

## TP 9 : Les arbres binaires de recherche

### Exercice :

1- Ecrire la **structure Noeud** qui est la structure d'un noeud d'un arbre binaire de recherche. La structure *Noeud* comprend une **valeur** (int) et **deux fils gauche et droit**, ils sont de type pointeurs sur la structure Noeud.

Ecrire des fonctions:

2- Pour initialiser l'arbre à NULL. Utiliser le prototype suivant:

*Noeud\* Initialiser(Noeud\* racine)*

3- Pour créer un noeud de l'arbre. Cette fonction initialise la **valeur** du noeud racine à **n** et les **fils gauche et droit** à **NULL**. Utiliser le prototype suivant: *Noeud\* creer\_noeud ( int n )*

4- Pour insérer un nouveau noeud dans l'arbre de manière récursive. Utiliser le prototype suivant: *Noeud\* ajouter\_noeud ( int n , Noeud\* racine )*

5- Pour afficher les valeurs des noeuds d'un arbre en utilisant les parcours préfixe, infixe et postfixe. Les prototypes des procédures sont les suivants:

*void affiche\_prefixe (Noeud\* racine )*

*void affiche\_infixe (Noeud\* racine )*

*void affiche\_postfixe (Noeud\* racine )*

6- Pour rechercher itérativement et récursivement une valeur **n** dans l'arbre. Les fonctions retournent l'adresse mémoire du noeud dont la valeur est égale à **n** et NULL sinon.

Les prototypes des fonctions sont:

*Noeud \* rechercher\_valeur\_iter ( int n , Noeud\* racine )*

*Noeud \* rechercher\_valeur\_recu ( int n , Noeud \* racine )*

7- Ecrire une procédure pour **enregistrer les noeuds** de l'arbre dans un **fichier**. Utiliser un type de parcours de votre choix (**préfixe, infixe** ou **postfixe**).

8- Ecrire le programme principal qui fait appel aux fonctions précédentes via un **menu**.

### Exemple d'application:

- Dans cet ordre, insérer les noeuds suivants dans l'arbre: 12, 5, 3, 7, 11, 8, 10, 15, 14 et 23.

- Rechercher les noeuds 8 et 20 dans l'arbre.

- Afficher les noeuds de l'arbres de 3 manières différentes.