



Rapport atelier 1: Prise en main de l'analyseur lexical Lex

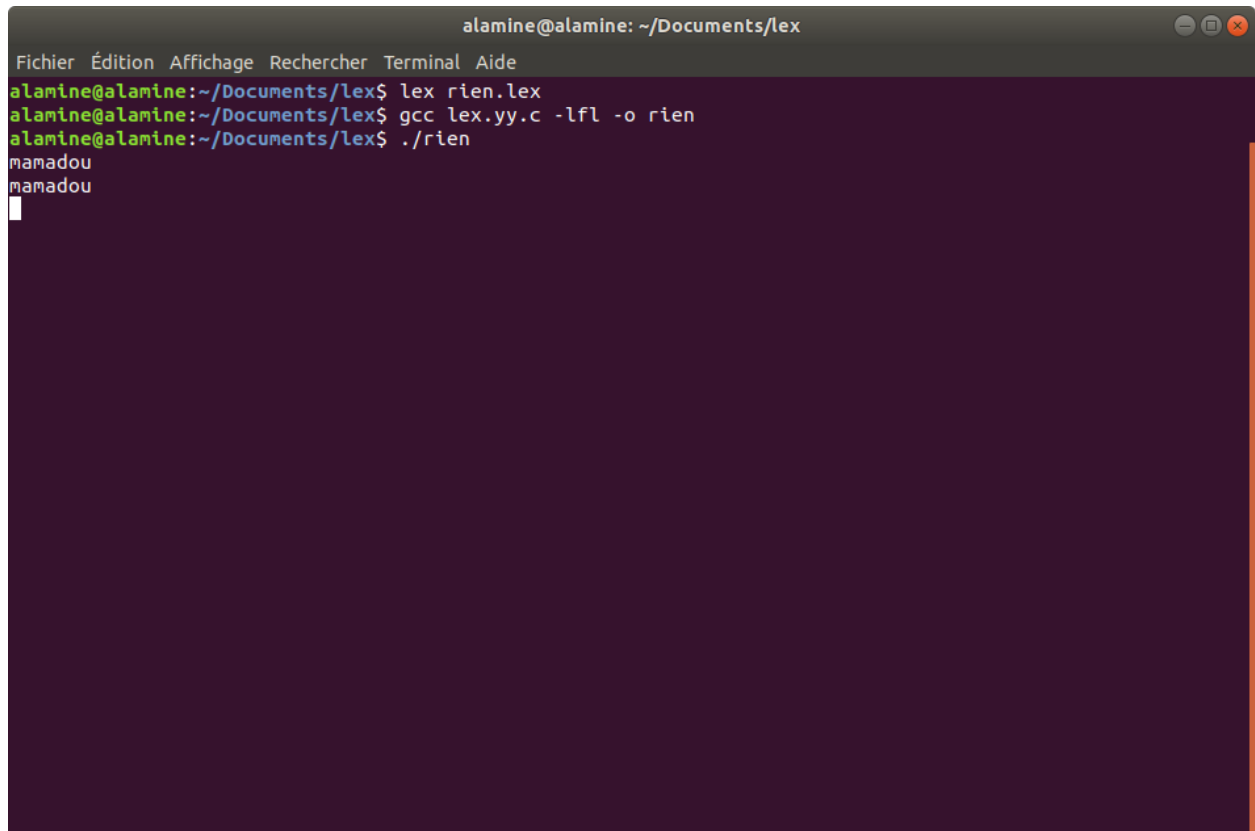
Membres du groupe:

