**calc.l**

%{

#include "y.tab.h"

%}

%%

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

[ \t] ; // Ignore spaces and tabs

\n return '\n';

. return yytext[0];

%%

***calc.y***

%{

#include <stdio.h>

#include <stdlib.h>

int yylex(void);

void yyerror(const char \*s);

%}

%token NUMBER

%%

input:

/\* empty \*/

| input line

;

line:

'\n'

| expr '\n' { printf("= %d\n", $1); }

;

expr:

expr '+' term { $$ = $1 + $3; }

| expr '-' term { $$ = $1 - $3; }

| term { $$ = $1; }

;

term:

term '\*' factor { $$ = $1 \* $3; }

| term '/' factor {

if ($3 == 0) {

yyerror("Division by zero");

exit(1);

}

$$ = $1 / $3;

}

| factor { $$ = $1; }

;

factor:

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

| NUMBER { $$ = $1; }

;

%%

void yyerror(const char \*s) {

fprintf(stderr, "Error: %s\n", s);

}

**Run the program**

**Step 1: Save the files**

* Save LEX file as calc.l
* Save YACC file as calc.y

**Step 2: Compile using flex and bison**

bash

CopyEdit

bison -d calc.y # Generates y.tab.c and y.tab.h

flex calc.l # Generates lex.yy.c

gcc lex.yy.c y.tab.c -o calc -lm

**Step 3: Run the calculator**

./calc

**Step 4: Try Input**

3 + 4 \* 2

(1 + 2) \* 5

10 / 2

Output:

= 11

= 15

= 5