C++ Coding Challenge pour HIDDEN FOUNDERS

Elaboré par :

Mejri Badr Eddine **Junior Consultant / Ingénieur**

Tâche => écrire un évaluateur et analyseur d'expressions mathématiques :

Faisons un évaluateur d'expression qui prend des chaînes telles que "3+ (5+ (4 * 3)) - 12", les évalue et renvoie le résultat (8) ou bien des expressions plus compliquées comme "5+(4 *(5 - 2)/2) -2 " pour avoir (9) comme résultat . Les nombres positifs et négatifs, les quatre opérateurs arithmétiques et les parenthèses sont autorisés dans l'expression.

Il faut qu'on prend en compte les différents priorités pour avoir un résultat robuste et puissante d'où on va utiliser une approche connu dans l'algorithmique pour construire notre code source :

Analyse par descente récursive :

Une approche classique (appelée analyse par descente récursive) considère une expression comme une structure hiérarchique. Au niveau supérieur, une expression est composée de plusieurs sommets.

Les sommets sont séparés par des signes "+" et "-". Ils comprennent des nombres simples (5,2) et des expressions plus complexes (4* (5 – 2) /2) pour notre cas.

Chaque sommet, à son tour, est constitué de plusieurs facteurs (nous appellerons "facteurs" les éléments séparées par "*" et "/").

Le sommet "5" peut être considéré comme un cas particulier, il est composé d'un seul facteur (5) .D'autre part, le sommet "4* (5 - 2) /2" comprend trois facteurs.

Il existe 2 types de facteurs: les atomes et les sous-expressions entre parenthèses. Dans notre simple évaluateur d'expressions, un atome n'est qu'un nombre. Les sous-expressions peuvent être analysées de la même manière que l'expression entière (vous avez à nouveau trouvé les sommets à l'intérieur des crochets, puis des facteurs, puis des atomes(la même démarche de raisonnement précédente)).

Donc, nous avons une hiérarchie, où les atomes constituent le niveau le plus bas, puis ils sont combinés en facteurs, et les facteurs en sommets. Une méthode parfaite pour analyser d'une manière très simple une telle expression hiérarchique consiste à utiliser la récursivité.

Netographie :

https://en.wikipedia.org/wiki/Recursive_descent_parser