Practical No :-3
Name:- Bhakti Bibhishan Bhoskar   **Roll No:-13125**

Code:-
Ass3.l
```
%{
#include<stdio.h>
#include "y.tab.h"
%}
%%
int|float|char return BUILTIN;
"," return COMMA;
";" return SC;
[a-zA-Z]+[a-zA-Z0-9]* return ID;
"\n" return NL;
%%
```

Ass3.y
```
%{
#include<stdio.h>
%}
%token ID BUILTIN SC NL COMMA
%%
start:BUILTIN varlist SC NL     {printf("\nValid
syntax\n");}
;
varlist:ID|varlist COMMA ID
;
%%
void yyerror()
{printf("\nerror\n");}
int yywrap()
{return 1;}
main()
{yyparse();}
```

Hello.txt
```
public class Helloworld {
public static void main(string args
[]){int i = 0;
      System.out.println("Hello
World");
}}
```

Token.l
```
%{%}
quotes ["""]
comment ("//".*)
comment2 ("/*".*"*/")
%%
public|private|protected {printf("%s : is a Access
specifier\n",yytext);}
class|System|out|println|void|main|static
{printf("%s : is a Keyword\n",yytext);}
int|double|char|String {printf("%s : is a
DataType\n",yytext);}
[A-Z]+[a-z]* {printf("%s : is a Class-
Name\n",yytext);}
[A-Z]*[a-z]+ {printf("%s : is a Variable\n",yytext);}
"+"|"-"|"<"|">"|"!"|"%"|"&"|"^"|"*"|"?"|"="
{printf("%s : is a Operator\n",yytext);}
"["|"]"|"{"|"}"|"("|")"  {printf("%s : is a
Bracket\n",yytext);}
","|"."|":"|";" {printf("%s : is a
Delimiter\n",yytext);}
[0-9]+ {printf(" %s : is a Number\n",yytext);}
{quotes} {printf(" %s : is a Quote\n",yytext);}
{comment} {printf(" %s : is a Single line
Comment\n",yytext);}
{comment2} {printf(" %s : is a Multiline
Comment\n",yytext);}
%%
main(argc,argv)
int argc;
char **argv;
{if(argc >1){
            FILE *file;
            file=fopen(argv[1],"r");
            if(!file){
                    fprintf(stderr,"Could not
open %s\n",argv[1]);
                    exit(1);}
            yyin=file;}
      yylex();}
int yywrap(){
return 1;
}
```

Output:-

Lexeme

public	Keyword
class	Keyword
Helloworld	Identifier
{	Separator
public	Keyword
static	Keyword
void	Keyword
main	Identifier
(	Separator
string	Identifier
args	Identifier
[	Separator
]	Separator
)	Separator
{	Separator
int	Keyword
i	Identifier
=	Operator
0	Literal
;	Separator
System	Identifier
.	Operator
out	Identifier
.	Operator
println	Identifier
(	Separator
"Hello World"	Literal
)	Separator
;	Separator
}	Separator
}	Separator