# Program:

## Infix.l

%{

#include "y.tab.h"

%}

%%

[0-9]+ { yylval.dval=atoi(yytext); return NUMBER;}

[0-9]\*"."[0-9]+ { yylval.dval=atof(yytext); return NUMBER;} [a-zA-Z] { return LETTER; }

"+" { return PLUS;}

"-" { return MINUS;} "\*" { return MULTIPLY;} "/" { return DIVIDE;}

"(" { return OPEN;}

")" { return CLOSE;} "\n" { return ENTER;} "$" { return 0;}

%%

## Infix.y

%{

#include<stdio.h> #include<math.h>

%}

%union { double dval; char symbol;

}

%token<dval>NUMBER

%token<symbol>LETTER

%token PLUS MINUS MULTIPLY DIVIDE OPEN CLOSE ENTER

%left PLUS MINUS

%left DIVIDE MULTIPLY

%nonassoc UMINUS

%type<dval>E

%%

print: E ENTER { printf("\n\v VALID INFIX EXP \n"); exit (0); }

;

E:E PLUS E

|

E MINUS E

|

E MULTIPLY E

|

E DIVIDE E

|

MINUS E %prec UMINUS {$$=-$2;}

|

OPEN E CLOSE { $$=$2;}

|

NUMBER { $$=$1; }

|

LETTER {$$=$1;}

;

%%

int main()

{

printf("\n Enter infix expression: "); yyparse();

return 0;

}

void yyerror( char \*msg)

{

printf("\n INVALID INFIX EXPRESSION ");

}

int yywrap(){return(1);}

## Output:

Ubuntu@Ubuntu-ThinkCentre-M72e:~$ lex infix.l Ubuntu@Ubuntu-ThinkCentre-M72e:~$ yacc -d infix.y Ubuntu@Ubuntu-ThinkCentre-M72e:~$ gcc lex.yy.c y.tab.c -w Ubuntu@Ubuntu-ThinkCentre-M72e:~$ ./a.out

Enter infix expression: a+b

VALID INFIX EXP.......

Ubuntu@Ubuntu-ThinkCentre-M72e:~$ ./a.out

Enter infix expression: (+a+b)

INVALID INFIX EXPRESSION.....: