**Lex File:**

%option noyywrap

%{

#include<stdio.h>

#include "y.tab.h"

extern int yylval;

%}

%%

[0-9]+ {

yylval=atoi(yytext);

return DIGIT;

}

[\t] ;

\n|. return yytext[0];

%%

**Yacc File:**

%{

#include<stdio.h>

#include "lex.yy.c"

void yyerror(const char \*c);

%}

%token DIGIT

%left '+' '-'

%left '\*' '/' '%'

%left '(' ')'

%%

line:line E '\n' {printf("%d\n",$2);}

|line '\n'

|/\*empty\*/

|error '\n' {yyerror("Re-enter Previous Line"); yyerrok;}

;

E:E '+' E {$$=$1+$3;}

|

E '-' E {$$=$1-$3;}

|

E '\*' E {$$=$1\*$3;}

|

E '/' E {$$=$1/$3;}

|

'(' E ')' {$$=$2;}

| DIGIT {$$=$1;}

;

%%

void main()

{

printf("\n Enter any Airthmetic Expression which can have operations Addition,Subtraction,Multiplication,Division,Modulous and round Brackets;\n");

yyparse();

printf("\nEntered Arithmetic Expression is valid\n");

}

void yyerror(const char \*str)

{

printf("Invalid Expression: %s\n\n",str);

}

**OUTPUT:**

