Date: 21/07/2025

Experiment No: 01

Aim: To implement Lexical Analyzer Using Lex Tool

Code:

```
%{
#include <stdio.h>
int COMMENT = 0;
%}

identifier [a-zA-Z_][a-zA-Z0-9_]*
whitespace [ \t\n]+

%%
#.* { printf("\n%s is a preprocessor directive", yytext); }

"/*" { COMMENT = 1; printf("\n\t%s is a COMMENT START", yytext); }
"*/" { COMMENT = 0; printf(" COMMENT END"); }

int|float|char|double|while|for|struct|typedef|do|if|break|continue|void|switch|return|else|goto {
 if (!COMMENT) printf("\n\t%s is a keyword", yytext); }

{identifier}\( { if (!COMMENT) printf("\nFUNCTION\n\t%s", yytext); }
\{ { if (!COMMENT) printf("\nBLOCK BEGINS"); }
\} { if (!COMMENT) printf("BLOCK ENDS "); }

{identifier}(\[[0-9]*\])? { if (!COMMENT) printf("\n%s IDENTIFIER", yytext); }
\".*?\" { if (!COMMENT) printf("\n\t%s is a STRING", yytext); }
[0-9]+ { if (!COMMENT) printf("\n%s is a NUMBER", yytext); }
```

```
\( { if (!COMMENT) ECHO; }
\)(\:)? { if (!COMMENT) { printf("\n\t"); ECHO; printf("\n"); } }

= { if (!COMMENT) printf("\n\t%s is an ASSIGNMENT OPERATOR", yytext); }

"<="|">="|"<"|">"|"==" { if (!COMMENT) printf("\n\t%s is a RELATIONAL OPERATOR", yytext); }

{whitespace} /* skip whitespace */

. { if (!COMMENT) ECHO; }  /* catch all other characters */

%%

int main(int argc, char **argv)
{
    FILE *file = fopen("var.c", "r");
    if (!file) {
        printf("Could not open the file\n");
        return 1;
    }
    yyin = file;
    yylex();
    printf("\n");
    fclose(file);
    return 0;
}

int yywrap()
{
    return 1;
}
```

Variable.c

```c
#include<stdio.h>
#include<conio.h>
void main()
{
int a,b,c;
a=1;
b=2;
c=a+b;
printf("Sum:%d",c);
}
```

Output:

```
ubuntu:~$ flex ex1.l
ubuntu:~$ gcc lex.yy.c -o lexer -lfl
ubuntu:~$ ./lexer

#include<stdio.h> is a preprocessor directive
#include<conio.h> is a preprocessor directive
        void is a keyword
FUNCTION
        main(
        )

BLOCK BEGINS
        int is a keyword
a IDENTIFIER,
b IDENTIFIER,
c IDENTIFIER;
a IDENTIFIER
        = is an ASSIGNMENT OPERATOR
1 is a NUMBER;
b IDENTIFIER
        = is an ASSIGNMENT OPERATOR
2 is a NUMBER;
c IDENTIFIER
        = is an ASSIGNMENT OPERATOR
a IDENTIFIER+
b IDENTIFIER;
FUNCTION
        printf(
        "Sum:%d" is a STRING,
c IDENTIFIER
        )
;BLOCK ENDS
ubuntu:~$
```

Result: The code has been executed and output displayed successfully.