

# TP 1 : Lex et Yacc

November 2021

## 1 Test

Dans TP1/test

Créer un fichier test.l contenant :

```
%{
#include <stdlib.h>
}%
blancs      [ \t]+
chiffre     [0-9]
entier      {chiffre}+
%%
{blancs}    { printf("blanc\n"); }
{entier}    { printf("entier %d\n", atoi(yytext)); }
"+"         { printf("plus\n"); }
"-"         { printf("moins\n"); }
"*"         { printf("multiplié\n"); }
"/"         { printf("divisé\n"); }
"("         { printf("parenthèse o\n"); }
")"         { printf("parenthèse f\n"); }
"\n"        { return 0; }
%%
```

---

23/

Dans le terminal, faire;

```
flex test.l
```

```
gcc lex.yy.c -ll
```

```
./a.out
```

## 2 Exercice Lex

**Q1.** Écrire un fichier .l qui supprime tout les espaces en trop (seul un espace maximum entre deux caractères est accepté).

```
a          a aaa      a
a a aaa a
```

**Q2.** Écrire un fichier .l qui reconnaît les VARIABLE, NOMBRE, MOT:

```
ab2C 123 aBc
VARIABLE NOMBRE MOT
```

**Q3.** Écrire un fichier .l qui double les MOTS et affiche la longueur des NOMBRES:

```
mot un 1233 11111111
motmot unun 4 8
```

**Q4.** Écrire un programme Flex qui lit une suite de mots et affiche la taille du mot le plus long et la taille moyenne des mots de la suite de mots.

**Q5.** On veut écrire un programme flex qui doit reconnaître une adresser URL.

```
www.exemple.com
accepte
https://www.dauphine.psl.eu/
accepte
https://www.dauphine.psl.eu/formations/licences/informatique-des-organisations/l3-
informatique-et-mathematiques-pour-la-decision-et-les-donnees
accepte
ww.exemple.com
erreur
www.exemple
erreur
www.exemple..com
erreur
```

**Q6.** Les nombres Romain sont composés des symboles suivants :

Symbole	I	V	X	L	C	R	M
Valeur (en base 10)	1	5	10	50	100	500	1000

Sous leur forme standard ils doivent respecter les principes suivants;

- Ils se lisent de gauche à droite
- Un symbole (à l'exception du M) ne peut être employé plus de trois fois à la suite.
- Tout symbole qui suit un symbole de valeur supérieure ou égale s'ajoute à celui-ci (exemple : 6 s'écrit VI).
- Les symboles sont groupés par ordre décroissant, sauf pour les valeurs à retrancher (règle suivante). Ainsi 665 s'écrit DCLXV (toute autre forme comme VXLCD, DCXVL ou DCLVX n'est pas accepté).
- Certains symboles qui précèdent un autre symbole de valeur supérieure se soustraient à ce dernier, uniquement selon les règles suivantes :
  - I est retranché a V ou X si il est situé devant V ou X (IV = 4 et IX = 9).
  - X est retranché a L ou C si il est situé devant L ou C (XL = 40 et XC = 90).
  - C est retranché a D ou M si il est situé devant D ou D (CD = 400 et CM = 900).

On écrira XLIX et non IL pour 49 et XCIX et pas IC pour 99.

Sur <http://chiffresromains.babuo.com/chiffres-romains-de-1-a-5000> vous trouverez un liste de tous les chiffres Romain de 1 à 5000.

Il vous est demandé d'écrire un fichier lex "*romain.l*" qui reconnaîtra des chiffres Romain entre 1 et 4999. Le programme devra écrire "accepter" s'il détecte un unique chiffres Romain qui occupe toute une ligne (on pourra utiliser ^et \$). Il écrira "refusé" pour toute autre ligne. La procédure prendra fin lorsque l'on rencontrera le mot "FIN".

**Exemple :** ./a.out

CCCLXXXIII

accepté

CMLXXXIX

accepté

XI XXX

Refusé

17

Refusé

XXXXI

Refusé

MCMIVV

Refusé

FIN

exit

On notera que l'évaluation du chiffre Romain en base 10 n'est pas demandé.