informatique

Table des matières

A/ Rappel des structures de données et des structures de contrôle	3
B/ Les structures de données avancées	5
B.1/ Les listes à deux dimensions (les tableaux à deux dimensions) :	5
⊙ Définition	5
⊙ Création : Vocabulaire et syntaxe	5
⊙ Accès aux éléments d'une liste à 2 dimensions.....	5
B.2/ Les dictionnaires (les enregistrements) :	7
⊙ Définition	7
⊙ Création d'un dictionnaire avec une clé et plusieurs valeurs.....	7
⊙ Accès aux éléments d'un dictionnaire	7
⊙ Création d'un tableau de dictionnaire.....	8
⊙ Manipulation d'un tableau d'enregistrements.....	8

A/ Rappel des structures de données et des structures de contrôle**Situation de problème 1 :**

Une société d'extraction de l'huile vierge d'olive (L'huilerie) dispose N boîtes de stockage d'olives. Les boîtes de stockage sont numérotées de 0 à N-1 avec ($2 \leq N \leq 50$). Après le pressage et l'extraction de l'huile, le responsable de qualité des produits a remarqué que l'huile extraite de certains boîtes n'est pas de qualité supérieure et cela peut influencer sur le prix de vente public. Après une étude faite sur les différentes étapes de production, il a remarqué que les olives restent emmagasinées beaucoup de temps avant leurs pressages. Il a conclu que pour avoir une huile de qualité supérieure, il faut que le temps de stockage soit le plus court possible c'est-à-dire que les premières olives récoltées doivent être les premières olives transformées.



Pour résoudre cette contrainte, le responsable de stockage a décidé d'écrire la date de récolte des olives sur chaque boîte et de stocker toutes les olives récoltées de la même date dans le même boîte et de fournir chaque jour aux ouvriers une feuille qui contient les numéros des cases ordonnées par ordre décroissant des dates de dépôt.

Exemple :

Boîtes :	18/10/2020	10/10/2020	04/11/2020	15/11/2020	14/11/2020
	0	1	2	3	4

Résultat :

L'ordre des boîtes :

1
0
2
4
3

Travail demandé :

1. Proposer une structure de données permettant de sauvegarder les dates de récolte des N boîtes d'olive.
2. En utilisant la structure de données choisie, écrire les algorithmes qui permettent d'aider le responsable de stockage dans son travail, puis de déduire les T.D.O. correspondant.
3. Implémenter les algorithmes précédant en langage Python.

