

Algorithmique et programmation procédurale

TP6 – ABR et parcours

Définir en C une bibliothèque d'ABR (ABR.c et ABR.h) basée sur le type de donnée

```
typedef struct ABR {
    int racine;
    struct ABR* filsG;
    struct ABR* filsD;
} ABR;
```

Implémenter les opérations suivantes :

- `ABR* inserer(ABR* a, int x)` : ajoute en feuille un sommet contenant l'entier x puis renvoie le nouvel ABR ;
- `ABR* supprimer(ABR* a, int x)` : supprime le sommet qui contient l'entier x ;
- `ABR* rechercher(ABR* a, int x)` : recherche le sommet qui contient l'entier x. Renvoie NULL si x n'existe pas dans l'ABR ;
- `int estABR(ABR* a)` : teste si l'arbre passé en paramètre est ABR ou pas ;
- `void parcoursProfRec(ABR* a)` : affiche le parcours en profondeur selon un algorithme récursif.

b) Ecrire un programme principal qui inclut votre bibliothèque et permet de tester ses fonctions. A partir d'un ABR vide au démarrage, l'utilisateur doit être capable d'effectuer via un menu textuel les opérations de l'exercice précédent. Après chaque action, votre programme affiche l'état actuel de l'arbre.

c) Ajouter à votre bibliothèque d'ABR la méthode

- `void parcoursLargeur(ABR* a)` : affiche le parcours en largeur

Pour cela, définir une bibliothèque de File (file.c et file.h) spécifiant la structure

```
typedef struct File {
    void* valeur;
    struct File* suivant;
} File;
```

ainsi que ses opérations

- `File* enfiler(File* f, void* x)` : enfile l'élément x et renvoie la file
- `File* défiler(File* f)` : défile le premier élément et renvoie la file
- `void* teteFile(File* f)` : renseigne sur le premier élément de la file (mais ne le défile pas !). Renvoie NULL si la file est vide.