

## Contrôle en Algorithme et Programmation

### MIPC (Section B)

Durée 2h 30 mn

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**. Donner les structures de données à utiliser pour représenter cette file et les variables pour avoir le bon contrôle sous la forme d'un enregistrement appelé **Var\_File**.

2- Donner les fonctions **initialiser**, **ajouter** et **retirer** pour manipuler la file dynamique. Les trois fonctions doivent être paramétrées par une variable de type **struct Var\_File**.

3- Notre objectif est de rendre un arbre binaire en un arbre binaire ordonné. Pour réaliser cet objectif, on a besoin de parcourir notre arbre en largeur. Ecrire cette fonction **parcourir\_largeur** et essayer qu'elle stocke les adresses des nœuds visités dans un tableau.

4- Ecrire la fonction **preparer\_noeud** qui retourne l'adresse de l'espace de mémoire d'allouer pour un nouveau nœud d'un arbre et qui place une valeur entière dans ce nœud.

5- Donner la fonction **ajouter\_noeud** qui ajoute un nœud dans un arbre binaire ordonné.

6- Ecrire la fonction **Nbre\_noeuds** qui permet de compter le nombre de nœuds de votre nouvel arbre.

7- Ecrire maintenant la fonction **rendre\_ab\_abo** qui permet de rendre un arbre binaire en un arbre binaire ordonné.

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