Lab4: Implement scanner phase of compiler using flex

**Group Members:**

1. Kathiriya Darshak (IT057)
2. Limbani Nihal (IT064)
3. Karia Stuti (IT055)

**F1) Flex programs as per lab manual.**

* flex is a tool for generating scanners: programs which recognize lexical patterns in text.
* flex reads the given input files (or its standard input if no file names are given) for a description of the scanner to generate.

Generate scanner using FLEX, which can identify following language –“D”, and display appropriate messages on recognizing them.

Rules of language “D” as per manual:

Code:

%{

#include<stdio.h>

int totaltoken= 0;

%}

%%

("if")|("else")|("while")|("do")|("continue")|("untill")|("for")|("case")|("switch") {totaltoken ++; printf("a.This is Keywords: %s\n",yytext);}

[a-zA-Z\_][a-zA-Z0-9\_]\* {totaltoken ++; printf("b.This is identifiers: %s\n",yytext);}

[0-9]+"."[0-9]+ {totaltoken ++; printf("c1.This is Floats : %s\n", yytext);}

[0-9]+ {totaltoken ++; printf("c2.This is Integer: %s\n",yytext);}

"\*"." " {totaltoken ++; printf("d.This is single line Commments start with \*\*\*: %s\n",&yytext[3]);}

[" "\t] {printf("e.This is a white space \n");}

[,|;(|)|{|}|.|\[|\]] {totaltoken ++; printf("f.This is Punctuation: %s\n",yytext);}

"+"|"-"|"\*"|"/"|"%" {totaltoken ++; printf("g.This is (mathametical) Operator: %s\n",yytext);}

"<="|">="|":="|"!="|"=="|"<>" {totaltoken ++; printf("g.This is (relational) Operator: %s\n",yytext);}

'[^']+' {totaltoken ++; printf("f.This is String: %s\n",yytext);}

"$" return 0;

%%

int yywrap(){}

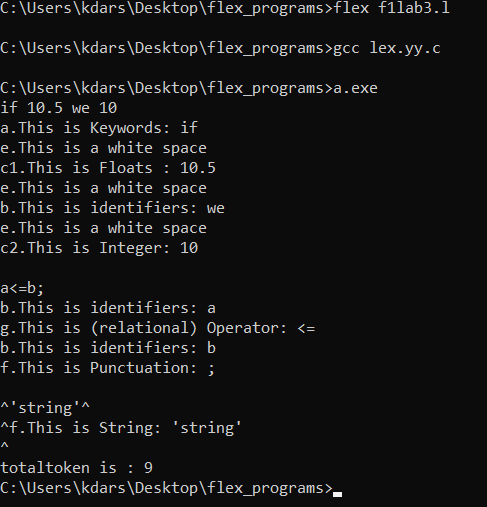
int main(){

yylex();

printf("totaltoken is : %d",totaltoken );

return 0;

}



**Code: (lab perform simple program)**

**%{**

**#include <stdio.h>**

**%}**

**%%**

**[0-9]+ { printf("got number token: %s \n", yytext); }**

**[0-9]+(.[0-9]+) {printf("got float number %s \n",yytext);}**

**\n {printf("got new line \n"); }**

**. {printf("got non numeric charachter %s \n",yytext);}**

**%%**

**int yywrap()**

**{**

**return 1;**

**}**

**int main()**

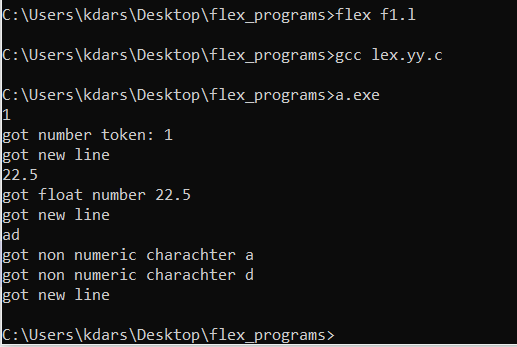
**{**

**yylex();**

**return 0;**

**}**

**Output:**

****