

Travaux Pratiques n°7

Cours d'Informatique de Deuxième Année

—Licence MI L2.2—

Expressions arithmétiques

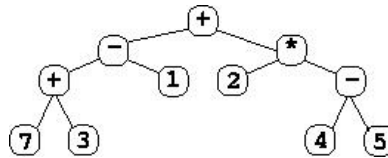
Dans ce TP, il s'agit de manipuler des expressions arithmétiques représentées sous forme d'arbres binaires. Une expression pourra contenir des entiers et des opérations. Pour limiter les problèmes d'arithmétique, on n'utilisera que les 3 opérateurs +, - et *.

Chaque expression constitue donc un arbre, ainsi

$((7 + 3) - 1) + (2 * (4 - 5))$

peut-être représenté par l'arbre **e** ci-dessous : Expression arithmétique

En terminologie d'arbres, les entiers sont des feuilles et les opérations sont des noeuds internes.



► Exercice 1.

Ecrire la structure (dynamique) *C* correspondante à une expression arithmétique.

Implémenter les fonctions d'allocation suivantes :

- `Expr creeExprInt(int i) ;`
- `Expr creeExprOpe(char o, Expr l, Expr r) ;`

Remarque : on utilisera deux constantes *INT* et *OPE* pour identifier la nature d'une expression.

► Exercice 2.

Écrire une procédure `afficheExpr1` pour obtenir l'affichage de **e** suivant :

+ - + 7 3 1 * 2 - 4 5

► Exercice 3.

Écrire une fonction `evalueExpr` qui évalue une expression arithmétique.

► **Exercice 4.**

Écrire une fonction `char* lectureFichier(char* nomFichier)` qui lit une chaîne de caractère dans un fichier.

► **Exercice 5.**

Écrire une fonction `lectureExpression` qui lit une expression arithmétique préfixe dans une chaîne de caractères passée en paramètre. Dans un premier temps les entiers saisis seront des chiffres positifs. Le format de vos fichiers devra être :

+ - + 7 3 1 * 2 - 4 5

On pourra coder cette fonction de manière récursive. On supposera que l'expression est complète.

► **Exercice 6.**

Modifier la fonction `afficheExpr1` pour qu'elle prenne en paramètre un descripteur de fichier de type `FILE *`.

Utiliser cette fonction pour écrire une fonction `ecrireFichier` qui écrit une expression arithmétique dans un fichier au format préfixe. Le fichier contiendra donc :

+ - + 7 3 1 * 2 - 4 5

► **Exercice 7.**

Modifier votre code pour manipuler des variables formelles VAR. Les variables seront identifiées par un seul caractère, de 'a' à 'z'. Programmer une fonction `diffExpr(Expr e, Char x)` qui calcule la dérivée partielle de l'expression `e` par rapport à `x`.

► **Exercice 8.**

Modifier le code de la fonction précédente pour prendre en compte tous les entiers. Le format du fichier sera donc (Attention à la position des espaces) :

+ - + -7 30 18 * -2 - -45 54