## **Lab Report - 5**

**Submitted by**

**Name: Iffat Firozy Rimi**

**Id: 163-15-8432**

**Problem Name: Write Lex and Yacc program to make a calculator which can execute an arithmetic expression containing addition, subtraction, multiplication, division, power(^) and parenthesis. The operands could be floating point number.**

**Code:**

**.l file:**

**%{**

**#include "y.tab.h"**

**extern int yylval;**

**%}**

**%option noyywrap**

**%%**

**[0-9]+ { yylval = atoi (yytext);**

**printf ("In lex: scanned the number %d\n", yylval);**

**return NUMBER; }**

**[ \t] { printf ("In lex: skipped whitespace\n"); }**

**\n { printf ("In lex: reached end of line\n");**

**return 0;**

**}**

**. { printf ("In lex: found other data \"%s\"\n", yytext);**

**return yytext[0];**

**/\* so yacc can see things like '+', '-', and '=' \*/**

**}**

**%%**

**.Y file:**

**%{**

**#include <stdio.h>**

**%}**

**%token NUMBER**

**%%**

**statement: expression { printf("In yacc: = %d\n", $1); } ;**

**expression: expression '+' term0 { $$ = $1 + $3;**

**printf ("In yacc: Recognized '+' expression.\n");**

**}**

**| expression '-' term0 { $$ = $1 - $3;**

**printf ("In yacc: Recognized '-' expression.\n");**

**}**

**| term0 { $$ = $1;**

**printf ("In yacc: Recognized a number.\n");**

**}**

**term0: term0 '\*' NUMBER { $$ = $1 \* $3;**

**printf ("In yacc: Recognized '\*' expression.\n");**

**}**

**| term0 '/' NUMBER { $$ = $1 / $3;**

**printf ("In yacc: Recognized '/' expression.\n");**

**}**

**| NUMBER { $$ = $1;**

**}**

**;**

**%%**

**int main (void) {**

**return yyparse();**

**}**

**/\* Added because panther doesn't have liby.a installed. \*/**

**int yyerror (char \*msg) {**

**return fprintf (stderr, "YACC: %s\n", msg);**

**}**

**Output:**

