

Projet Structures de Données et Algorithmiques

Le projet de SDA2 consistait à concevoir une calculatrice polonaise utilisant un arbre binaire. J'ai commenté au maximum mon programme afin qu'il soit le plus clair possible. Pour ce projet, j'ai du reprendre à zéro plusieurs fois, car je pensais pouvoir directement tout placer dans un arbre. Or, ceci était trop complexe, j'ai préféré créer une structure contenant les éléments qui me seront nécessaires pour la création de l'arbre.

Cette structure contient notamment un tableau où je stocke tous les éléments que l'utilisateur rentre. Ma deuxième structure est tout simplement l'arbre. J'ai suivi les conseils reçus en TP, j'ai créé un *char type* qui détermine si l'élément est un opérateur ou un chiffre. Ma fonction principale appelle les autres fonctions. Je la considère comme principale, car c'est elle qui va récupérer les éléments. Grâce à la fonction C *strtok* et la fonction C *fgets*, je stocke les éléments dans ma structure avec le tableau (et également le nombre d'éléments). Ensuite, j'ai créé une fonction récursive (au début j'avais une fonction itérative, mais après les conseils de mes camarades, j'ai constaté que la fonction était plus optimale en récursive) qui va créer l'arbre. Ici, j'utilise la fonction *strtod* car lorsque je suis confronté à un chiffre, ce dernier est le cas arrêtant ma récursivité. Je l'enracine dans l'arbre. Lorsque je rencontre un opérateur ne nécessitant deux opérandes, alors je rappelle la fonction sur les deux fils et j'enracine l'opérateur. Pour un opérateur nécessitant une seule opérande, je rappelle la fonction seulement pour le fils gauche.

Et enfin, j'ai une dernière fonction récursive également qui me permet de parcourir l'arbre. Elle effectue les calculs nécessaires et rappelle la fonction. Le cas qui permet de terminer la récursivité est également lorsque je rencontre un chiffre, alors je le renvoie tout simplement. Je me suis occupé des `\n` et `#` dans la fonction utilisant *fgets* et *strtok*. Il fait l'impasse si il rencontre `#` ou si il rencontre `\n`. J'ai effectué tous les tests présent sur le sujet et la démonstration fournie sur Moodle.

J'ai tout testé avec *valgrind* afin de vérifier qu'il n'y avait pas de soucis de mémoire. J'ai tenté de mettre en place la prise en compte des variables, mais mes tentatives n'ont pas vraiment abouti, donc je ne suis pas aller plus loin.