

TP5 : Structures de données en langage C

Parcours MIPC – 1 séance

Soit la structure de donnée suivante qui permet de représenter un nœud d'un arbre binaire d'entiers :

```
typedef struct elt_arbre {  
    int info ;  
    struct elt_arbre *sag ;  
    struct elt_arbre *sad ;  
} Noeud ;
```

et soit une **file dynamique** dont les données sont des pointeurs de type **Noeud**. En se basant sur les structures de données suivantes :

<pre>typedef struct file { Noeud *donnee ; struct file *suivant ; } File;</pre>	<pre>typedef struct { File *debut; File *fin; int taille; } Var_File;</pre>
---	---

1- Donner les fonctions **initialiser**, **ajouter** et **retirer** pour bien manipuler la file dynamique. La fonction **retirer** doit rendre l'adresse du nœud stockée dans l'élément à extraire. Les trois fonctions doivent être paramétrées par une variable de type ***Var_File**.

Notre objectif est de rendre un arbre binaire en un arbre binaire ordonné. Pour réaliser cet objectif, nous avons besoin de parcourir notre arbre en largeur.

2- Ecrire la fonction **parcourir_largeur** et paramétrer cette fonction par un tableau pour stocker les adresses des nœuds visités de l'arbre et par une variable entière pour sauvegarder le nombre de nœuds.

3- Ecrire la fonction **preparer_noeud** qui retourne l'adresse de l'espace de mémoire d'allouer pour un nouveau nœud d'un arbre et place une valeur entière dans ce nœud passée comme un paramètre de la fonction.

4- Donner la fonction **ajouter_noeud** qui ajoute un nœud dans un arbre binaire ordonné.

5- Ecrire maintenant la fonction **rendre_ab_abo** qui permet de rendre un arbre binaire en un arbre binaire ordonné.

6- Ecrire la fonction qui permet d'en déduire le minimum et le maximum de l'arbre binaire ordonné.