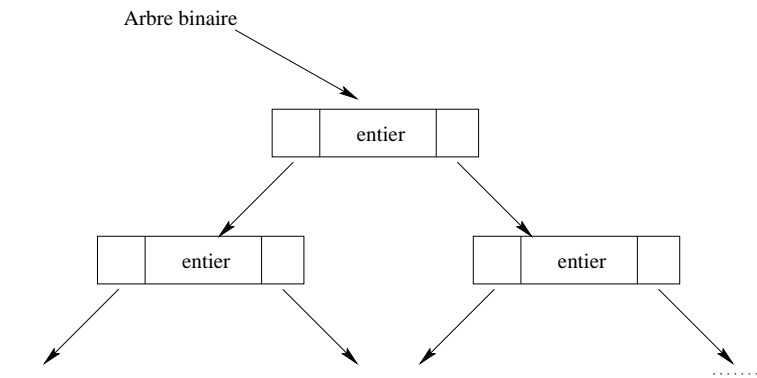


TP7

Unité Arbre binaire d'entiers

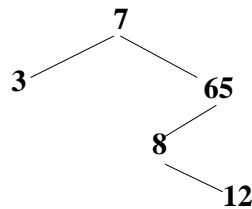
Un arbre binaire d'entiers est représenté par un pointeur vers une cellule qui contient d'une part un entier et d'autre part deux pointeurs vers d'autres cellules.



Créer l'unité Arbre binaire et le fichier de test associé qui permettent de :

1. Définir le type ARBRE d'entier(dynamique).
2. Définir le sous-programme INITIALISE l'arbre binaire (donné en paramètre) .
3. Définir un sous-programme qui permet de tester si l'arbre (donné en paramètre) est VIDE.
4. Définir un sous-programme qui permet d'ajouter un entier (donné en paramètre) à un arbre (donnée en paramètre). Pour ajouter un entier, on doit parcourir l'arbre de façon à positionner l'entier à sa place : l'arbre est trié en GRD (gauche racine droite : le plus petit entier est le plus à gauche possible de l'arbre).

Exemple : insérer successivement 7, 65, 8, 12 et 3 donne l'arbre suivant :



5. Définir un sous-programme qui permet d'enlever le plus petit élément de l'arbre (donné en paramètre) ce sous-programme doit renvoyer l'élément en question.
6. Définir un sous-programme récursif AFFICHE qui permet d'afficher tous les éléments de l'arbre dans l'ordre croissant.
7. ***Si vous avez du temps, implémenter une des méthodes de parcours itératives vues en CTD et essayez d'évaluer les différences au niveau performance***
 - (a) Peut-on transformer le sous-programme récursif AFFICHE en un sous-programme itératif, qui fait la même chose, sans utiliser de structure de données auxiliaire et sans modifier le type ARBRE ?
 - (b) Si non, quelle structure de données auxiliaire, autre qu'un arbre proposez-vous ? Donnez alors le sous-programme itératif correspondant.

- (c) Peut-on éviter la structure de données auxiliaire en modifiant le type ARBRE ? Donnez alors le nouveau type ARBRE et le sous-programme itératif correspondant.

Vous devez tester TOUS vos sous-programmes en envisageant TOUS LES CAS possibles.