Ndoumbé SOW

Mamadou DIOP



1. Tâche 1: Étude du format

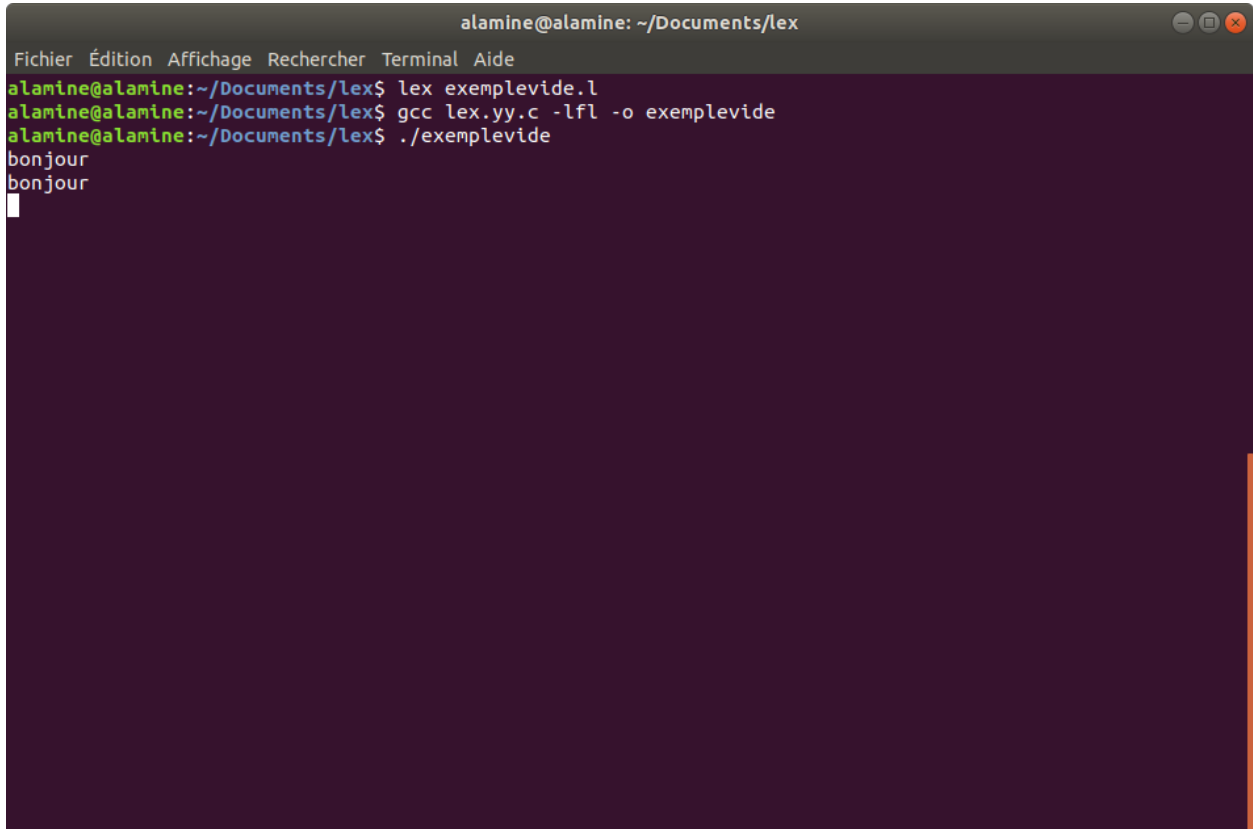


```
alamine@alamine: ~/Documents/lex
Fichier  Édition  Affichage  Rechercher  Terminal  Aide
alamine@alamine:~/Documents/lex$ lex rien.lex
alamine@alamine:~/Documents/lex$ gcc lex.yy.c -lfl -o rien
alamine@alamine:~/Documents/lex$ ./rien
mamadou
mamadou
```

The image shows a terminal window with a dark purple background. The title bar at the top reads "alamine@alamine: ~/Documents/lex". Below the title bar is a menu bar with the following items: "Fichier", "Édition", "Affichage", "Rechercher", "Terminal", and "Aide". The terminal content shows a series of commands and their outputs: the user runs "lex rien.lex", then "gcc lex.yy.c -lfl -o rien", and finally "./rien". The output of the last command is "mamadou" on two separate lines. The terminal window has standard Linux window controls (minimize, maximize, close) in the top right corner.

