

Atelier Flex

Exercice 1

1. Créez un programme, qui extrait et affiche des nombres binaires valides dans une chaîne de caractères (nom du fichier de description at1_1.l, nom du programme at1_1.exe), le format d'affichage doit être comme suit :
Nombre binaire : 010101
Tester un programme avec différentes chaînes et noter les nombres binaires reconnus à partir des chaînes suivantes :
 - a. 10154697411014100000014
 - b. A10101b98741e418574
 - c. 10 1014
2. Créer un programme similaire au précédent et qui reconnaît des nombres octaux at1_2.exe)
3. Créer un programme at1_3 qui reconnaît des nombres hexadécimaux.
4. Créer at1_4 qui reconnaît des nombres binaires, octaux et hexadécimaux.
(NB : créer un analyseur insensible à la casse)
5. Testez votre programme avec les chaînes suivantes :
 - a. 1000111
 - b. 1234
 - c. fg
 - d. f10014108h100017845123Noter les résultats obtenus.

Exercice 2

Écrire un programme qui reconnaît les identificateurs C valides dans un fichier et les affiche.

Exercice 3

Écrire un parseur qui analyse un fichier et remplace toutes les occurrences du symbole @ par [at]

Exercice 4

Écrire un programme qui lit un fichier C et qui reconnaît les unités lexicales (ou lexèmes) suivantes :

- un type de données
- un identificateur
- un commentaire
- un opérateur
- un mot clé (for, break, exit, while, return)

Exercice 5

Considérons le codage - qui transforme des mots sur l'alphabet {a, b, c} en des mots sur l'alphabet {0, 1} défini de la façon suivante :

$f(a) = 0$

$f(b) = 01$

$f(c) = 11$.

Par exemple, $f(aabca) = 0001110$.

Écrire un programme FLEX qui code un mot lu sur l'entrée standard. En d'autres termes, le programme lit sur l'entrée standard un mot u sur l'alphabet {a,b,c} et écrit sur la sortie standard le mot v sur l'alphabet {0, 1} tel que $f(u)=v$.