TD4 : MPC

Exercice 26. [Librairie d'arbres binaires de flottants] Considérons le fichier d'entête `tree.h` suivant:

```
#ifndef TREE_H
#define TREE_H
typedef struct tree_s* tree_t;

void init(tree_t *); /* initialise un arbre */
tree_t new_tree(float); /* cree une feuille */
int is_empty(tree_t); /* vraie ssi l'arbre est vide */
int is_leaf(tree_t); /* vraie ssi l'arbre est une feuille */
tree_t fils_gauche(tree_t); /* renvoie le fils gauche de l'arbre */
tree_t fils_droit(tree_t); /* renvoie le fils droit de l'arbre */
float value_root(tree_t); /* renvoie la valeur de la racine de l'arbre */

void destroy(tree_t); /* désalloue tout l'arbre */

int cmp_tree(tree_t,tree_t); /* vrai ssi les deux arbres sont identiques */
int print_tree(tree_t); /* affiche les elements contenus dans l'arbre
de gauche a droite. Renvoie le nombre de noeuds de l'arbre  */

#endif /*  TREE_H */
```

Donnez le code du fichier `tree.c` associé (i.e. qu'il vous faut définir `struct tree_s` et l'ensemble des fonctions déclarées ici).

```
void init(tree_t* t){
    *t = NULL; // Arbre vide
    Return;
}
```

```
struct tree_s {
    float v;
    struct tree_s *fg;
    struct tree_s *fd;
}
```

```
tree_t new_tree(float val) {
    assert(0!=(tree_t t = (tree_t) malloc(sizeof(tree_s))))
    t->v = val;
    t->fg = t->fd = NULL
    Return t
}
```

```
void insert(float val, tree_t *t) {
    if ((*t) == NULL) {
        (*t) = new_tree(val);
        return;
    }
    if ((*t)->v < val) {
        insert(val, &((*t)->fg));
    }else {
        insert(val, &((*t)->fd));
```

```
int is_empty(tree_t t){
    Return t == NULL;
}

int is_leaf(tree_t t){
    Return !is_empty(t) &&
           is_empty(t->fg) &&
           is_empty(t->fd);
}
```

```
char* print_tree(tree_t t){
    if(is_empty(t)) printf("()"); Return;
    printf("(%d," t->val);
    print_tree(t->fg);
    printf(",");
    printf(t->fd); printf(")"); Return;}
```

```
void destroy(tree_t *t){
    if (is_empty(*t)) return;
    destroy(&(*t)->fg);
    destroy(&(*t)->fd);
    free(*t);
    *t = NULL;
```