

TP Algo 3

1. Arbre

Le fichier `tree.c` contient la déclaration d'un arbre ainsi qu'un main créant un arbre contenant une expression numérique.

1. Créez le code de la fonction `affiche_profondeur` qui affiche le contenu de tous les noeuds grace à un parcours en profondeur.
2. Créez le code de la fonction `affiche_largeur` qui affiche le contenu de tous les noeuds grace à un parcours en largeur.
3. Créez le code de la fonction `calcule_resultat` qui calcule le résultat de l'expression contenue dans l'arbre
4. Créez le code de la fonction `free_arbre` qui libère tous les noeuds alloués.

2. Table de hashage

Pour cet exercice utilisez le fichier `hashmap.c` Ce fichier contient la déclaration d'une structure représentant un fichier ainsi que son chemin.

L'objectif du TP sera de créer une hashmap associant un nom de fichier à cette structure.

1. Complétez le code de la fonction `table_hashage_add` pour qu'elle ajoute un fichier dans la table. Utilisez la fonction `hash` pour calculer la position du fichier dans le tableau.
2. Complétez le code de la fonction `table_hashage_search` pour qu'elle renvoie la structure correspondant au nom de fichier donné.
3. Complétez le code de la fonction `table_hashage_free` pour qu'elle libère la mémoire allouée par la table de hashage

3. Index

Pour cet exercice, utilisez le fichier `files.c` Ce fichier contient un programme qui accepte en argument le chemin d'un dossier et affiche tous les fichiers contenus dans ce dossier et ses sous dossiers.

1. Reprenez votre code de l'exercice précédent et enregistrez tous les fichiers trouvés par le programme dans une table Choisissez comme taille de tableau le nombre de fichiers trouvés.
2. Modifiez le programme pour qu'il permette à l'utilisateur de taper le nom d'un fichier, le programme devra lui afficher le chemin vers le fichier en cherchant dans la table de hashage.
3. Essayez sur un repertoire qui contient de nombreux fichiers.