

TD 2

Programmation par contrat

EXERCICE 1 - Type `arbre_binaire_t` et `expression`

Dans cet exercice, on suppose qu'une expression est un arbre binaire dont les noeuds internes valent soit '+' soit '*' et que les feuilles sont étiquetées par des caractères minuscules.

1. Écrire les contrats des fonctions suivantes

```
/*
*****
* Fichier arbre_binaire.h
*****
*/
#ifndef __ARBRE_BINAIRE_H__
#define __ARBRE_BINAIRE_H__

struct arbre_binaire_s{
    char valeur;
    struct arbre_binaire_s * gauche;
    struct arbre_binaire_s * droit;
};

typedef struct arbre_binaire_s * arbre_binaire_t;
typedef struct arbre_binaire_s * expression;

#define GAUCHE 0
#define DROITE 1

/**
 * Cette fonction renvoie un arbre binaire initialise a la valeur val.
 * @param val est la valeur que l'on souhaite donner au nouveau noeud.
 * @return un arbre binaire initialise a la valeur val.
 */
arbre_binaire_t creer_arbre(char val);

/**
 * Cree un arbre binaire dont la racine est etiquetee par 'val'
 * et dont les sous arbres sont 'gauche' et 'droit'.
 */
arbre_binaire_t inserer_racine(char val, arbre_binaire_t gauche, arbre_binaire_t droit);

/**
 * Cette fonction renvoie 1 si la valeur 'val' est stockee dans l'arbre et 0 sinon.
 */
int present(arbre_binaire_t a, char val);

/**
 * Cette fonction renvoie le nombre de noeud de l'arbre.
 */
int taille(arbre_binaire_t a);

/**
 * Cette fonction prend en entree un arbre et libere l'espace memoire de tous les noeuds
 * qu'il contient.
 */
void detruire_arbre(arbre_binaire_t a);

/**
 * Cree un arbre localement complet dont la racine est etiquetee par 'minuscule',
 * un caractere ecrit en minuscule.
 */
expression creer_feuille(char minuscule);

/**
 * Cree un arbre localement complet dont la racine est etiquetee par 'op'
 * et dont les sous arbres sont 'gauche' et 'droit'.
 */
expression inserer_operation(char op, expression gauche, expression droit);
```

```
/**
 * Cette fonction renvoie le nombre de noeud de l'arbre associe a l'expression
 */
int taille_expression(expression a);

/**
 * Cette fonction insere un noeud de valeur 'val' dans l'arbre binaire a.
 * On suppose que l'arbre est soit vide, soit une feuille.
 * Si l'arbre est vide, on renvoie une nouvelle racine
 * Si l'arbre est une feuille on ajoute une nouvelle feuille du cote 'cote'.
 */
arbre_binaire_t inserer_feuille(arbre_binaire_t a, char val, int cote);

#endif
```

2. Écrire les tests unitaires associés à ces contrats, excepté la fonction **detruire**.