

### Série d'exercices n°3

**Exercice 01 :**

Enrichir le TAD arbre binaire avec les opérations suivantes:

- 1- Min\_valeur: recherche la plus petite valeur dans un arbre binaire
- 2- Max\_valeur: recherche la plus grande valeur dans un arbre binaire
- 3- Arbre\_recherche: vérifie si un arbre donné est un arbre de recherche
- 4- Trouver\_feuille\_gauche : recherche la feuille qui se trouve à l'extrême gauche
- 5- Trouver\_feuille\_droite: recherche la feuille qui se trouve à l'extrême droite
- 6- Insérer\_gauche : Insérer un noeud à l'extrême gauche
- 7- Insérer\_droite: Insérer un noeud à l'extrême droite
- 8- Permuter\_droite\_racine: permuter la feuille qui se trouve à l'extrême droite et la racine

**Exercice 02 :**

- 9- Enrichir le TAD arbre binaire par l'opération Nb\_feuilles qui calcule le nombre de feuilles d'un arbre binaire.
- 1- Donner la fonction C correspondante en prenant la représentation chaînée des arbres.

**Exercice 03:**

Ecrire une fonction `int nb_noeuds_ayant_2_fils(A :Arbre)` qui retourne le nombre de noeuds de l'arbre binaire A ayant deux fils.

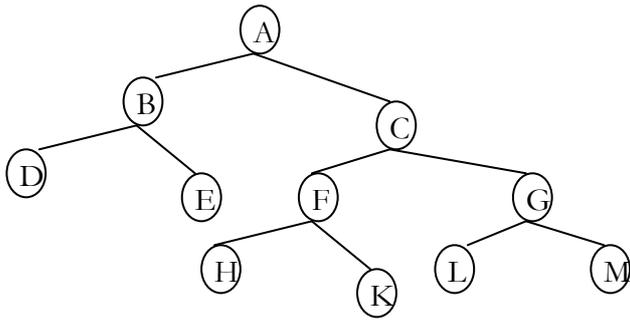
**Exercice 04:**

- 1- Utiliser la fonction `Arbre construire(n :entier, g,d: Arbre)` pour écrire la fonction `ajouter(B: Arbre , m :entier)` qui ajoute l'entier m à l'arbre binaire de recherche B.
- 2- Ecrire une fonction `recherche(A :Arbre , n :entier)` qui détermine si la valeur n se trouve ou non dans l'arbre binaire de recherche A.

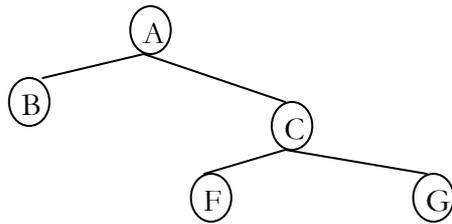
**Exercice 05:**

Il y a une queue de cerise dans un arbre quand il a un noeud interne dont les deux fils sont des feuilles.

- 1- Enrichir le TAD arbre binaire par l'opération `nbre_cerise` qui donne le nombre de queue de cerise dans un arbre binaire.
- 2- Ecrire une fonction qui calcule le nombre de queue de cerise dans un arbre binaire.
- 3- En utilisant la représentation chaînée, écrire un programme JAVA qui calcule le nombre de queue de cerise dans un arbre binaire.
- 4- Ecrire une fonction C un programme JAVA qui supprime les feuilles de toutes les queues de cerise d'un arbre. Par exemple, si l'arbre est



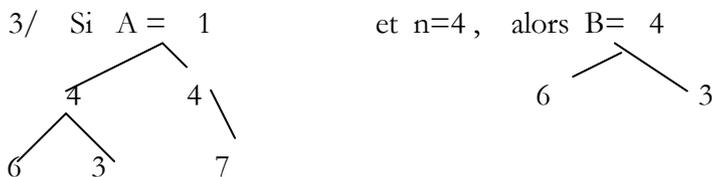
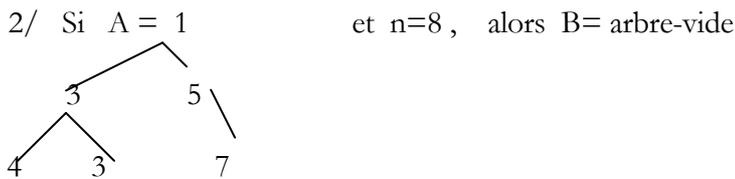
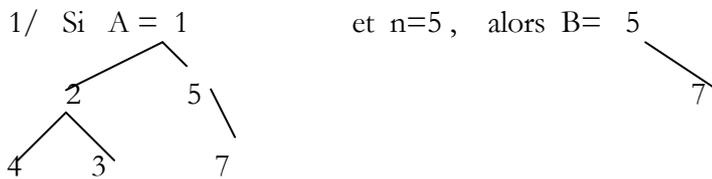
il sera transformé en :



**Exercice 06:**

1. Enrichir le TAD arbre binaire par l'opération **sous-arbre** qui étant donné un nœud **n**, elle retourne un sous arbre **B** d'un arbre donné **A** ayant comme racine n.

Exemples :

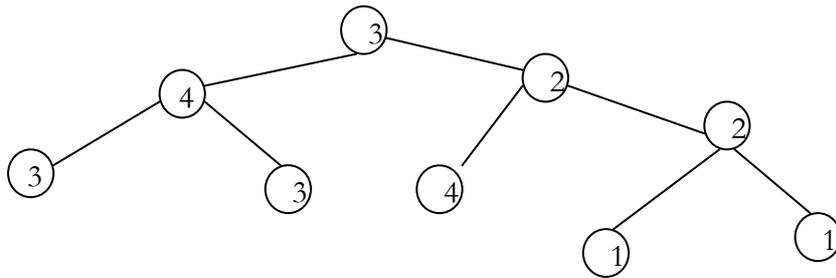


- 2- En utilisant l'opération précédente, enrichir le TAD arbre binaire par l'opération **nbre-descendants** qui retourne le nombre de descendants d'un nœud dans un arbre binaire.
- 3- En utilisant la représentation chaînée des arbres écrire un programme JAVA qui retourne le nombre de descendants d'un nœud donné

**Exercice 7:**

Soit  $(p, mg, md)$  un mobile formé d'un poids  $p$  central et de deux « sous-mobiles »  $mg$  à gauche et  $md$  à droite. On dira que  $p$  est équilibré si le poids de  $mg$  est égal au poids de  $md$  et si  $md$  et  $mg$  sont équilibrés.

Exemple de mobile équilibré  
de poids 23 :



- 1- Proposez une représentation informatique de ce mobile.
- 2- Spécifiez l'opération `mobile_équilibré` qui teste si un mobile est équilibré ou non (en utilisant l'opération prédéfinie `poids`).
- 3- Ecrire un programme JAVA qui teste si un mobile est équilibré ou non.