

(f)lex

(
(c)



(f)lex

GNU

Unix

lex

C

```
> (f)lex specification_file.l
```

C

lex.yy.c

:(f)lex

```
> gcc lex.yy.c -l (f)l
```

(specification_file.l)

.(...)

.1

```
%{  
declaration (in C Language) of variables, constants, ...  
%}  
  
declaration of regular definitions  
  
%%  
Translation rules  
%%
```

principal bloc & auxiliary functions (Optional)

(principal bloc & auxiliary functions)

main()



(f)lex

.2

:

(f)lex

\$ \ ^ [] { } < > - + . * / | ?

(f)lex

(f)lex

.(capital letters)

(small letters)

:

()

/ ^ \$

-1

\

[]

-2

—

-3

&	.	c
\+	c z a (small letters)	lc
“abc*+”	s	“s”
ab	.r2 r1	r1r2

<code>^ abc</code>	<code>.</code>	<code>^</code>
<code>abc\$</code>	<code>.</code>	<code>\$</code>
<code>[abc]</code>	<code>.s</code>	<code>[s]</code>
<code>[^abc]</code>	<code>.s</code>	<code>[^s]</code>
<code>a*</code>	<code>. 0 r</code>	<code>r*</code>
<code>a+</code>	<code>.r</code>	<code>r+</code>
<code>a?</code>	<code>.r</code>	<code>r?</code>
<code>a{3}</code>	<code>.r m</code>	<code>r{m}</code>
<code>a{2, 5}</code>	<code>.n m r</code>	<code>r{m, n}</code>
<code>a b</code>	<code>.r2 r1</code>	<code>r1 r2</code>
<code>ab / cd</code>	<code>.r2 r1</code>	<code>r1 / r2</code>
<code>(a b)? c</code>	<code>.r</code>	<code>(r)</code>
	<code>.</code>	<code>\n</code>
	<code>.Tabulation</code>	<code>\t</code>
	<code>.(f)lex</code>	<code>EOF</code>

(f)lex

.3

<code>char yytext[]</code>	<code>.</code>
<code>int yyleng</code>	<code>.</code>
<code>int yylex()</code>	<code>.yywrap()</code>
<code>int yywrap()</code>	<code>.) 0 . 1 (</code>
<code>int main()</code>	<code>.yylex()</code>
<code>unput(char c)</code>	<code>.(lex (f)lex)</code>
<code>int yylineno</code>	<code>.</code>

yyterminate()	.(lex (f)lex)
----------------	----------------

.4

-d	.(debug mode)
-i	.(capital letters) (small letters)
-l	.unix lex
-s	.

*.L .5

```
%{
int nbvowels, nbconsonant, nbPunct;
}%

Consonant      [b-df-hj-np-xz]
Punctuation    [,;:?!\\.]

%%
[aeiouy]        nbvowels++;
{consonant}     nbconsonant++;
{punctuation}   nbPunct++;
.|\\n          // Nothing to do
%%

main(){
    nbvowels = nbconsonant = nbPunct = 0;
    yylex();
    printf("No %d vowels, %d consonants and %d punctuations.\n",
nbvowels, nbconsonant, nbPunct);
}
```

p

. * 0

```
Digit      [0-9]
Integer    {digit}+

%%
{integer}   printf("I deny the integers");
^p(\\.)*\\n  printf("I hate lines starting with p\\n");
.           if (yytext[0]=='o') printf("**")
            else printf("%c", yytext[0]);
```