

Les Structures de Données Python

TP1 - Les structures de données séquentielles

Halim Djerroud

révision 0.1

Partie 1 : Algorithmique

Exercice 1

Soit le tableau ci-dessous

0	1	2	3	4	5	6	7	8	9	10	11	12	13
-12	11	13	12	23	10	21	15	32	?	?	?	?	?

1. Quelle est sa taille physique ?
2. Quelle est sa taille logique ?
3. Dessiner le tableau après chacune des opérations suivantes et mettre à jour sa taille logique. Pour chaque opération, considérer l'état du tableau après l'opération précédente.
 - (a) Ajout de la valeur 15 à la fin du tableau
 - (b) Ajout de la valeur 10 à la neuvième position.
 - (c) Suppression de la première valeur du tableau.
 - (d) Suppression de la première occurrence de la valeur 15.
 - (e) Suppression de toutes les occurrences de la valeur 10.
 - (f) Suppression de la valeur 36.

Exercice 2

1. Écrire un algorithme qui décale vers la gauche les valeurs contenues dans un tableau. Exemple :
 - avant : 10 12 20 30 35 21
 - après : 12 20 30 35 21 10
2. Écrire un algorithme qui cherche la plus grande valeur dans un tableau, On suppose que le tableau est déjà rempli.

Exercice 2

1. Écrire un algorithme qui cherche un nombre saisi au clavier dans un tableau d'entiers. L'algorithme affiche un message pour dire si la valeur existe ou non dans le tableau. Écrire deux versions de cet algorithme selon que le tableau soit ordonné dans l'ordre croissant ou non ordonné.
2. Écrire un algorithme qui affiche toutes les positions auxquelles une valeur saisi au clavier apparaît dans un tableau d'entiers.
3. Écrire un algorithme qui affiche le nombre d'occurrences d'une valeur dans un tableau d'entier.
4. Écrire un algorithme qui vérifie si deux tableaux contiennent les mêmes valeurs sans prendre en compte leurs positions mais en prenant en compte leurs nombres d'occurrences.

Exercice 3

1. Écrire un algorithme qui calcule, dans un tableau T3, l'union de deux tableaux T1 et T2. Nous supposons que chaque valeur de T1 et T2 apparaît une seule fois.
2. Écrire un algorithme qui calcule, dans un tableau T3, la fusion de deux tableaux ordonnés T1 et T2 telle que T3 soit également ordonné. Pensez à l'efficacité de votre algorithme.
3. Écrire un algorithme qui calcule, dans un tableau T3, l'intersection de deux tableaux T1 et T2. Nous supposons que chaque valeur de T1 et T2 apparaît une seule fois.

Partie 2 : Programmation - Les listes

Exercice 1

Remarque :

- Dans ce qui suit, et pour chaque exercice, vous devez d'abord écrire l'algorithme avant de le traduire en Python.
- Pour chaque fonction écrite, vous devez écrire un programme pour la tester.
- Pour chaque exercice écrit vous devez donner sa complexité (dans le meilleur des cas, le pire cas et le cas moyen).

AUCUNE FONCTION PREDEFINIE N'EST AUTORISÉE POUR CETTE PARTIE

1. Écrire un programme qui remplit une liste de valeurs entières données par l'utilisateur. Le programme affiche ensuite la liste.
2. Écrire une fonction qui calcule et affiche le minimum d'une liste.
3. Écrire une fonction qui calcule et affiche la moyenne d'une liste.
4. Écrire une fonction qui cherche une valeur dans une liste. Si la valeur existe, la position de sa première occurrence est retournée sinon retourner **None**.
5. Écrire une fonction qui retourne le nombre d'occurrences d'une valeur donnée dans une liste.
6. Écrire une fonction qui insère une valeur dans une liste ordonnée (en la maintenant ordonnée).
7. Écrire une fonction qui vérifie si une liste est ordonnée (dans l'ordre croissant).
8. Écrire une fonction qui décale vers la gauche les valeurs contenues dans une liste.
 - avant : 'D' 'E' 'C' 'A' 'L' 'A' 'G' 'E'
 - après : 'E' 'C' 'A' 'L' 'A' 'G' 'E' 'D'
9. Étant donnée une liste d'entiers L telle que $L[i] \in \{0, 1\}$, écrire une fonction qui retourne la position i telle que L[i] est le début de la plus longue série consécutive de zéros.
10. Écrire une fonction qui vérifie si deux listes contiennent les mêmes valeurs aux mêmes positions.
11. Écrire une fonction qui vérifie si deux listes contiennent les mêmes valeurs sans prendre en compte leurs positions, ni leurs nombres d'occurrences.
12. Écrire une fonction qui supprime la première occurrence d'une valeur, si elle existe, dans une liste.
13. Écrire une fonction qui supprime toutes les occurrences d'une valeur dans une liste.
14. Écrire une fonction qui calcule, dans une liste L3, la fusion de deux listes ordonnées L1 et L2 telle que L3 est également ordonnée. Pensez à l'efficacité de votre algorithme.
15. Écrire une fonction qui calcule, dans une liste L3, l'intersection de deux listes L1 et L2. Nous supposons que chaque valeur apparaît une seule fois.
16. Écrire une fonction qui trie une liste de valeurs entières dans l'ordre croissant.

