

Structure de données : C4 les listes chaînées

TP

1 Exercice 1 : comprendre la structure de la liste chaînée 🏆

1. Créer une liste chaînée `l_c` avec 3 valeurs différentes à l'aide de la classe `Cellule` du cours dans Thonny.
2. Que renvoie : `print(l_c)` ? `print(l_c.valeur)` ? `print(l_c.suivante)` ?
3. Faire un schéma de votre liste chaînée avec les noms des emplacements mémoire de chacun des éléments, et des flèches reliant les différentes cellules.
4. Comment atteindre votre dernière valeur en utilisant une syntaxe du type de la question 2?

2 Exercice 2 : Version itérative nième-élément 🏆

Réécrire la fonction `nieme_element` du cours en utilisant une boucle `while`.

3 Exercice 3 : modification de listes et concaténation 🏆

1. Créer une liste chaînée `l_c1` contenant les valeurs 9,8,7 avec la classe `Cellule` du cours.
2. Créer une liste chaînée `l_c2` contenant les valeurs 6,5,4,3.
3. Modifier `lc_1` en modifiant le **dernier attribut suivante** de `l_c1` afin d'obtenir une liste ayant des valeurs de 9 à 3. Afficher le 4ième puis le 7ième élément de la liste `l_c1`.
4. Créer une liste `l_c3` contenant les valeurs 2,1.
5. Modifier `lc_2` en modifiant le **dernier attribut suivante** de `l_c2` afin d'obtenir une liste ayant des valeurs de 6 à 1.
6. Afficher le 7ième élément de la liste `l_c1`. Que remarquez-vous? Pourquoi cela peut-il être gênant de procéder ainsi pour concaténer des listes?

4 Exercice 4 : Affichage 🏆

Ecrire une fonction `affiche_liste(l_c)` qui affiche, en utilisant la fonction `print`, tous les éléments de la liste `l_c` séparés par des espaces, suivis d'un retour chariot. L'écrire comme une fonction récursive puis avec une boucle `while`.

5 Exercice 5 : Liste d'entiers 🏆 🏆

Ecrire une fonction `listeN(n)` qui reçoit en argument un entier `n`, supposé positif ou nul, et renvoie la liste chaînée des entiers 1,2, ..., `n` dans l'ordre. Si `n = 0`, la liste renvoyée est vide.

6 Exercice 6 : parcours de listes 🏆 🏆

Soit la liste suivante `liste_c = Cellule('l', Cellule('a', Cellule(' ', Cellule('n', Cellule('s', Cellule('i', Cellule(' ', Cellule('c', Cellule('“““, Cellule('e', Cellule('s', Cellule('t', Cellule(' ', Cellule('l', Cellule('a', Cellule(' ', Cellule('v', Cellule('i', Cellule('e', None))))))))))))))))))`

1. Ecrire une fonction `occurences(x, liste_c)` qui renvoie le nombre d'occurences de la valeur `x` dans la liste `liste_c`. L'écrire de façon récursive puis avec une boucle `while`.
2. Ecrire une fonction `trouve(x, liste_c)` qui renvoie le rang de la première occurrence de `x` dans `liste_c`, et `None` sinon. L'écrire de façon récursive puis avec une boucle `while`.

7 Exercice 7 : Tri par insertion 🏆 🏆 🏆

1. Ecrire une fonction `insérer(x, lst)` qui prend en arguments un entier `x` et une liste d'entiers `lst`, supposée triée par ordre croissant, et qui renvoie une nouvelle liste dans laquelle `x` a été inséré à sa place. L'écrire de façon récursive.
2. Ecrire une fonction `tri_par_insertion(lst)` qui prend en argument une liste d'entiers `lst` et renvoie une nouvelle liste, contenant les mêmes éléments et triée par ordre croissant. L'écrire de façon récursive.

Sources :

- * Cours NSI S Ramstein Lycée Quenot à Villeneuve d'Ascq
- * Cours NSI A Wilm Lycée Beaupré à Haubourdin
- * NSI Terminale Prépabac Hatier
- * NSI Terminale Ellipses T. Balabonski, S. Conchon, JC. Filliatre, K Nguyen
- * NSI Terminale Ellipses S Bays
- * NSI Terminale Ellipses JC Bonnefoy B Petit