

Compilation

TD1 & TP1: Analyse Lexicale

Exercice 1

Écrire un programme (f)lex qui permet de:

1. Compter le nombre de mots en affichant le nombre de caractères de chacun.
2. Compter le nombre de parenthèses ouvrantes et vérifier si le nombre de parenthèses fermantes correspond à celui des parenthèses ouvrantes.
3. Supprimer, du texte entré, tous les blancs et les tabulations se trouvant à la fin des lignes.
4. Remplacer toute suite de blancs et de tabulations restantes par un seul blanc.

Exercice 2

Écrire un programme (f)lex qui permet d'afficher les niveaux d'imbrication des parenthèses.

Exemple : l'expression (() (())) () doit donner 1 2 2 2 3 1

TP : Analyseur lexical

Ecrire à l'aide de (f)lex un analyseur lexical qui reconnaît les lexèmes suivants :

ENTIER : un nombre entier

REEL : un nombre réel

CHAÎNE : une chaîne de caractères entre guillemets

IDENTIFICATEUR : un identificateur type C

COMMENTAIRE : une suite de caractères commençant par #

IF : le mot réservé if

ELSE : le mot réservé else

WHILE : le mot réservé while

Votre analyseur devra :

- Ignorer les espaces, tabulations, passages à la ligne et les lignes vides,
- Afficher, pour chaque lexème, son type et sa valeur.
- Traiter des cas d'erreurs: afficher le numéro de ligne, le type de l'erreur et la chaîne concernée.

Par exemple, si le fichier en entrée est le suivant :

```
"ce fichier contient 8 lignes"
-2.3 5.6e8 +4.2 4
-.1e2 2.3er
ident truc1 x1 1x
"rigolo" #commentaire
"ah ah chaîne non terminée
-t
if machin while pif
#fin
```

Alors, votre analyseur devra indiquer, sur la sortie standard:

```
CHAINE : ce fichier contient 8 lignes
REEL : -2.300000
REEL : 560000000.000000
REEL : 4.200000
ENTIER : 4
REEL : -10.000000
IDENT : r
IDENT : ident
IDENT : truc1
IDENT : x1
ENTIER : 1
IDENT : x
CHAINE : rigolo
COMMENT: on ignore
IDENT : t
MCLEF : if
IDENT : machin
MCLEF : while
IDENT : pif
COMMENT: on ignore
```

Et le fichier contenant les messages d'erreurs contiendra :

```
2.3e *** ligne 3 *** exposant attendu
"ah ah chaine non terminee *** ligne 6 *** fin de chaine attendue
- *** ligne 7 *** nombre attendu
```