TP6 : Expressions préfixes et postfixes

1 Mini-calculatrice préfixe

On se propose ici de réaliser une petite calculatrice en C qui sera limitée aux chiffres. On supposera de plus que l'expression est donnée en notation préfixée sans espaces.
Exemple : $*4 + -623$ est équivalent à $4 * ((6 - 2) + 3)$.
On se propose d'écrire une fonction `int calcul(char *expr, int *pos)` qui calculera l'expression `expr` à partir de la position $*pos$ jusqu'à une certaine position. Cette nouvelle position sera mise dans (*pos). Par exemple pour l'expression donnée ci-dessus, l'évaluation pour (*pos) = :

- 0 donne $*4 + -623 = 28$. (*pos) vaudra en sortie 7
- 1 donne $4 = 4$. (*pos) vaudra en sortie 2
- 2 donne $+ - 623 = 7$. (*pos) vaudra en sortie 7
- 3 donne $-62 = 4$. (*pos) vaudra en sortie 6
- 4 donne $6 = 6$. (*pos) vaudra en sortie 5
- 5 donne $2 = 2$. (*pos) vaudra en sortie 6
- 6 donne $3 = 3$. (*pos) vaudra en sortie 7

La fonction demandée se déroulera de la manière suivante :

- Lire le caractère c situé à la position (*pos) et incrémenter (*pos).
- Si c est un chiffre alors retourner ce chiffre et finir la fonction
- Si c est une opération alors appelez récursivement cette fonction avec la nouvelle valeur de (*pos) pour obtenir la valeur de la première partie de l'opération. Appeler la fonction une deuxième fois pour en obtenir la deuxième partie. Réaliser l'opération lue et retourner cette valeur

2 Mini-calculatrice postfixe

De manière similaire, on va réaliser une calculatrice postfixe. L'expression postfixe équivalente à celle du premier exercice est $462 - 3 + *$. Pour calculer une telle expression on va se servir d'une pile : quand on lit une chiffre en l'ajoute au sommet de la pile, quand on lit un opérateur op, on enlève les deux chiffres c_1 et c_2 du sommet de la pile, on calcule $c_2 op c_1$ et on rajoute cette valeur en haut de la pile.
Répondre aux question suivantes :

- Tester cette méthode sur quelques expressions de votre choix.
- Créer un fichier *pile.h* contenant les entêtes des fonctions de gestion de pile (création, ajout et suppression d'un élément en sommet de pile) ainsi que la définition du type *pile*. Créer ensuite un fichier *pile.c* contenant le code de ces fonctions.
- Finalement, créer un fichier principal pour calculer la valeur d'une expression postfixe.
- Pour compiler, utiliser un *Makefile*