2. Tâche 2: Étude des définitions

2.1. Test du programme



```
alamine@alamine: ~/Documents/lex
Fichier Édition Affichage Rechercher Terminal Aide
alamine@alamine:~/Documents/lex$ lex exemplevide.l
alamine@alamine:~/Documents/lex$ gcc lex.yy.c -lfl -o exemplevide
alamine@alamine:~/Documents/lex$ ./exemplevide
bonjour
bonjour
```

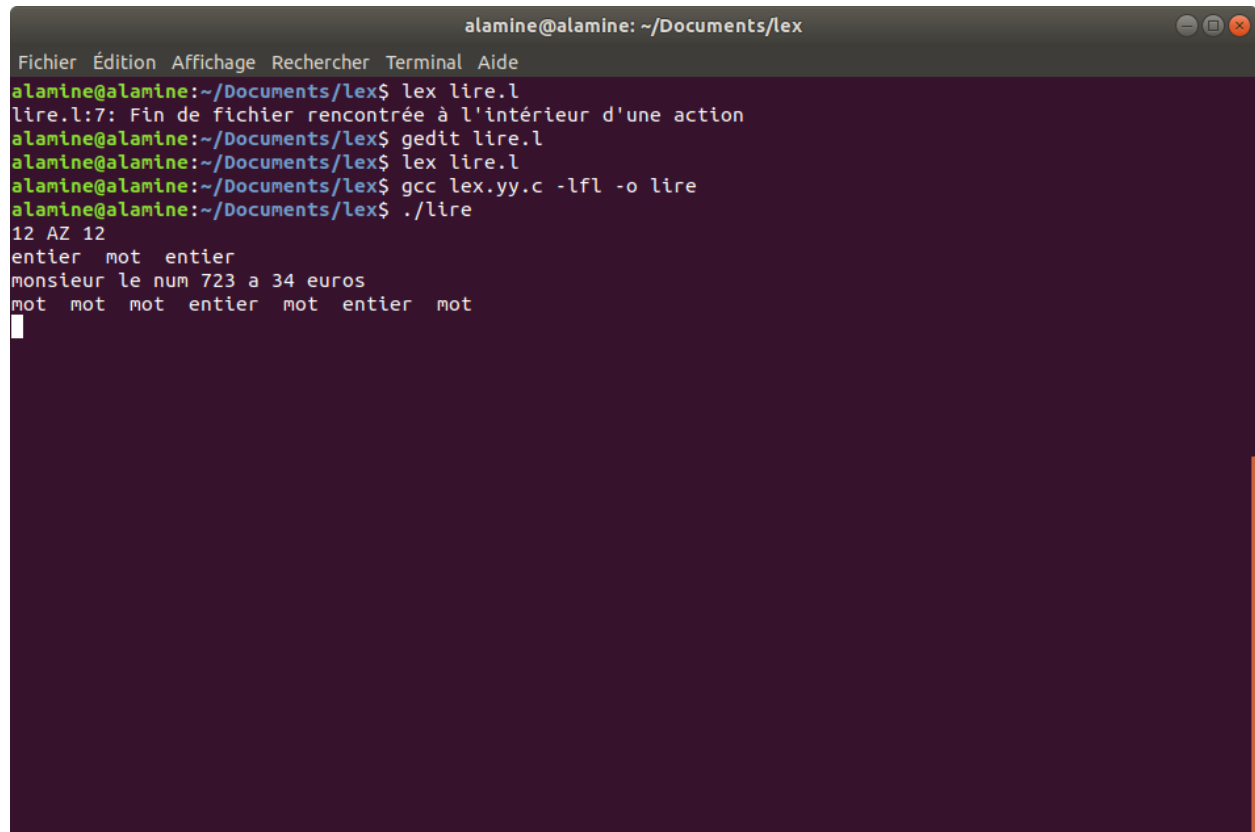
2.2. Analyse et interprétation

On peut remarquer qu'on a un programme qui ne contient que des définitions. Le programme retourne tout ce qu'on lui donne en entrée

3. Tâche 3: Étude des règles

3.1. Fichier *lire.l*

3.1.1. Test du programme



```
alamine@alamine: ~/Documents/lex
Fichier  Édition  Affichage  Rechercher  Terminal  Aide
alamine@alamine:~/Documents/lex$ lex lire.l
lire.l:7: Fin de fichier rencontrée à l'intérieur d'une action
alamine@alamine:~/Documents/lex$ gedit lire.l
alamine@alamine:~/Documents/lex$ lex lire.l
alamine@alamine:~/Documents/lex$ gcc lex.yy.c -lfl -o lire
alamine@alamine:~/Documents/lex$ ./lire
12 AZ 12
entier  mot  entier
monsieur le num 723 a 34 euros
mot mot mot entier mot entier mot
```

3.1.2. Analyse et interprétation

Le programme du fichier lire.l permet de reconnaître des mots et des entiers.

