

Execution

```
lex lex.in  
cc main.c -ll  
a.out < lex.data
```

Sample output

```
Token type=1, token=ABC  
Token type=3, token==  
Token type=1, token=ABC  
Token type=3, token=+  
Token type=2, token=234  
Token type=3, token=/  
Token type=1, token=abc123  
Token type=3, token=*  
Token type=2, token=123  
Token type=1, token=BC  
Token type=1, token=LastLine  
Token type=3, token==  
Token type=3, token==  
Token type=2, token=123  
Token type=1, token=ABC  
End of file  
Token type=0, token=
```

Sample use of LEX

Lex input -- file lex.in

```
L    [A-Z,a-z]
D    [0-9]
LD   [A-Z,a-z,0-9]
S    [=,+,-,*,/]
B    [" "]
%%
{L}{LD}*    {return(1);}
{D}+    {return(2);}
{S}    {return(3);}
{B}    ;
;
%%
/*set end of file for calling routine*/
yywrap()
    {printf("End of file0);
    yytext[0] = ' 00';
    yytext[1] = '#';
    return(1);}
```

main C program -- file main.c

```
/* include yylex from lex */
#include "lex.yy.c"

main()
    {int tokentype;
    /* yywrap sets end of file character */
    while(yytext[1] != '#')
        /* get token */
        {tokentype=yylex();
        printf("Token type=%d, token=%s0,tokentype,yytext);
        }
    }
```

Sample data -- file lex.data

```
ABC=ABC+234/abc123*123 BC
LastLine ==123ABC
```