

```

%{

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 |struct |typedef |do |
if |break |continue |void |switch |return |else |
goto {printf("\n %s is a keyword", yytext);}

"/" {COMMENT=1;}{printf("\n %s is a COMMENT LINE",yytext);}

{identifier}\( {if(!COMMENT)printf("\nFUNCTION IS: %s", yytext);}

\{ {if(!COMMENT)printf("\n BLOCK BEGINS");}
\} {if(!COMMENT)printf("BLOCK ENDS ");}

{identifier}(\[[0-9]*\])? {if(!COMMENT) printf("\n %s-> IDENTIFIER", yytext);}

\".*\" {if(!COMMENT)printf("\n %s is a STRING", yytext);}

[0-9]+ {if(!COMMENT) printf("\n %s is a NUMBER ", yytext);}

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

\(\ ECHO;

"=" {if(!COMMENT)printf("\n %s is an ASSIGNMENT OPERATOR",yytext);}
"+" {if(!COMMENT) printf("\n %s is an ADDITION OPERATOR", yytext);}
"-" {if(!COMMENT) printf("\n %s is a SUBTRACTION OPERATOR", yytext);}
"*" {if(!COMMENT) printf("\n %s is a MULTIPLICATION OPERATOR", yytext);}
"/" {if(!COMMENT) printf("\n %s is a DIVISION OPERATOR", yytext);}
%" {if(!COMMENT) printf("\n %s is a MODULUS OPERATOR", yytext);}
\<= |\>= |\< |== |\> {if(!COMMENT) printf("\n %s is a RELATIONAL
OPERATOR",yytext);}

%%

int main(int argc, char **argv)
{

```

```
FILE *file;
file=fopen("Sample_input.c","r");
if(!file)
{
printf("could not open the file");
exit(0);
}
yyin=file;
yylex();
printf("\n");
return(0);
}

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