

LISTES CHAINEES

EXERCICE1 : Concaténation de 2 listes chaînées (6points)

Pour réaliser cette opération vous devez suivre les étapes ci-dessous :

1) CONVERSION : Ecrire un algorithme qui permet de saisir 2 nombres décimaux et les convertir en binaire. Les résultats de ces conversions seront stockés dans 2 tableaux **V1 et V2**

2) CONSTRUCTION D'UNE LISTE CHAINEE : Ecrire un algorithme de construction des listes chaînées **L1 et L2** à partir des Tableaux binaires **V1 et V2**.

3) CONCATENATION : Ecrire un algorithme de Concaténation la liste **L1 et L2**.

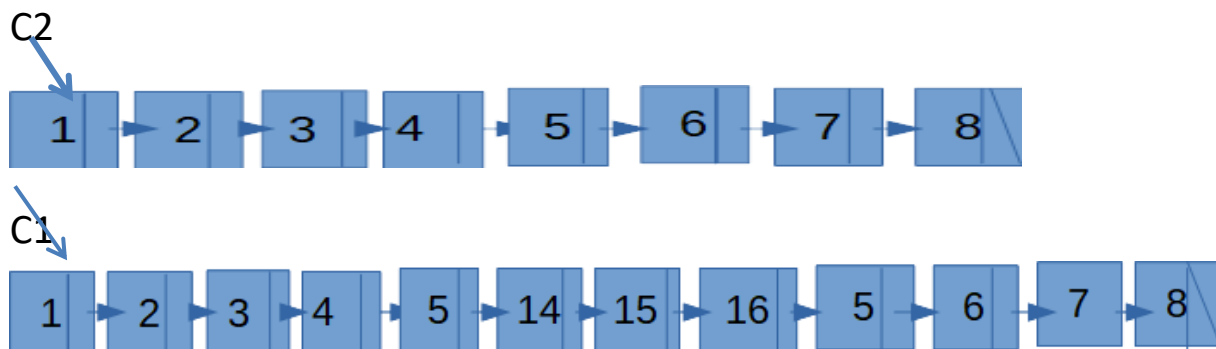
4) AFFICHAGE : Ecrire un algorithme d'affichage du résultat de la **liste binaire finale**.

EXERCICE2 : (4points)

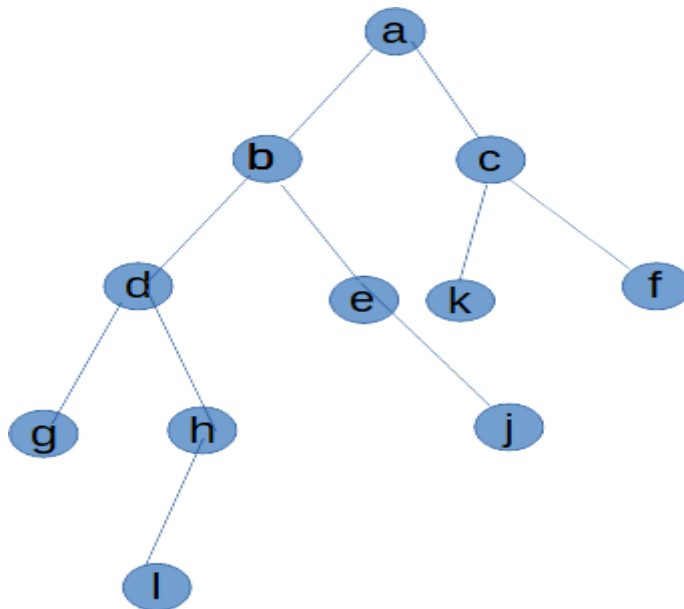
Chaine C2 est dite Extraite d'une chaîne C1, si les 2 premières valeurs de C1 et de C2 sont égales, les 2 dernières valeurs de C1 et de C2 sont égales, et les autres sont égales à 2.

- 1) Ecrire un algorithme qui détermine si la chaîne C2 est Extraite de la chaîne principale C1.
- 2) Estimer la complexité

Exemple :



EXERCICE4 : ARBRE(6points)



- a) Qu'elle est la hauteur de cet arbre ?
- b) Qu'elle est la taille de cet arbre ?
- c) Est-ce que cet arbre est un arbre parfait, donnez la raison ? un arbre dégénéré ? Donner l'ordre de cet arbre.
- d) Ecrire la structure de cet arbre en vue d'une recherche (ou d'une confirmation) qu'un nœud est ancêtre commun de 2 nœuds quelconques.
- e) Affichez les parcours en largeur et en profondeur de cet arbre.
- f) Représenter uniquement l'exécution du Sous Arbre-Droit en affichant les appels récurifs et les traitements des nœuds.
- g) Représenter cet arbre par un tableau en vue de réaliser des traitements, donner des indications de positions pour chercher le PERE et les FILS à partir d'une position quelconque.
- h) Donnez la représentation arborescente de cette Expression arithmétique :
Exp=a*b/c+(d+e)

EXERCICE3 : TRI4points)

Soit **T** un tableau de n boules, dont certaines sont Vertes, d'autres Jaunes et d'autres Violettes (le nombre de boules de chaque couleur n'est pas connu et n'est pas déterminant pour la suite du problème).

1) Ecrire un algorithme itératif qui répartit le tableau de manière à ce que toutes les boules jaunes soient placées devant, suivies des boules violettes, et enfin les boules vertes, pour former le tableau de couleurs.

Paramétrer le tableau initial des boules, le tableau intermédiaire et le tableau final.

2) Afficher l'exécution de l'algorithme pour un tableau de 8 boules, en désignant les indices associés aux boules jaunes classées, aux boules violettes classées, aux boules non classées et aux boules vertes classées.

Cet algorithme doit s'exécuter sur le même tableau, et doit donc effectuer uniquement des déplacements de boules, et doit avoir une complexité en temps en $O(n)$. Il n'est pas autorisé d'effectuer un comptage des boules de chaque couleur, ni de changer la couleur d'une boule.

Suggestion : Utiliser trois indices. S'assurer que votre algorithme préserve l'invariant suivant :

Ceci est illustré par la figure suivante à un instant donné avant la fin de l'exécution de l'algorithme.

Tableau T



Situation Initiale

i	j		k
Jaunes classées	Violettes classées	Boules non classées	Vertes classées

Situation Intermédiaire



Situation Finale