

Exercices – Analyse Lexicale

- 1° Ecrire un programme **flex** qui compte le nombre de mots d'un texte
- 2° (commande **wc** d'Unix) Ecrire un programme **flex** qui compte le nombre de caractères, de mots et de lignes du texte source
 - a) en restreignant l'alphabet aux lettre minuscules et majuscules, aux chiffres et aux caractères d'espacement.
 - b) en étendant l'alphabet à tous les symboles.
- 3° Ecrire un programme **flex** qui vérifie si une expression est correctement parenthésée.
- 4° Ecrire un programme **flex** qui remplace dans le texte source toutes les occurrences de **bon** par **mauvais** sauf lorsque c'est un sous mot d'un plus grand mot.
- 5° Ecrire un programme **flex** qui supprime dans le texte source tous les mots qui commencent par une voyelle.
- 6° Ecrire un programme **flex** qui "traduit" les abréviations contenues dans un texte donné. On considérera les abréviations suivantes :
 - **cad** : abréviation de *c'est à dire*
 - **ssi** : abréviation de *si et seulement si*
 - **thlang** : abréviation de *théorie des langages*
 - **afd** ou **AFD** : abréviation de *automate fini déterministe*
- 7° Ecrire un programme **flex** qui place correctement les majuscules et les minuscules dans un fichier texte donné.
- 8° Ecrire un programme **flex** qui remplace toute suite de blancs ou de tabulations par un seul blanc, supprime les espaces en fin de ligne, ainsi que les lignes vides.

- 9° Ecrire à l'aide de **flex** un analyseur lexical qui reconnaît les lexèmes suivants :
- ENTIER** : un nombre entier
 - REEL** : un nombre réel
 - CHAINE** : une chaîne de caractères entre guillemets
 - IDENTIFICATEUR** : un identificateur type langage C
 - COMMENTAIRE** : un commentaire est une ligne commençant par #
 - IF** : le mot réservé if
 - ELSE** : le mot réservé else
 - WHILE** : le mot réservé while

L'analyseur devra afficher, pour chaque lexème, son type et sa valeur. Les messages d'erreurs seront envoyés sur **stderr**.

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 trucl xl lx
"rigolo" #commentaire
"ah ah chaine non terminée
-t
if machin while pif
#fin
```

L'analyseur devra indiquer

```
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 : trucl
IDENT : xl
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 terminée *** ligne 6 *** fin de chaine
attendue
- *** ligne 7 *** nombre attendu
```