**Lab Report No: 04**

Problem name: Write a flex code to lexical analyze of arithmetic expression. (Arithmetic expression can contain variables, real numbers and arithmetic operators.)

Course name: Compiler Design Lab

Course code: CSE-332

Date of Performance: 25.07.2019

Date of Submission: 01.08.2019

Submitted to:

Mr. Fakhrul Islam

Lecturer

Department of Computer Science and Engineering

Faculty of Science and Information Technology,

Daffodil International University

Submitted by

Student name: Iffat Firozy Rimi

Student id: 163-15-8432

Code:

%{

/\*Definition section \*/

%}

%%

[ ]+ {fprintf(yyout, "");}

[0-9]+ { fprintf(yyout, "Lexeme: %s\tToken: Real Number\n", yytext); }

[+] { fprintf(yyout, "Lexeme: %s\tToken: Addition Operator\n", yytext); }

[-] { fprintf(yyout, "Lexeme: %s\tToken: Subtraction Operator\n", yytext); }

[\*] { fprintf(yyout, "Lexeme: %s\tToken: Multiplication Operator\n", yytext); }

[/] { fprintf(yyout, "Lexeme: %s\tToken: Division Operator\n", yytext); }

[%] { fprintf(yyout, "Lexeme: %s\tToken: Modulus Operator\n", yytext); }

[=] { fprintf(yyout, "Lexeme: %s\tToken: Assignment Operator\n", yytext); }

[a-zA-Z][a-zA-Z1-9]\* { fprintf(yyout, "Lexeme: %s\tToken: Variable\n", yytext); }

%%

int yywrap(){}

int main()

{

extern FILE \*yyin, \*yyout;

yyin = fopen("Input.txt", "r");

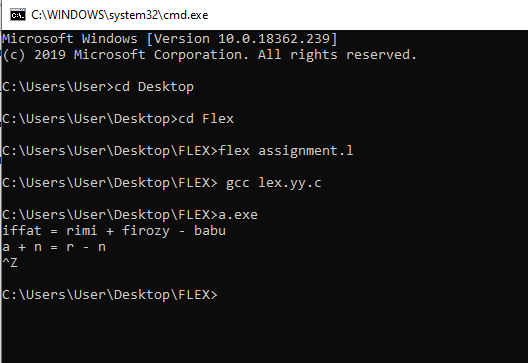
yyout = fopen("Output.txt", "w");

yylex();

return 0;

}

INPUT:



Output:

