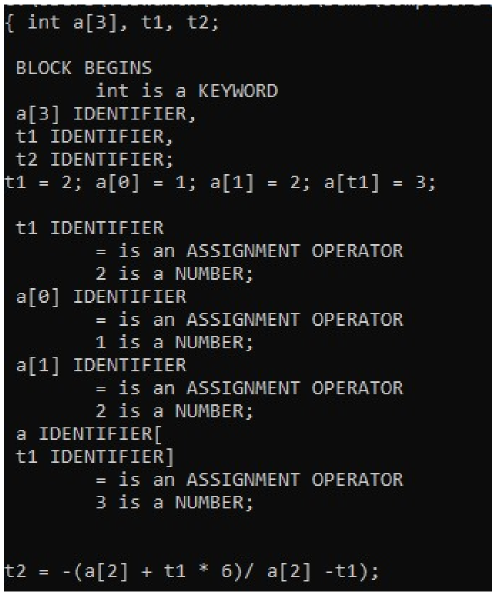
Week-5

Implementation of lexical analyzer using LEX

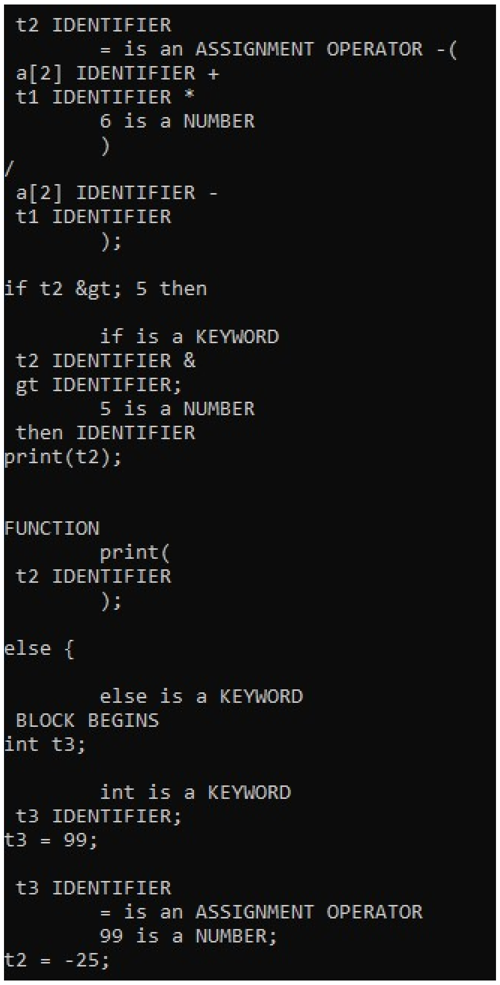
```
/* program name is lexp.l */
%{

int COMMENT=0;
int cnt=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;}
```

```
"*/" {COMMENT = 0; cnt++;}
{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 Total No.Of comments are %d",cnt);
return 0;
}

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

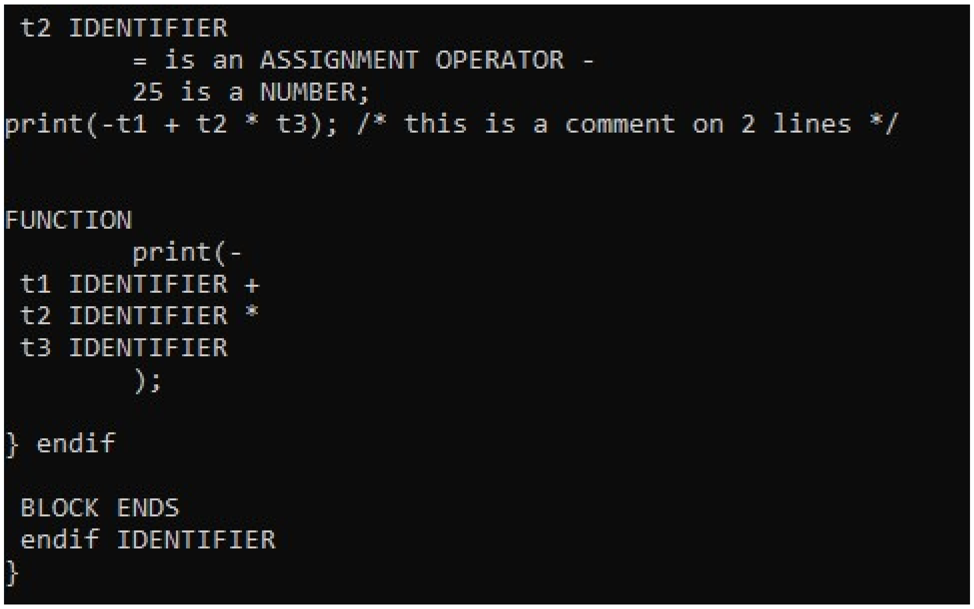
Output:



```
{ int a[3], t1, t2;

 BLOCK BEGINS
        int is a KEYWORD
 a[3] IDENTIFIER,
 t1 IDENTIFIER,
 t2 IDENTIFIER;
t1 = 2; a[0] = 1; a[1] = 2; a[t1] = 3;

 t1 IDENTIFIER
        = is an ASSIGNMENT OPERATOR
        2 is a NUMBER;
 a[0] IDENTIFIER
        = is an ASSIGNMENT OPERATOR
        1 is a NUMBER;
 a[1] IDENTIFIER
        = is an ASSIGNMENT OPERATOR
        2 is a NUMBER;
 a IDENTIFIER[
 t1 IDENTIFIER]
        = is an ASSIGNMENT OPERATOR
        3 is a NUMBER;


t2 = -(a[2] + t1 * 6)/ a[2] -t1);
```



```
 t2 IDENTIFIER
        = is an ASSIGNMENT OPERATOR -(
 a[2] IDENTIFIER +
 t1 IDENTIFIER *
        6 is a NUMBER
        )
/
 a[2] IDENTIFIER -
 t1 IDENTIFIER
        );

if t2 &gt; 5 then

        if is a KEYWORD
 t2 IDENTIFIER &
 gt IDENTIFIER;
        5 is a NUMBER
 then IDENTIFIER
print(t2);


FUNCTION
        print(
 t2 IDENTIFIER
        );

else {

        else is a KEYWORD
 BLOCK BEGINS
int t3;

        int is a KEYWORD
 t3 IDENTIFIER;
t3 = 99;

 t3 IDENTIFIER
        = is an ASSIGNMENT OPERATOR
        99 is a NUMBER;
t2 = -25;
```



```
 t2 IDENTIFIER
        = is an ASSIGNMENT OPERATOR -
        25 is a NUMBER;
print(-t1 + t2 * t3); /* this is a comment on 2 lines */


FUNCTION
        print(-
 t1 IDENTIFIER +
 t2 IDENTIFIER *
 t3 IDENTIFIER
        );

} endif

 BLOCK ENDS
 endif IDENTIFIER
}
```