

```

/*****
*** Gestion d'arbres binaires dynamiques ***
*****/
#ifndef ARBREBINAIRE_H
#define ARBREBINAIRE_H

#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <stdbool.h>

#define MALLOC(x) ((x * ) malloc(sizeof(x)))

#define ELEMENTNULL 0
#define ARBRENULL NULL

/*
 * Definition du type d'Element
 */
typedef int Element;

/*
 * Structure Arbre et PArbre (pointeur sur Arbre)
 */
typedef struct arb {
    Element elmt;
    struct arb* fg;
    struct arb* fd;
    int equilibre;
    int hauteur;
} Arbre;

typedef Arbre* PArbre;

#include "file.h"

/* Structure d'affichage graphique
 */
typedef struct {
    int elmt;
    int info;
} TArbBin;

/*****
 * Prototypes Arbre
 */
int max(int , int );
int min(int , int );
bool estVide(PArbre );
bool estFeuille(PArbre );
Element racine(PArbre );
PArbre modifierRacine(PArbre , Element );
bool existeFilsGauche(PArbre );
bool existeFilsDroit(PArbre );
PArbre filsGauche(PArbre );
PArbre filsDroit(PArbre );
int hauteur(PArbre );
int taille(PArbre );
int nbFeuilles(PArbre );
int degreNoeud(PArbre );
int degreArbre(PArbre );
bool degenere(PArbre );

```

```
bool fourchu(PArbre );
void ABVersTabRec(PArbre , int , TArbBin *, int );
TArbBin* ABVersTab(PArbre , int );
PArbre creerArbre(Element );
PArbre ajouterFilsGauche(PArbre , Element );
PArbre ajouterFilsDroit(PArbre , Element );
PArbre supprimerFilsGauche(PArbre a);
PArbre supprimerFilsDroit(PArbre );
void traiter(Element );
void parcoursPrefixe(PArbre );
void parcoursInfixe(PArbre );
void parcoursLargeur(PArbre );
void affArbreGraphique(PArbre, int );
void affArbre(PArbre );

#endif
```