Cours de programmation séquentielle

La calculatrice

1 La calculatrice

1.1 Buts

— Créer et utiliser une librairie de pile.
— Implémenter l'algorithme de transformation d'une expression infixe en une expression postfixe.
— Implémenter l'algorithme d'évaluation d'une expression postfixe à l'aide de la pile d'entiers.

Vous avez vu tous ces algorithmes dans le cours d'algorithmique avec le Professeur Albuquerque et moi-même.

1.2 Enoncé

Il s'agit de réaliser un programme qui, en deux phases, évalue une expression arithmétique composée de nombres réels et des opérations : `+`, `-`, `*`, `/`, `^`.
La première phase transformera l'expression infixe en une expression postfixe. La seconde phase évaluera l'expression postfixe à l'aide d'une pile. Je rappelle ici pour la seconde fois, que ces deux algorithmes ont été vus aux cours.

1.3 Cahier des charges (pas forcément dans cet ordre)

— Utiliser une librairie de pile que vous avez déjà implémentée, mais en utilisant un nouveau type (les `char`), avec les fonctionnalités `push`, `pop`, `peek`, `is_empty`. Elle sera utilisée pour la gestion des parenthèses, ainsi que pour l'évaluation de l'expression postfixe.
— Créer une fonction qui prend en argument une expression infixe et qui retourne l'expression postfixe correspondante. Les expressions in/postfixes sont des chaînes de caractères (`char *`).
— Créer une fonction qui prend en argument une expression postfixe, l'évalue à l'aide de la pile, et retourne un double qui est le résultat de l'évaluation.

Vous devrez probablement implémenter d'autres fonctions, ne vous limitez pas à ce qui est conseillé ici si vous pensez que c'est nécessaire.

1.4 Indications

Pour simplifier la lecture de votre chaîne de caractère infixe, nous allons supposer que les opérandes sont des entiers à un seul chiffre de `0` à `9`. Les opérations sont `+`, `-`, `*`, `/`, `^` et que des parenthèses peuvent être présentes. Nous supposerons

également qu'il n'y a pas d'espaces dans la chaîne infixe et que finalement il n'y a pas non plus l'opérateur "négatif" (comme dans le nombre entier `-3`). La chaîne de caractère ne doit **pas** être lue à la ligne de commande.

1.5 Tests

Dans votre `Makefile` vous devez créer une cible `tests` qui exécute tous les tests de vos librairies. N'oubliez donc pas d'en faire!

Pour contrôler la qualité de votre programme, créez une batterie de tests. Vous devez également créer un programme qui illustre le fonctionnement de votre calculatrice en utilisant les expressions suivantes par exemple:

```
./calculator
7+2*3
13
./calculator
8-5-2
1
./calculator
(7-5)/(2-(1-9)^(2^3))
-0.00000011921
./calculator
2/(7-3*2)
2
./calculator
3*(5-(7-1)*3)/2
-19.5
./calculator
5^(3/2)+(5)^(2/3)
14.10
```