**NAME : DHANRAJ SUBHASH KORE**

**SUB : COMPILER DESIGN LAB ASSIGNMENT**

**BATCH : B-3 ROLL NO : 60**

**Q. LEX/FLEX specification and programming regular expressions**

**LEX CODE:**

%{

    #include<stdio.h>

%}

alpha [A-Za-z]

id [a-zA-Z0-9\_][a-zA-Z0-9\_]\*

email {id}+(\.{id})\*@{id}+(\.{alpha}+)+

%%

\n {};

\".\*\" {printf("\tLiteral: %s\n",yytext);}

for|if|else|"else if"|printf {printf("\tKeyword: %s\n",yytext);}

int|float|double|char {printf("\tData Type: %s\n",yytext);}

"=="|"="|<|"++"|"%" {printf("\tOperator: %s\n",yytext);}

"{" {printf("\tOpening Block:  %s\n",yytext);}

"}" {printf("\tClosing Block:  %s\n",yytext);}

"("|"["|")"|"]" {printf("\tBrackets :  %s\n",yytext);}

";" {printf("\tSemicolon:  %s\n",yytext);}

"//".\*"\n" {printf("\tSingle line comment : %s\n", yytext);}

\/\\*(.\*\n)\*.\*\\*\/ ; {printf("\tMulti line comment : \n%s\n", yytext);}

{id} {printf("\tIdentifier: %s\n",yytext);}

{email} {printf("\tEmail: %s\n",yytext);}

.|\n;

%%

int main()

{

    yyin = fopen("input.txt","r");

    yylex();

    return 0;

}

int yywrap()

{

    return 1;

}

**OUTPUT:**



