

Lab program 3:

3. Write a program to implement the Lexical Analyzer Using LEX Tool.

Aim: Program to implement the Lexical Analyzer Using LEX Tool.

Program:

```
/* program name is lexp.l */
%{
/* program to recognize a c program */
int COMMENT=0;
}%
identifier [a-zA-Z][a-zA-Z0-9]*
%%
#.* { printf("\n%s is a PREPROCESSOR DIRECTIVE",yytext);}
int |float |char |double |while |for |do |if |break |continue |void |switch |case |long |struct |const |
typedef |return |else |
goto { printf("\n\t%s is a KEYWORD",yytext);}
"/*" { COMMENT = 1;}
/*{ printf("\n\n\t%s is a COMMENT\n",yytext);}*/
"*/" { COMMENT = 0;}
/* printf("\n\n\t%s is a COMMENT\n",yytext);}*/
{identifier}\( {if(!COMMENT)printf("\n\nFUNCTION\n\t%s",yytext);}
\{ {if(!COMMENT) printf("\n BLOCK BEGINS");}
\} {if(!COMMENT) printf("\n 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\t%s is a NUMBER",yytext);}
\(\)|\)? {if(!COMMENT) printf("\n\t");ECHO;printf("\n");}
\(\{ ECHO;
= {if(!COMMENT)printf("\n\t%s is an ASSIGNMENT OPERATOR",yytext);}
\<= |
\>= |
\< |
== |
\> {if(!COMMENT) printf("\n\t%s is a RELATIONAL OPERATOR",yytext);}
%%
int main(int argc,char **argv)
{
if (argc > 1)
{
FILE *file;
file = fopen(argv[1],"r");
if(!file)
{
printf("could not open %s \n",argv[1]);
exit(0);
}
yyin = file;
}
yylex();
printf("\n\n");
```

```
return 0;
} int yywrap()
{
return 0;
}
```

Input:

```
$vi var.c
#include<stdio.h>
main()
{
int a,b;
}
```

Output:

```
$lex lex.l
$cc lex.yy.c
$./a.out var.c
#include<stdio.h> is a PREPROCESSOR DIRECTIVE
FUNCTION
main ()
BLOCK BEGINS
int is a KEYWORD
a IDENTIFIER
b IDENTIFIER
BLOCK ENDS
```