# ASSIGNMENT: 02

Name: Abhijeet Biswas
SRN: 201900400
Roll No: 05
Div: B

## Question:

Design a Lexical analyzer for the subset of C Language using LEX or FLEX. Read input from file. Also Create symbol table. Upload single file with input , output and source code.

## Input (C Program):

```
int main()
{
   int 2Abhijeet;
   printf("Hello Abhijeet");
}
```

## Code:

```
%{
#include <stdio.h>
#include<string.h>
struct symEntry{
int index;
char lexeme[30];
};
struct symEntry symtable[30];
int sti=0;
```

```
int line=1;
void put_symtab(){
int j;
if(sti==0){
symtable[sti].index=sti+1;
strcpy(symtable[sti].lexeme,yytext);
sti++;
return;
}
for(j=0;j<sti;j++){
if(strcmp(symtable[j].lexeme,yytext)==0){
return;
}
}
symtable[sti].index=sti+1;
strcpy(symtable[sti].lexeme,yytext);
sti++;
}
%}
letter [a-zA-Z]
number [0-9]
delim ["|"]
%%
"int"|"if"|"double"|"long"|"goto"|"static"|"float"|"short"|"while"|"char"|"con
st"|"void"|"else"|"return"|"printf"|"scanf" {printf("\n %d\t%s \t\tKeyword",
line, yytext);}
"("|")"|"{"|"}"|"["|"]"|";"|"," {printf("\n %d\t%s \t\tDelimiter", line, yytext);}
```

```
{delim} {printf("\n %d\t%s \t\tDelimiter", line, yytext);}

"+"|"-"|"*"|"%"|"/"|"++"|"--" {printf("\n %d\t%s \t\tArithmetic Operator", line, yytext);}

"=="|"<"|">"|"<="|">=" {printf("\n %d\t%s \t\tRelational Operator", line, yytext);}

"=" {printf("\n %d\t%s \t\tAssignment Operator",line, yytext);}

{letter}+|({letter}{number})* {printf("\n %d\t%s \t\tIdentifier", line, yytext);put_symtab();}

{number}+ {printf("\n %d\t%s\t\tConstant", line, yytext);}

{number}+{letter}+ {printf("\n %d\t%s \tERROR This is ILLEGAL", line, yytext);}

{delim}({letter}|{number})*{delim} {printf("\n %d\t%s \tString Constant/Literal", line, yytext);}

"\n" {line++;}

%%

void print_st(){

int j;

printf("\nSymbol Table : \n");

printf("------------------------\n");

printf("| Line\t|\tLexeme\t|\n");

printf("------------------------\n");

for(j=0;j<sti;j++){

printf("| %d\t|\t%s\t|\n",symtable[j].index,symtable[j].lexeme);

}

printf("------------------------\n");

}

int main(int argc)

{
```

```
yyin = fopen("input.txt", "r");
printf("\nLine | Lexeme | \tToken\n");
yylex();
printf("\n\n");
print_st();
fclose(yyin);
}
int yywrap()
{
return 1;
}
```

**Output:**