Week - 5 :

1) Write a lex program to take the file name as an input. Need to apply the lexical analyser on the input file contents. If the lexeme is part of single line comment or multiline comment need to ignore that lexeme. If it is not part of comment and it is any one of the following then display it as a keyword (int, char, float, double, void, return). If the lexeme is not a keyword and starts with a letter followed by a letter / digit combination then display it is an identifier, insert the identifier in the symbol table. If the lexeme is an integer display it as an integer, if the lexeme is a floating point value, display it as float.

Code:

```
digit [0-9]*
id [a-zA-Z][a-zA-Z0-9]*
num [0-9]\.[0-9]
%{
 #include<stdio.h>
 #include<string.h>
 int i=0, j=0, cnt=0, n=0, com=0, scom=0;
 char st[10][10];
 int ~~exists~~ lookup (char st[10][10], char* id, int n);
%}
```

```
\n {scom=0, n++; }
"//" {scom=1; printf("\n Single line comment \n \n"
"/*" {com=1; printf ("\n Comment Start \n"); }
"*/" {com=0; printf("\n Comment end \n"); }
int|float|char|double|void|main {
    if(!com&&!scom) printf("\n %s is keyword", yytext);
"<=" {if(!com && !scom) printf("\n %s is Relational
operator less than or equal to", yytext); }
"<" {if (!com && !scom) printf("\n %s is Relational
operator less ~~greater~~ than ~~or equ~~", yytext); }
">=" {if (!com) printf ("\n %s is Relational operator
greater than or equal to", yytext); }
">" {if(!com && !scom) printf ("~~$~~\n %s is Relational
Operator greater than"~~$~~, yytext); }
"==" {if(!com && !scom) printf("\n %s is Relational
operator equal to", yytext); }
"!=" {if (!com && !scom) printf("\n %s is Relational
operator not equal to", yytext); }
{id} {if (!com & !scom) printf("\n %s is identifier",
yytext); }
```

```
{num} {if(!com &&!scom) printf("\n %s is float",
  yytext); }
{digit} {if(!com &&!scom) printf("\n %s is digit
  yytext); }
@ yyterminate();
%%
int main (int argc, char* argv[])
{  if (argc!=2) {
     fprintf (stderr,"Usage: %s <input_file>\n",
                                    argv[0]);
     getchar ();
     return 1;
  }
  FILE* input_file = fopen (argv[1], "r");
  if (!input_file) {
      perror ("Failed to open input file");
      return 1;
  }
  yyin= input_file;
  yylex();
printf ("\n\n\n no of lines = %d \n \n", n);
fclose (input_file);
getchar();
return 1;
}
```

```
yywrap(
{
    return 1;
}
int lookup (char st[10][10], char *id, int n)
{
    for(j=0; j<n; j++)
    { if(!strcmp (st[j], id))
        return 1;
    return 0;
}
```

Output:

```
//hi world
single line comment
a=3
a is identifier
3 is digit
b==c
b is identifier
== is Relational operator equal to
d=3.25
d is identifier
3.25 is float
@
```