

## ÉNONCÉ – FEUILLE DE RÉPONSE

**NOM** ..... **Prénom** .....

Les types de données considérés sont ainsi définis :

type structure nœud

info : entier

sag : adresse nœud // sous-arbre gauche

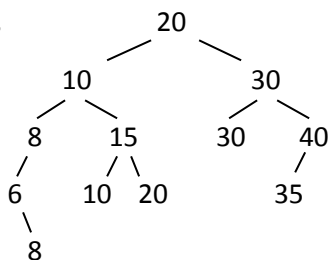
sad : adresse nœud // sous-arbre droit

fintype

type adresse nœud : arbre

Les algorithmes demandés sont typiquement attendus en langage algorithmique et devront être autonomes.

**Exercice 1.**



Cet arbre est-il un <i>arbre binaire de recherche</i> (au sens de la définition donnée en TD) ?	OUI - <b>NON</b> (entourer la bonne réponse)
Si non, remplacer dans l'arbre chaque valeur qui pose problème par la plus petite valeur qui peut convenir, ou indiquer ci-contre toute autre raison du problème.	Fils gauche de 15 de valeur 10 : remplacer par 11
On souhaite afficher les valeurs de tous les nœuds de l'arbre. Qu'afficherait un <i>parcours en profondeur infixe</i> (ou <i>in-ordre</i> ) ?	6 8 8 10 10 15 20 20 30 30 35 40

## Exercice 2.

Fonction appartient( a : arbre, x : entier) : adresse nœud

// Donnée : un arbre quelconque a

// Donnée : une valeur x à rechercher

// Résultat : adresse d'un nœud de a contenant x, ou NULL si aucun.

// Variables locales : .... (si besoin)

Début

Si a = NULL alors retourner NULL

Fin.

// Sinon

Si a → info = x alors retourner a

r ← appartient(a → sag, x)

si r ≠ NULL alors retourner r

retourner appartient(a → sad, x)

## Exercice 3.

Procédure supprimerFeuilles( a : arbre, p : EntierNat)

// Donnée modifiée : un arbre quelconque a ; l'arbre modifié final est l'arbre initial dont on a supprimé toutes les  
// feuilles de profondeur p (s'il en existe).

// Exemple. Si on prend l'arbre de l'exercice 1 et une profondeur p=3, il s'agit de supprimer les

// feuilles de valeur 10, 20 et 35.

// Donnée : une profondeur p de type entier naturel

// Variables locales : .... (si besoin)

Début

Si a = NULL alors retourner ;

Fin.

// Sinon

Si p = 0 alors

Si a → sag=NULL et a → sad=NULL alors

libérer a ;

a ← NULL ;

finsi

retourner ;

finsi

supprimerFeuilles(a → sag, p-1) ;

supprimerFeuilles(a → sdg, p-1) ;