Solution :

```
#procédure saisie qui permet la saisie de la valeur N avec 2<=N<=50
def saisie () :
    global N
    valid= False
    while valid==False :
        N = int (input("Saisir N entre 2 et 50"))
        valid = 2<=N<=50
"""procédure remplissage qui permet de remplir un tableau T par les dates"""
def remplissage ( N ) :
    global T
    T=[""]*50
```

```
for i in range(N) :
    T[i]= input("Saisir la date de la case N° "+ str(i))

""" une fonction qui retourne une date sous la forme aaaammjj en une valeur entière """
def conversion_date_nombre ( d ) :
    j=d[0:2]
    m=d[3:5]
    a=d[6:10]
    return int(a+m+j)

"""procédure comptage du calcul pour chaque date son ordre par rapport aux autres dates"""

def comptage ( T , N ) :
    global TR
    TR=[0]*N
    for i in range (N) :
        d1=conversion_date_nombre ( T[i] )
        R=0
        for j in range (N) :
            d2= conversion_date_nombre ( T[j] )
            if d1>d2 :
                R= R+1
        TR[R]=i

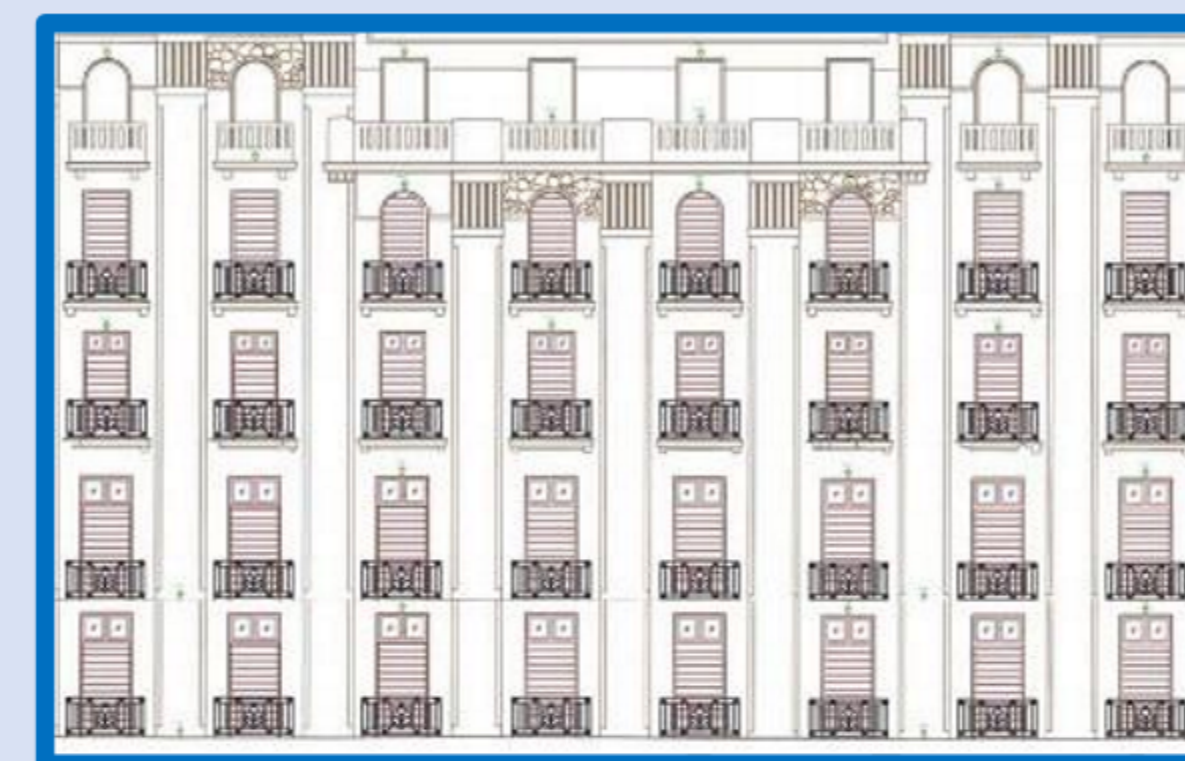
# procédure affichage
def affichage( TR , N ) :
    print("L'ordre des cases du jour" )
    for i in range( N ) :
        print (TR[i])

# Le programme principal
saisie ()
remplissage (N)
comptage ( T , N )
affichage( TR , N)
```

B/ Les structures de données avancées**Situation de problème 2 :**

Un nouvel hôtel est composé de N étages avec ($1 \leq N \leq 10$) et chaque étage est constitué de M chambres avec ($4 \leq M \leq 100$). Chaque chambre peut avoir l'un des états suivants :

- "D" : la chambre est disponible,
- "N" : la chambre est Non disponible, elle est louée par un client
- "H" : la chambre est Hors service.



Le directeur de cet hôtel décide d'acquérir un programme qui lui permet de :

- Connaître le nombre de chambres disponibles.
- Connaître le nombre de chambres non disponibles.
- Connaître le nombre de chambres hors services.
- Le pourcentage d'occupation de l'hôtel, sachant que :
 - pourcentage d'occupation = $\frac{\text{le nombre de chambres non disponibles}}{\text{Le nombre total de chambres}}$
- Les numéros des chambres hors services par étage.

Travail demandé :

1. Peut-on utiliser la structure de données tableau à une dimension pour stocker les informations des chambres de cet hôtel. Si oui, expliquer comment. Sinon proposer un autre type qui répond à ce besoin.
2. En utilisant la structure de données choisie, écrire l'algorithme qui permet d'aider le directeur de l'hôtel dans son travail, puis de déduire le T.D.O. correspondant.
3. Implémenter l'algorithme précédant en langage Python.

B.1/ Les listes à deux dimensions (tableaux à deux dimensions) :**⊙ Définition**

Une liste à 2 dimensions ou tableau à 2 dimensions est une structure de données tabulaire composée de lignes et de colonnes. L'accès à une cellule (représentant l'intersection d'une ligne et d'une colonne) s'effectue à partir de deux indices, l'un représentant l'accès ligne et l'autre représentant l'accès colonne.

