TP : Structures de données

On souhaite manipuler un **arbre binaire de recherche d'entiers** à l'aide du langage C.

Chaque nœud contient :

- une valeur entière
- un pointeur vers le fils gauche
- un pointeur vers le fils droit

1. Définir la structure Node représentant un nœud d'un arbre binaire.
2. Écrire une fonction « **Node* creerNoeud(int val)** » qui crée un nouveau nœud.
3. Écrire une fonction « **Node* insererABR(Node *a, int val)** » qui insère une valeur dans un arbre binaire de recherche.
4. Écrire une fonction « **int compterNoeuds(Node *a)** » qui retourne le nombre total de nœuds.
5. Écrire une fonction récursive « **void parcoursInfixe(Node *a)** » qui parcours et affiche les éléments de l'arbre avec un parcours infixe.
6. Écrire une fonction récursive « **int sommeNoeuds(Node *a)** » qui retourne la somme des valeurs des nœuds d'un arbre.
7. Écrire une fonction récursive : « **int estDegeneré(Node *a)** » qui retourne 1 si l'arbre est dégénéré 0 sinon.
   Un arbre dégénéré ne contient **aucun nœud ayant deux fils non nuls**.
8. Ecrire une fonction « **int arbresIdentiques(Node *a1, Node *a2);** » qui vérifie si deux arbre sont identiques.

Vous pouvez copier cette fonction main pour tester :

```c
int main() {

    Node *arbre = NULL;

    /* Insertion des valeurs */
    int valeurs[] = {10, 5, 15, 3, 7, 12, 20};

    for (int i = 0; i < 7; i++) {
        arbre = insererABR(arbre, valeurs[i]);
    }

    /* Affichage infixe */
    printf("Parcours infixe (ordre trié) : ");
    parcoursInfixe(arbre);
    printf("\n");

    /* Nombre de nœuds */
    printf("Nombre total de noeuds : %d\n", compterNoeuds(arbre));

    /* Somme des valeurs */
    printf("Somme des valeurs : %d\n", sommeNoeuds(arbre));

    /* Test arbre dégénéré */
    if (estDegeneré(arbre)) {
        printf("L'arbre est dégénéré.\n");
    } else {
        printf("L'arbre n'est pas dégénéré.\n");
    }

    return 0;
}
```