Exercice 2

Traiter l'ensemble des questions de l'exercice 1 en utilisant les fonctions prédéfinies dans Python.

Partie 3 : Les tuples

Exercice 1

1. Écrire en langage Python une fonction qui prend en paramètre un tuple et qui le convertit en chaîne de caractères (sans oublier de tester la fonction dans un programme principal, cette remarque est valable pour l'ensemble des exercices).
2. Écrire en langage Python une fonction qui prend en paramètre un tuple et qui retourne son 4^{ème} élément, ainsi que son 4^{ème} élément en partant de la fin.
3. Écrire en langage Python une fonction qui reçoit une chaîne de caractères contenant des nombres séparés par des virgules en paramètre, et qui génère puis retourne une liste et un tuple contenant les nombres composants cette chaîne. Exemple :

```
Entrée :  
S = « 34,67,55,33,12,98 »  
Sortie :  
['34', '67', '55', '33', '12', '98']  
( '34', '67', '55', '33', '12', '98')
```

Exercice 2

Écrire puis tester une fonction qui prend en paramètres une liste de tuples (nom, âge, taille) et qui l'ordonne dans l'ordre croissant. La liste de tuples est donnée par l'utilisateur, et le critère de tri est le suivant :

1. d'abord le nom (ordre alphabétique) ;
2. puis l'âge ;
3. puis la taille.

```
Entrée :  
Tom,19,80  
John,20,90  
Jony,17,91  
Jony,17,93  
Json,21,85  
Sortie :  
[('John', '20', '90'), ('Jony', '17', '91'), ('Jony', '17', '93'),  
( 'Json', '21', '85'), ('Tom', '19', '80')]
```

Partie 4 : Les chaînes de caractères

Exercice 1

AUCUNE FONCTION PREDEFINIE N'EST AUTORISÉE POUR CETTE PARTIE

1. Écrire un programme qui demande à l'utilisateur de saisir une chaîne de caractères et compte le nombre de caractères qui composent cette chaîne.
2. Demande à l'utilisateur de saisir une chaîne de caractères et qui convertit les lettres minuscules dans cette chaîne de caractères en lettres majuscules.
3. Écrire un programme qui demande à l'utilisateur de saisir une chaîne de caractères et calcule le nombre de lettres alphabétiques, le nombre de chiffres et le nombre de caractères spéciaux qu'elle contient.
4. Écrire un programme qui demande à l'utilisateur de saisir deux chaînes de caractères et concaténer les deux chaînes dans une troisième variable
5. Écrire un programme qui demande à l'utilisateur de saisir deux chaînes de caractères et compare les deux chaînes de caractères selon l'ordre alphabétique.
6. Écrire un programme qui demande à l'utilisateur de saisir une chaîne de caractères et inverse la chaîne et l'affiche à l'écran.
7. Écrire un programme qui demande à l'utilisateur de saisir une chaîne de caractères et vérifie si la chaîne est un palindrome.

8. Écrire un programme qui demande à l'utilisateur de saisir une chaîne de caractères et trouve le caractère le plus fréquent dans cette chaîne.
9. Écrire un programme qui demande à l'utilisateur de saisir une chaîne de caractères. Demande à l'utilisateur de saisir un caractère. Supprime la première occurrence de ce caractère dans la chaîne.
10. Écrire un programme qui demande à l'utilisateur de saisir une chaîne de caractères. Demande à l'utilisateur de saisir un caractère. Supprime la dernière occurrence de ce caractère dans la chaîne.

Exercice 2

Traiter l'ensemble des questions de l'**exercice 1** en utilisant les fonctions prédéfinies dans Python .