

TP 7 : Arbre binaire de recherche

On définit un arbre binaire de recherche par la structure :

```
typedef struct noeud_S *ABR;  
  
struct noeud_S {  
    int info;  
    ABR filsGauche;  
    ABR filsDroit;  
} noeud;
```

1. Ecrire une fonction qui permet de créer un ABR.

Son prototype est : `ABR creerNoeud(int i, ABR g, ABR d)`

2. Ecrire une fonction qui permet d'insérer un élément dans un ABR.

Son prototype est : `ABR insererABR(ABR a, int i)`

3. Ecrire une fonction qui permet d'afficher l'ABR en ordre croissant.

Son prototype est : `void parcours(ABR a)`

4. Ecrire une fonction qui permet de rechercher un élément dans un ABR.

Son prototype est : `int rechercherABR(ABR a, int n)`

5. Ecrire une fonction qui permet de rechercher un élément dans un arbre binaire.

Son prototype est : `int rechercherAB(ABR a, int n)`

6. Ecrire deux fonctions qui permettent de rechercher la plus petite et la plus grande valeur dans un ABR.

Leurs prototypes sont : `ABR chercherMin(ABR a)` et `ABR chercherMax(ABR a)`

7. Ecrire une fonction qui permet de supprimer un élément d'un ABR, vous pouvez utiliser l'une des fonctions définies dans la question 5 pour la suppression d'un noeud qui contient deux fils.

Son prototype est : `ABR supprimerABR(ABR a, int valeur)`