La ligne 1 à la ligne 3 correspond à la partie de définition et la ligne 6 et 7 à la partie des règles. La ligne 6 permet de reconnaître les entiers et la ligne 7 les mots.

3.1.3. Modifier lire.l

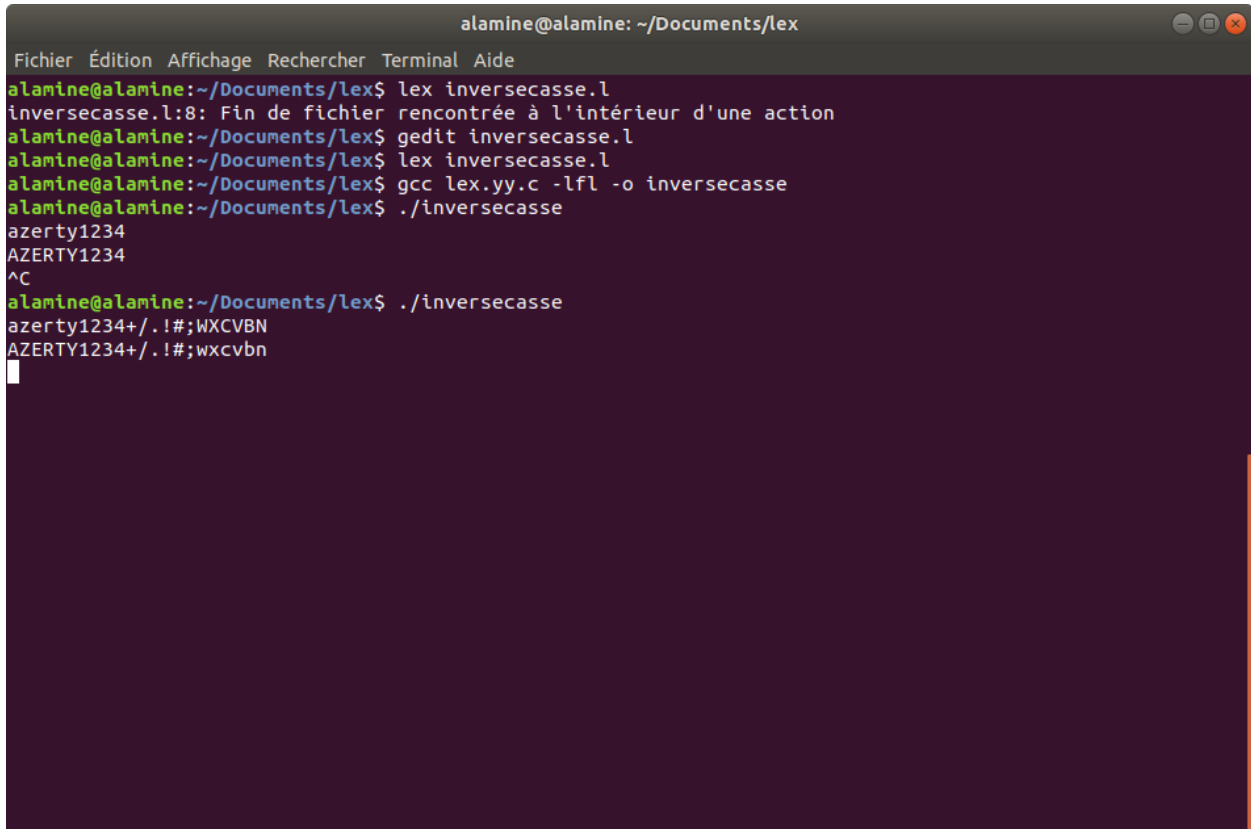
Modifier lire.l pour qu'il prenne en compte les nombres réels, les nombres avec virgule et les nombres exponentiels.

```
/* -- lire.lex --  
   Ce programme reconnait des mots et des entiers  
*/  
  
%%  
[0-9]+      printf("entier ");  
[0-9]+([.][0-9]+)      printf("nombre-décimal ");  
[0-9]+([.][0-9]+)?E([+-]?[0-9]+)      printf("nombre-exponentiel ");  
[a-zA-Z]+   printf("mot ");  
|
```

```
alamine@alamine: ~/Documents/lex
Fichier Édition Affichage Rechercher Terminal Aide
alamine@alamine:~/Documents/lex$ lex lire.l
alamine@alamine:~/Documents/lex$ gcc lex.yy.c -lfl -o lire
alamine@alamine:~/Documents/lex$ ./lire
12 abi 34.5 2.25 9E14
entier mot nombre-décimal nombre-décimal nombre-exponentiel
1.6E-12 23E+4
nombre-exponentiel nombre-exponentiel
```

