
TD N°2

Exercice 01:

1. Donner une expression régulière qui valide une forme simplifiée des adresses électroniques.

Définition simplifiée :

- une adresse électronique est constituée d'un champ ou plusieurs suivies d'un @ suivi d'un champ ou plusieurs.
- un champ est constitué d'un caractère ou plusieurs (lettre, chiffre, -, _).

2. Les identifiants de fichiers distants se présentent, dans la plupart des cas, comme suit:
user@hostname:filename

Les parties des identifiants sont constituées de mots, qui sont des séquences d'une ou plusieurs lettres et chiffres. La partie *user* contient un seul mot. La partie *hostname*: se compose d'un ou plusieurs mots séparés par des points (comme dans *www.uio.no*), *filename* se compose d'un ou plusieurs mots séparés par des barres obliques (/) avec une barre oblique facultative au début et / ou à la fin. La partie « *user @* » est facultative et peut être omise. La totalité de « *user @ hostname*: » peut également être omise, y compris les deux-points de fin «:». La partie « *user @* » peut ne pas apparaître mais la partie *hostname*: doit toujours présente.

Exercice 02 : Soit le code Lex suivant :

```
%%  
a*b { printf( "1"); }  
ca { printf( "2"); }  
a*ca* { printf("3"); }  
%%
```

Pour la chaîne d'entrée "abcaacacaaabbaaabcaaca», donner la sortie de cet analyseur lexical?

Exercice 03: Soit le code Lex suivant :

```
%%  
aa { printf("1"); }  
b?a+b? { printf("2"); }  
b?a*b? { printf("3"); }  
.|n { printf("4"); }
```

1. Donnez un exemple d'une entrée de ce fichier qui produira 123 en sortie, ou expliquez pourquoi il n'en existe pas.
2. Donnez un exemple d'entrée à ce fichier qui produira 321 comme sortie, ou expliquez pourquoi il n'en existe pas.

Exercice 04:

Ecrire un programme lex qui remplace dans un texte chaque lettre par sa suivante en reversant la casse (a par B, B par c, Z par a).

Exemple :

Entrée : Je me sens VRAIMENT bien !

➔Sortie: kF NF TFOT wsbjnfou CIFO !