

TD11 : Pile et file

Exercice 1 : Validité du parenthésage d'une expression

Un problème fréquent pour les compilateurs et les traitements de texte est de déterminer si les parenthèses d'une chaîne de caractères sont équilibrées et proprement incluses les unes dans les autres. On désire donc écrire une fonction qui teste la validité du parenthésage d'une expression :

- on considère que les expressions suivantes sont valides : "()", "[([bonjour+]essai)7plus-];"
- alors que les suivantes ne le sont pas : "((", ")", "4(essai)".

Notre but est donc d'évaluer la validité d'une expression en ne considérant que ses parenthèses et ses crochets. On suppose que l'expression à tester est dans une chaîne de caractères, dont on peut ignorer tous les caractères autres que '(', '[', ']' et ')'. Écrire en langage algorithmique la fonction `valide(ch : chaîne de caractères) : booléen` qui retourne vrai si l'expression passée en paramètre est valide, faux sinon.

Exercice 2 : Notation polonaise

La notation polonaise, ou encore notation post-fixée, consiste à faire précéder les opérandes des opérateurs. Par exemple, au lieu d'écrire $42 + 13$, en notation polonaise on écrit $42\ 13\ +$. Un des avantages de cette notation est qu'elle rend inutile l'usage des parenthèses : pour $3 \times (42 + 13) - 5$, on note $3\ 42\ 13\ +\ \times\ 5\ -$ et il n'y a aucune ambiguïté. Dans cet exercice on va définir une fonction `polonaise` qui prendra en paramètres une expression post-fixée sous la forme d'un tableau de chaînes de caractères `{ "3", "42", "13", "+", "*", "5", "-" }` pour l'exemple précédent) et sa taille, et renverra le résultat (numérique) de l'évaluation de cette expression.

L'algorithme est le suivant. On lit les éléments du tableau un par un et on les empile sur une pile initialement vide. A chaque fois qu'on rencontre un opérateur, plutôt que de l'empiler, on l'applique aux deux derniers éléments de la pile et le résultat remplace ces deux derniers éléments.

- Créer une fonction `estOperateur` en C++ qui prend une chaîne de caractères en paramètre (classe `string`) et retourne vrai si cette chaîne représente un opérateur valide (ici seulement "+", "-", "/" ou "*"), et faux sinon.
- Créer une fonction `calcul` qui prend en paramètre une chaîne de caractères représentant un opérateur `op` et deux opérandes réelles `a` et `b`, et qui retourne la valeur numérique $a\ op\ b$.
- En utilisant les deux fonctions précédentes, écrire la fonction `polonaise`. Vous pourrez convertir une chaîne de caractères en un réel en utilisant les instructions suivantes :

```
double monReel = std::stod(maChaineRepresentantUnReel);
```