

**Structures de données et algorithmes**  
**Feuille de TP n°4**  
**Bibliothèque « arbres binaires » et expressions arithmétiques**

**Exercice. Bibliothèque de fonctions sur les arbres binaires**

Reportez-vous à la feuille de TP n°3. Finissez impérativement de coder dans le fichier `binary_tree.c` toutes les fonctions de `binary_tree.h`

**Exercice. Expressions arithmétiques**

Reportez-vous à l'exercice 1 de la feuille de de TD n°4. Écrivez dans un fichier appelé `expression.c` un programme C qui

- lit en argument sur la ligne de commande une chaîne de caractères *expr* implémentant une expression arithmétique préfixe, *i.e.* un mot de  $\mathcal{E}_A$ ;
- construit l'arbre syntaxique *t* de *expr* (en appelant la fonction `parse_expr`) puis l'affiche (en appelant la fonction `show_binary_tree` de la bibliothèque « arbres binaires »);
- évalue l'expression *expr* en appelant la fonction `eval_tree` et affiche le résultat;
- récupère dans une chaîne de caractères l'expression préfixe correspond à *t* (en appelant la fonction `tree_to_expr`) et vérifie que cette chaîne est bien égale à *expr*;
- libère toute la mémoire allouée sur le tas.

Écrivez, sur le modèle du fichier `item.h`, un fichier `item_char.h` dans lequel `item` redéfinit le type `char`.

Remplacez, dans le fichier `binary_tree.h`, la directive

```
#include "item.h" par  
#include "item_char.h".
```

Écrivez un fichier `makefile` configurant la commande `make` pour engendrer un exécutable à partir des fichiers sources `item_char.h`, `binary_tree.h`, `binary_tree.c` et `expression.c`.

Lancez la commande `make` pour tester l'exécutable.