⊙ Création : Vocabulaire et syntaxe

```
Nom_tableau = [
    [ Valeur1_Ligne0_Col0 , Valeur2_Ligne0_Col1 , Valeur3_Ligne0_Col2 ] ,
    [ Valeur1_Ligne1_Col0 , Valeur2_Ligne1_Col1 , Valeur3_Ligne1_Col2 ] ,
    [ Valeur1_Ligne2_Col0 , Valeur2_Ligne2_Col1 , Valeur3_Ligne2_Col2 ]
]
```

Exemples :

- ❖ Tab = [] c.a.d la liste est vide
- ❖ Tab = [[1 , 2 , 3 , 4] , [5 , 6 , 7 , 8] , [9 , 10 , 11 , 12]]

⊙ Accès aux éléments d'une liste à 2 dimensions

Pour accéder à un élément d'une liste, il suffit d'écrire le nom de la liste suivi de l'indice de la ligne entre deux crochets puis de l'indice de la colonne entre deux crochets :

Nom_Liste [indice_Ligne] [indice_Colonne]

Exemple : Tab [2] [3] = 12

⊙ Initialisation d'une liste à 2 dimensions par une valeur :

`Mat = [[valeur] * Nombre de colonnes for i in range (Nombre de lignes)]`

Solution :

```
def saisie () :
    global N, M
    valid=False
    while valid==False :
        N=int(input("Saisir le Nombre d'étage"))
        valid= 2<=N<=6
    valid=False
    while valid==False :
        M=int(input("Saisir le Nombre de chambre par étage"))
        valid= 2<=M<=60
def remplissage ( N , M ) :
    global Mat
    Mat=[[ "" ]*M for i in range (N)]
    for L in range ( N ) :
        for C in range ( M ) :
            valid= False
            while valid==False :
                Mat[L][C]= input("saisir l'état de la chambre N° "+ str(C)
                    +" qui existe dans l'étage "
                    + str(L+1)+"\n D: disponible, N : Non disponible"+
                    " ou H : Hors service")
                valid = Mat[L][C] in ["D", "N", "H"]
def nombre_chambre (Mat, N , M , etat ):
    nb=0
    for L in range( N ) :
        for C in range( M ) :
            if (Mat[L][C]==etat ) :
                nb=nb+1
    return nb
def affichage_Hors_services ( Mat , N , M ) :
    for L in range( N ) :
        print ("\nLes chambres hors services du "+ str(L+1)+ "étage ")
        for C in range( M ) :
            if (Mat[L][C]=="H" ) :
                print (str(C), end=" | ")
saisie ()
remplissage ( N , M )
print ("Le nombre de chambres disponibles = ", nombre_chambre (Mat, N , M , "D" ) )
print ("Le nombre de chambres non disponibles = ", nombre_chambre (Mat, N , M , "N" ) )
print ("Le nombre de chambres hors services = ", nombre_chambre (Mat, N , M , "H" ) )
print ("pourcentage d'occupation = %.3f"%(nombre_chambre (Mat, N , M , "N" ) / (N*M)))
affichage_Hors_services ( Mat , N , M )
```

Situation de problème 3 :

L'Association Ornithologique Tunisienne a décidé d'informatiser les N fiches d'adhésions de ces adhérents. Chaque adhérent possède les informations suivantes qui seront saisies au clavier :

- **Ident** : L'identifiant est une chaîne de 8 chiffres qui permet d'identifier d'une façon unique un adhérent.
- **Nom_Prénom** : une chaîne de caractères contenant le nom et le prénom,
- **Age** : représente l'âge de l'adhérent.
- **Civilité** : il contient "F" pour Femme ou "H" pour Homme.
- **Paiement** : il contient "O" si l'adhérent a payé sa cotisation ou "N" dans le cas contraire.

**Travail demandé :**

1. Peut-on utiliser une structure de données déjà vue pour stocker les informations d'un adhérent. Si oui, expliquer comment. Sinon proposer un autre type qui peut satisfaire ce besoin.
2. En utilisant la structure de données choisie, écrire l'algorithme qui permet de stocker les informations d'un adhérent (Identifiant, Nom_Prénom, Age, Civilité et paiement) puis de les afficher, puis de déduire le T.D.O. correspondant.
3. Implémenter l'algorithme précédant en langage Python.

B.2/ Les dictionnaires (les enregistrements) :**⊙ Définition**

Un dictionnaire est un objet conteneur qui stocke une collection de pair champ/Valeur(s). Il permet une récupération rapide, une suppression et une mise à jour de la valeur à l'aide du champ (clé) correspondant.

