

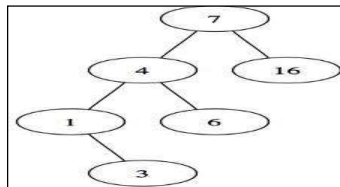


SÉRIE N° 8 : ARBRES BINAIRES (SUITE)

Exercice 1 :

1. Ecrire la fonction **NbrFeuilles(A)** qui prend en paramètre un arbre A et qui retourne le nombre de feuilles de l'arbre A (ou nœuds externes).
2. Ecrire la fonction **NbrNoeudInternes (A)** qui prend en paramètre un arbre A et qui retourne le nombre de nœuds internes de l'arbre A.
3. Ecrire la fonction **AffichageLargeur(A)** qui prend en paramètre un arbre A et qui affiche les éléments de l'arbre en largeur

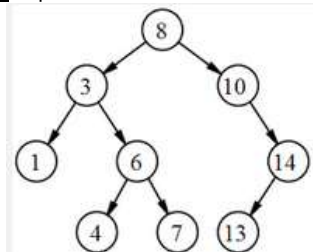
Exemple :



Pour l'arbre ci-dessus, la fonction affiche : 7 4 16 1 6 3

Exercice 2 : Arbre binaire de recherche

Un arbre binaire A est un arbre binaire de recherche si, pour tout nœud s de A, les contenus des nœuds du sous-arbre gauche de s sont strictement inférieurs au contenu de s, et que les contenus des nœuds du sous-arbre droit de s sont strictement supérieurs au contenu de s.



1. Écrire une fonction **ABR(A)** qui vérifie si un arbre donné est bien un arbre binaire de recherche.
2. Écrire la fonction **mem(A,x)** qui prend en paramètre un ABR et un nombre x et qui recherche si x appartient à A.
3. Écrire une fonction qui prend un arbre binaire de recherche et renvoie son minimum.
4. Écrire une fonction qui prend un arbre binaire de recherche et renvoie son maximum.
5. Écrire la fonction **add(A,x)** qui ajoute un élément donné x à un arbre binaire de recherche donné A tout en garantissant que l'arbre reste un arbre binaire de recherche.