EXPERIMENT NO. 10

AIM :- To write a YACC program for desk calculator.

ALGORITHM:-

1. Define following YACC specifications along with the required actions

Line Expr ‘\n’;

Expr Expr’+’ Term;

Term Term’\*’ Factor

|Factor ;

Factor ‘(‘ expr ‘)’

|DIGIT;

1. Write a function yylex for lexical analysis of the input. The function generates following tokens (a) DIGIT (b) + (c) \* (d) ( (e) )

Compile the program and execute.

CODE:-

LEX PROGRAM:-

%{

   /\* Definition section \*/

  #include<stdio.h>

  #include "y.tab.h"

  extern int yylval;

%}

/\* Rule Section \*/

%%

[0-9]+ {

          yylval=atoi(yytext);

          return NUMBER;

       }

[\t] ;

[\n] return 0;

. return yytext[0];

%%

int yywrap()

{

 return 1;

}

YACC PROGRAM:-

%{

   /\* Definition section \*/

  #include<stdio.h>

  int flag=0;

%}

%token NUMBER

%left '+' '-'

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

%left '(' ')'

/\* Rule Section \*/

%%

ArithmeticExpression: E{

         printf("\nResult=%d\n", $$);

         return 0;

        };

 E:E'+'E {$$=$1+$3;}

 |E'-'E {$$=$1-$3;}

 |E'\*'E {$$=$1\*$3;}

 |E'/'E {$$=$1/$3;}

 |E'%'E {$$=$1%$3;}

 |'('E')' {$$=$2;}

 | NUMBER {$$=$1;}

 ;

%%

//driver code

void main()

{

   printf("\nEnter Any Arithmetic Expression which

                   can have operations Addition,

                   Subtraction, Multiplication, Divison,

                          Modulus and Round brackets:\n");

   yyparse();

   if(flag==0)

   printf("\nEntered arithmetic expression is Valid\n\n");

}

void yyerror()

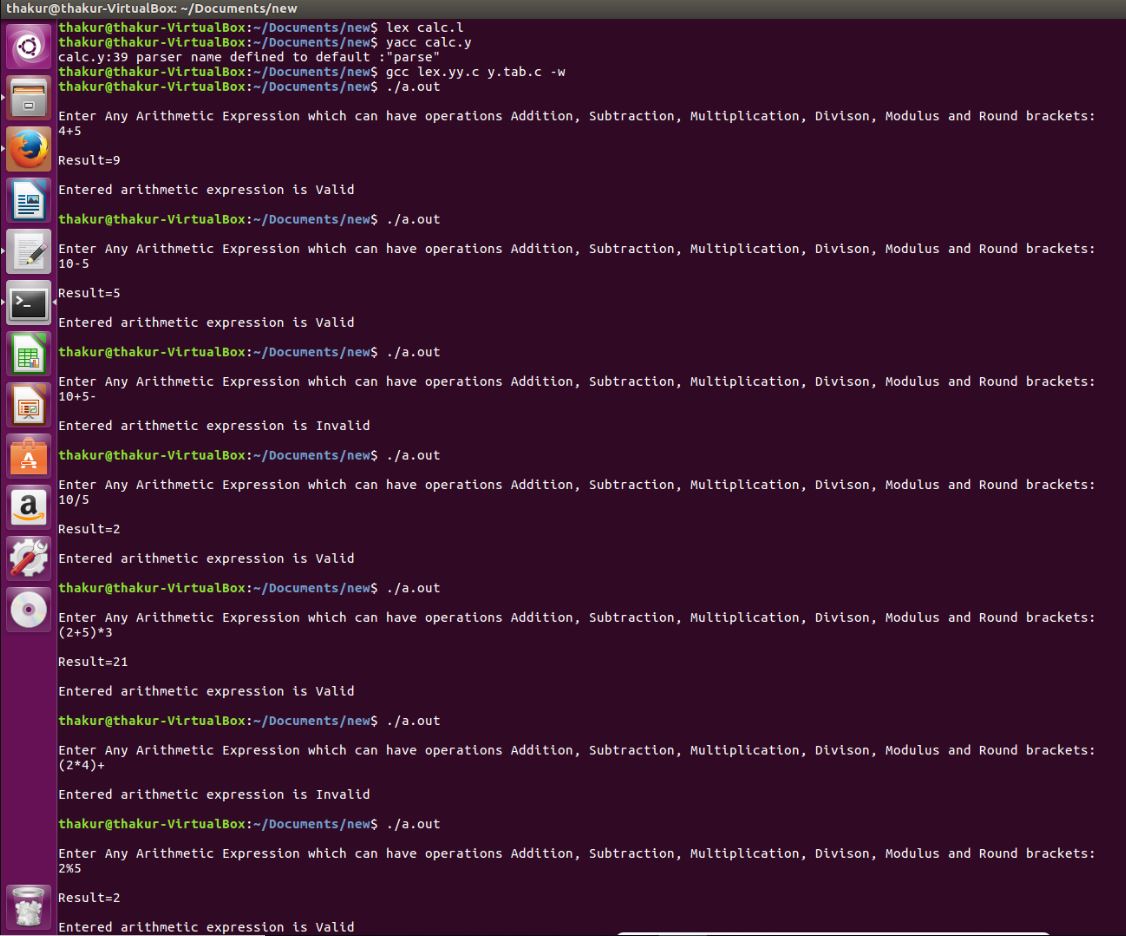
{

   printf("\nEntered arithmetic expression is Invalid\n\n");

   flag=1;

}

OUTPUT:-



RESULT:- Thus,YACC program for desk calculator has been successfully executed.