ASSIGNMENT: 04

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

Question:

Design a YACC and corresponding LEX specification to compute the value of an expression. Consider arithmetic, trigonometric , 1/x , sqrt(x) , x ^y etc. operators.

Input:

1. 5+30*2

2. -6-10

3. cos(90)+3

Code:

ass4.y –

```
%{
	#include <stdio.h>
	#include <math.h>
%}

%union	//to define possible symbol types
{double p;}
%token<p>num
%token SIN COS TAN LOG SQRT

%left '+"-'	//lowest precedence
%left '*"/'	//highest precedence
```

```
%nonassoc uminu  //no associativity
%type<p>exp        //Sets the type for non-terminal
%%

/*for storing the answer */
ss: exp {printf(" Answer = %g\n",$1);}

/* for binary arithmatic operators */
exp:   exp'+'exp   { $$=$1+$3; }
     |exp'-'exp  { $$=$1-$3; }
     |exp'*'exp  { $$=$1*$3; }
     |exp'/'exp  {
                   if($3==0)
                   {
                      printf("Divide by Zero");
                   }
                   else $$=$1/$3;
                 }
     |'-'exp        {$$=-$2;}
     |SIN'('exp')'   {$$=sin($3);}
     |COS'('exp')'   {$$=cos($3);}
     |TAN'('exp')'   {$$=tan($3);}
     |LOG'('exp')'   {$$ =log($3);}
     |SQRT'('exp')'  {$$ =sqrt($3);}
     |num;
     |'('exp')'      {$$=$2;}
```

```
%%

int main()
{
        do
        {
   printf("\nExpression:");
                yyparse();     /* Parse the sentence repeatedly until the i/p runs out
*/
        }while(1);
}

yyerror(char *s;)   /* to print error message when an error is parsing of i/p */
{
  printf("Syntax Error");
}
```

ass4.l –

```
%{
  #include <math.h>
  #include "y.tab.h"

%}

%%
[0-9]+|[0-9]*\.[0-9]+ {
                yylval.p = atof(yytext);
                return num;
```

}

sin {return SIN;}

cos {return COS;}

tan {return TAN;}

log {return LOG;}

sqrt {return SQRT;}

[\t] ;

\n return 0;

. return yytext[0];

%%

Output: