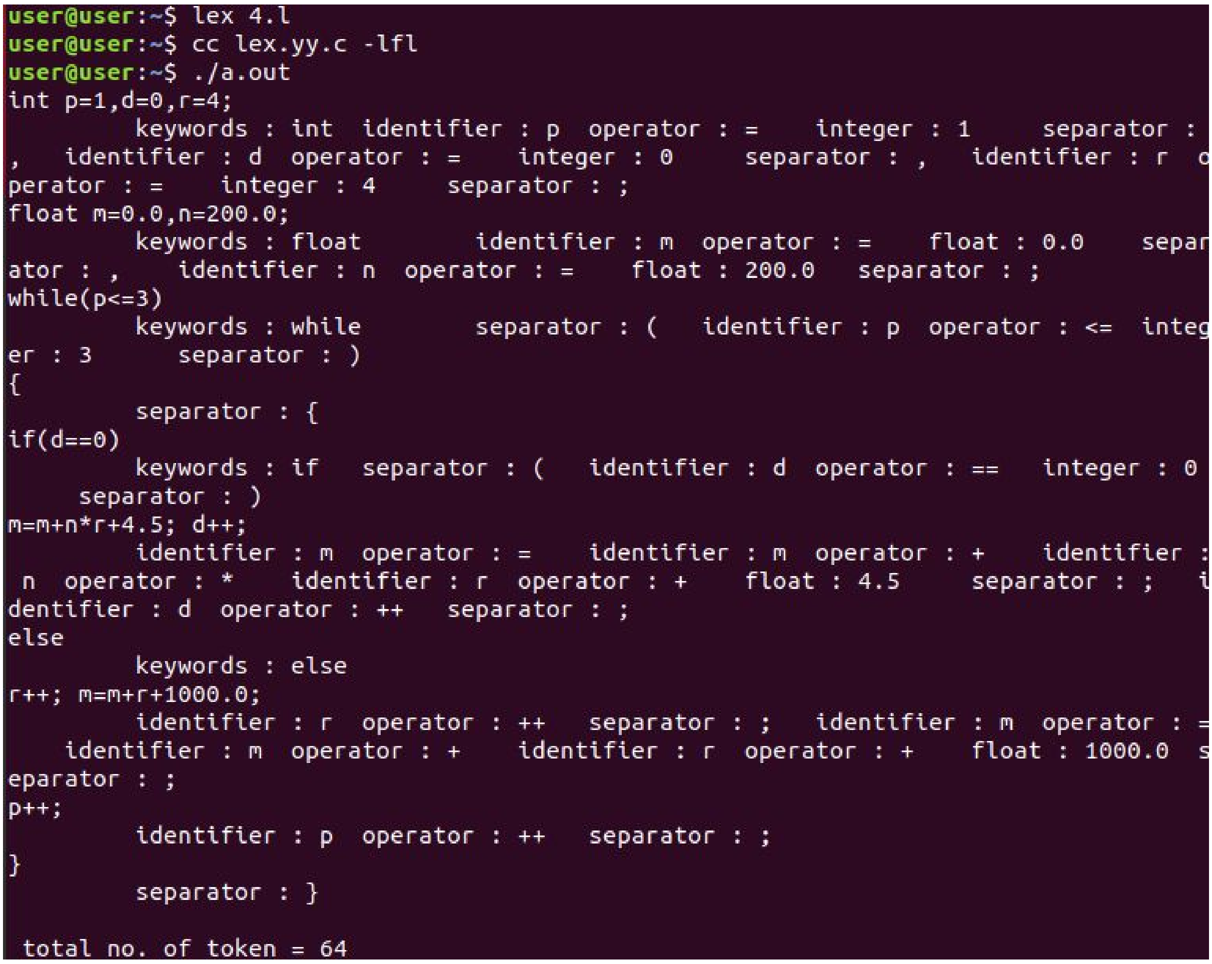
```
/*Lex code to count total number of tokens */

%{
int n = 0 ;
%}

// rule section
%%

//count number of keywords
"while"|"if"|"else" {n++;printf("\t keywords : %s", yytext);}

// count number of keywords
"int"|"float" {n++;printf("\t keywords : %s", yytext);}

// count number of identifiers
[a-zA-Z_][a-zA-Z0-9_]* {n++;printf("\t identifier : %s", yytext);}

// count number of operators
"<="|"=="|"="|"++"|"-"|"*"|"+" {n++;printf("\t operator : %s",
yytext);}

// count number of separators
[(){}|, ;] {n++;printf("\t separator : %s", yytext);}

// count number of floats
[0-9]*"."[0-9]+ {n++;printf("\t float : %s", yytext);}

// count number of integers
[0-9]+ {n++;printf("\t integer : %s", yytext);}


.    ;
%%


int main()
```



```
user@user:~$ lex 4.l
user@user:~$ cc lex.yy.c -lfl
user@user:~$ ./a.out
int p=1,d=0,r=4;
         keywords : int  identifier : p  operator : =    integer : 1     separator :
,   identifier : d  operator : =    integer : 0     separator : ,   identifier : r  o
perator : =    integer : 4     separator : ;
float m=0.0,n=200.0;
         keywords : float        identifier : m  operator : =    float : 0.0    separ
ator : ,    identifier : n  operator : =    float : 200.0   separator : ;
while(p<=3)
         keywords : while        separator : (   identifier : p  operator : <=  integ
er : 3      separator : )
{
         separator : {
if(d==0)
         keywords : if   separator : (   identifier : d  operator : ==   integer : 0
     separator : )
m=m+n*r+4.5; d++;
         identifier : m  operator : =    identifier : m  operator : +    identifier :
 n  operator : *    identifier : r  operator : +    float : 4.5     separator : ;   i
dentifier : d  operator : ++   separator : ;
else
         keywords : else
r++; m=m+r+1000.0;
         identifier : r  operator : ++   separator : ;   identifier : m  operator : =
    identifier : m  operator : +    identifier : r  operator : +    float : 1000.0  s
eparator : ;
p++;
         identifier : p  operator : ++   separator : ;
}
         separator : }

 total no. of token = 64
```

```
{

    yylex();

    printf("\n total no. of token = %d\n", n);

}
```