ASSIGNMENT NO:2

Title : Design and Develop Lexical Analyzer.

Name   : Sarang Barshikar
Roll No   : 48

CODE:

%{

#include<stdio.h>

%}

%%

[\t] ;

[0-9] {printf(" \n%s is an digit",yytext);}

[a-zA-Z] {printf("\n%s is an identifier",yytext);}

[@|%|^|&|(|)|.|{|}] {printf("%s is special symbols ",yytext);}

[+|-|*|/|=] {printf("%s is operator",yytext);}

[$] return 0;

[#] {printf("%s is preprocessor",yytext);}

[;|,] {printf("%s is delimiter",yytext);}

[<<]; {printf("%s is extraction operator",yytext);}

[>>]; {printf("%s is insertion operator",yytext);}

```
if|else|switch|case|break {printf("%s is a keyword");}

.|\ ECHO;

%%

main()

{

yylex();

}


int yywrap()

{

return 0;

}
```
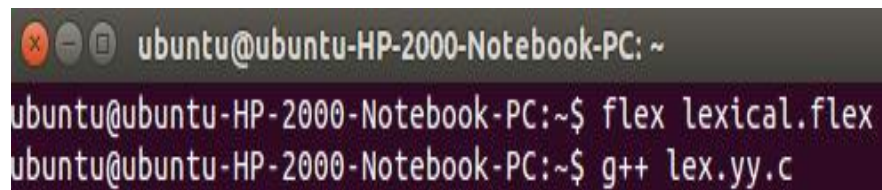
OUTPUT:


```
ubuntu@ubuntu-HP-2000-Notebook-PC: ~

ubuntu@ubuntu-HP-2000-Notebook-PC:~$ flex lexical.flex
ubuntu@ubuntu-HP-2000-Notebook-PC:~$ g++ lex.yy.c
```

```
ubuntu@ubuntu-HP-2000-Notebook-PC:~$ ./a.out
1

1 is an digit
9

9 is an digit
a

a is an identifier
z

z is an identifier
A

A is an identifier
Z

Z is an identifier
@
@ is special symbols
%
% is special symbols
&
```

```
ubuntu@ubuntu-HP-2000-Notebook-PC: ~
&
& is special symbols
^
^ is special symbols
|
| is special symbols
{
{ is special symbols
}
} is special symbols
+
+ is operator
=
= is operator
*
* is operator
-

#
# is preprocessor
;
; is delimiter
,
, is delimiter
```