⊙ Création d'un dictionnaire avec une clé et plusieurs valeurs

Enregistrement = { champ1 : Valeur1 , champ2 : Valeur2 , champN : ValeurN }

Exemples :

- ❖ employe = { } c.a.d le dictionnaire est vide
- ❖ employe = { "Ident" : 123456 , "Nom_Prénom" : "Ali ben salem" , "Age" : 43 }

⊙ Accès aux éléments d'un dictionnaire

On peut accéder aux éléments d'un dictionnaire, il suffit d'écrire le nom du dictionnaire suivi de la clé entre deux crochets.

Exemple :

```
employe = { "Nom" : "Ali ben salem" , "Age" : 43 }
print ( employe [ "Nom" ] )
```

Solution :

```
def adhesion () :
    global adherent
    adherent={}
    valid=False
    while valid==False :
        adherent["Mat"] =input("Saisir la matricule chaine de 8 chiffres")
        valid = (len (adherent["Mat"])==8) & (adherent["Mat"].isnumeric()==True)
    adherent["np"]=input("saisir le nom et le prénom")
    adherent["age"]=int(input("saisir l'âge >=10"))
```

```

valid=False
while valid==False :
    adherent["genre"]=input("Saisir le genre")
    valid=adherent["genre"] in ["F", "H"]
valid= False
while valid==False :
    adherent["paiement"]=input("La cotisation est payée O/N ? ")
    valid=adherent["paiement"] in ["O", "N"]

def affichage (adherent ) :
    print("La matricule = ", adherent["Mat"])
    print("Le nom et le prénom = ", adherent["np"])
    print("L'age = ", adherent["age"])
    print("Le genre = ", adherent["genre"])
    print("paiement de la cotisation = ", adherent["paiement"])

adhesion ()
affichage (adherent )

```

⊙ Création d'un tableau de dictionnaire Vocabulaire et syntaxe :

Enregistrement = { champ1 : Valeur1 , champ2 : Valeur2 , champ3 : Valeur3 }

T [indice] = Enregistrement

Exemple :

employe = { "Nom" : "Ali ben salem" , "Age" : 43 }

T = [""] * 5

T[0] = employe

employe = { "Nom" : "Ali Slem" , "Age" : 42 }

T[4] = employe

print (T)

⊙ Manipulation d'un tableau d'enregistrements

T[indice] ["champ"]

Exemple :

- afficher le nom du premier employé : T[0]["nom"]
- afficher le nom du 5ème employé : T[4]["nom"]

Activité 4 :

1. Peut-on utiliser la structure de données tableau à une dimension pour stocker les informations des N adhérents avec ($5 \leq N \leq 1000$). Si oui, expliquer comment. Sinon proposer un autre type qui peut satisfaire ce besoin.
2. En utilisant la structure de données choisie, écrire l'algorithme qui permet de stocker les informations des adhérents puis de les afficher, puis de déduire le T.D.O. correspondant.
3. Implémenter l'algorithme précédant en langage Python.

Solution :

```

def saisie () :
    global N
    valid=False
    while valid==False :
        N = int(input("saisir N entre 5 et 1000"))
        valid= 3<=N<=1000
def adhesions ( N ) :

    global T
    T=[{}]*1000
    for i in range ( N ) :
        adherent={}
        valid=False
        while valid==False :
            adherent["Mat"] =input("Saisir la matricule chaine de 8 chiffres")
            valid = (len (adherent["Mat"])==8) & (adherent["Mat"].isnumeric()==True)
        adherent["np"]=input("saisir le nom et le prénom")
        adherent["age"]=int(input("saisir l'age >=10"))
        valid=False
        while valid==False :
            adherent["genre"]=input("Saisir le genre")
            valid=adherent["genre"] in ["F", "H"]
        valid= False
        while valid==False :
            adherent["paiement"]=input("a -t'il payer la contisation O/N ? ")
            valid=adherent["paiement"] in ["O", "N"]
        T[i]= adherent

def affichage (T , N ) :
    for i in range ( N ) :
        print("La matricule = ", T[i]["Mat"])
        print("Le nom et le prénom = ", T[i]["np"])
        print("L'age = ", T[i]["age"])
        print("Le genre = ", T[i]["genre"])
        print("paiement de la contisation = ", T[i]["paiement"])
        print("-----")

saisie ()
adhesions ( N )
affichage (T , N )

```