

- TP 2 -

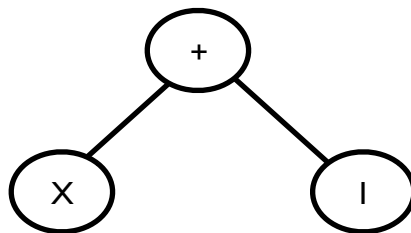
Expressions mathématiques

On souhaite écrire un programme permettant de manipuler des expressions mathématiques. On utilisera des arbres binaires pour représenter ces expressions mathématiques et les opérations sur les arbres pour les manipuler. On ne considère dans un premier temps que des expressions arithmétiques construites avec des entiers, la variables X et les opérateurs « $+$ » et « $*$ ». Toutes les expressions sont complètement parenthésées, c'est-à-dire toute expression construite avec un opérateur binaire est entourée par des parenthèses. Par exemple, l'expression $(X+1)$ est bien formée mais pas $2+3$.

Il faudra utiliser l'unité qui implémente les primitives des arbres binaires et qui a été développée précédemment et la compléter avec les opérations nécessaires pour la manipulation des expressions mathématiques.

1. Construction arbres

Proposer un type pour les valeurs contenues dans les noeuds de l'arbre et implanter dans un programme principal la construction d'un arbre arithmétique. Par exemple, l'expression « $(X+1)$ » est encodée par l'arbre binaire



2. Afficher l'expression

Implanter les fonctions permettant de générer les chaînes de caractères correspondant aux écritures prefixée et infixée de l'expression. Par exemple, si l'arbre ci-dessus est appelé `aplus1` dans le programme alors l'exécution du code

```

prefixe(aplus1);
tree_print(arb);

```

doit produire l'affichage

```

+ X 1
(X+1)

```

3. Calculer la valeur d'une expression

Implanter la fonction dont les entrées sont un entier N , un arbre binaire A correspondant à une expression mathématique et qui retourne la valeur en N de l'expression $f(X)$ décrite par A . Par exemple, l'exécution du code

```
printf("Valeur = %f", evaluate(aplus1,1));
```

doit produire l'affichage

```
Valeur = 2.0
```

4.Dériver une expression

Implanter la fonction dont l'entrée est un arbre binaire A décrivant une expression $f(X)$ et qui retourne l'arbre binaire de l'expression $f'(X)$. Par exemple, l'exécution du code

```
tree_print( derivate(aplus1) );
```

doit produire l'affichage

```
(1+0)
```

5.Expressions mathématiques

Compléter votre implantation pour permettre la construction d'expressions mathématiques contenant des fonctions telles que sinus et cosinus.

6.Construction à partir d'une chaîne de caractères

Implanter la fonction dont l'entrée est une chaîne de caractères décrivant une expression mathématique et qui retourne l'arbre binaire de cette expression. Par exemple, l'exécution du code

```
tree_print( build_tree("1+sin(X)",...);
```

doit produire l'affichage

```
(1+sin(X))
```