

Lab Structures de données n°1

« Collecter des données n'est pas savoir. La véritable connaissance est un processus de réflexion, un savoir faire, et non une accumulation de données, à rebours de l'idée reçue que la connaissance se trouverait toute faite dans quelque chose : un livre, un ordinateur, une encyclopédie en ligne. »

Dimitri Casali

I. Liste Chainée

Durant ce lab, nous nous proposons de construire une liste chaînée et de construire les fonctions qui nous permettront d'y naviguer ou de la modifier.

a. Définition de la cellule :

On se propose d'avoir la structure suivante comme cellule :

Structure Cellule

valeur : entier, // un entier à stocker

suivant : cellule* // un pointeur sur la cellule suivante

Fin Structure

Ecrivez la fonction « generer_cellule() » qui retourne un dictionnaire contenant les deux champs de la structure cellule spécifié ci dessus.

La valeur de suivant sera initialisée à None.

Dans toute la suite du problème, toute création d'une nouvelle cellule sera faite en utilisant cette fonction.

b. Première liste chaînée :

Construisez manuellement une liste chaînée qui contient les valeurs 4, 8, 16 et 32.

c. Accès à une cellule

Ecrivez en python la fonction qui a la signature suivante :

```
def element(Liste_chainée,position) : return cellule
```

Cette fonction prend donc en paramètre une liste chaînée, et retourne la cellule à l'index spécifié par le paramètre position.

Pensez à gérer les cas d'erreurs !

d. Ajout d'une cellule

Ecrivez en python la fonction qui a la signature suivante :

```
def ajout(Liste_chainée,cellule) : return None
```

Cette fonction prend en paramètre une cellule et l'ajoute comme premier élément de la liste chaînée.

e. Insertion d'une cellule

Ecrivez en python la fonction qui a la signature suivante :

```
def insertion(Liste_chainée,cellule,position) : return None
```

Cette fonction a pour but d'ajouter une nouvelle cellule à la chaîne à l'index spécifié par position.

f. Suppression d'une cellule

Ecrivez en python la fonction qui a la signature suivante :

```
def suppression(Liste_chainée ,position) : return cellule
```

Cette fonction a pour but de retirer la cellule appartenant à la liste chaînée prise en paramètre à l'index spécifié par position et retourne la cellule retirée.

II. Applications

a. Recherche d'un élément

A partir de la définition d'une liste chaînée donnée dans la partie précédente, écrivez une fonction qui prend en paramètre une liste chaînée et un entier et qui retourne la position de la première occurrence de cet entier dans la liste, None s'il n'y est pas

b. Mini tri

Ecrivez une fonction qui cherche la valeur maximale dans une chaîne puis place la cellule contenant cette valeur à la tête de la liste.

c. Tri

A partir de la fonction définie précédemment, construisez une fonction qui trie toute une liste chaînée.

Bon travail