3.2. Fichier *inversecasse.l*

3.2.1. Test du programme



```
alamine@alamine: ~/Documents/lex
Fichier Édition Affichage Rechercher Terminal Aide
alamine@alamine:~/Documents/lex$ lex inversecasse.l
inversecasse.l:8: Fin de fichier rencontrée à l'intérieur d'une action
alamine@alamine:~/Documents/lex$ gedit inversecasse.l
alamine@alamine:~/Documents/lex$ lex inversecasse.l
alamine@alamine:~/Documents/lex$ gcc lex.yy.c -lfl -o inversecasse
alamine@alamine:~/Documents/lex$ ./inversecasse
azerty1234
AZERTY1234
^C
alamine@alamine:~/Documents/lex$ ./inversecasse
azerty1234+/.!#;WXCvBN
AZERTY1234+/.!#;wxcvbn
```

3.2.2. Analyse et interprétation

A la fin de la reconnaissance d'une unité lexicale, la chaîne reconnue est la valeur de la variable `yytext` de type `char *`. Étant donné qu'on reconnaît caractère par caractère, le programme va changer la casse de chaque caractère rencontré d'où le résultat.

3.3. Fichier *parenthese.l*

3.3.1. Test du programme

```
alamin@alamin: ~/Documents/lex
Fichier Édition Affichage Rechercher Terminal Aide
alamin@alamin:~/Documents/lex$ lex parathese.l
parathese.l:18: Fin de fichier rencontrée à l'intérieur d'une action
alamin@alamin:~/Documents/lex$ gedit parathese.l
alamin@alamin:~/Documents/lex$ lex parathese.l
alamin@alamin:~/Documents/lex$ gcc lex.yy.c -lfl -o parathese
alamin@alamin:~/Documents/lex$ ./parathese
(
)
)
Votre parenthesage est incorrect
(
(

```

3.3.2. Analyse et interprétation

Le programme permet de vérifier le parenthésage. S'il y'a de parenthèse ouvrantes il n'affiche rien et s'il n'y en a pas et qu'il y'a une parenthèse fermante il affiche "votre parenthèse est incorrecte".

3.4. Fichier *compteur.l*

Modifier *compteur.l* pour qu'il puisse compter le nombre de lignes, caractères et de mots.


```

/* -- compteur.l --
    Ce programme compte le nb de lignes et
    le nb de caracteres
*/
int nbLignes = 0, nbCaracteres = 0, nbmots = 0;

%%
\n          ++nbLignes; ++nbCaracteres;
[a-z]+      ++nbmots; nbCaracteres += strlen(yytext);
.           ++nbCaracteres;

%%
void main()
{
    yylex();
    printf( "Nb de lignes = %d, Nb de caracteres = %d, Nb de mots = %d\n", nbLignes, nbCaracteres, nbmots );
}

```

```

alamine@alamine: ~/Documents/lex
Fichier Édition Affichage Rechercher Terminal Aide
alamine@alamine:~/Documents/lex$ lex compteur.l
alamine@alamine:~/Documents/lex$ gcc lex.yy.c -lfl -o compteur
alamine@alamine:~/Documents/lex$ ./compteur <test
Nb de lignes = 4, Nb de caracteres = 33, Nb de mots = 6
alamine@alamine:~/Documents/lex$ cat test
bonjour ffr vrfd
momo
comment
va
alamine@alamine:~/Documents/lex$

```