**[[1]](#footnote-0)PRACTICAL 7**

.

**AIM** - Postfix Expression Evaluation

**CODE**-

**Practical7\_1099.l**

%{

#include "y.tab.h"

#include <stdio.h>

#include <stdlib.h>

// Declare yyerror to avoid implicit declaration warning

void yyerror(const char \*s);

%}

%%

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

"+" { return '+'; }

"-" { return '-'; }

"\*" { return '\*'; }

"/" { return '/'; }

"in" { return IN; }

// skip whitespace

[ \t\n]+ { /\* ignore \*/ }

. { yyerror("Invalid character"); return 0; }

%%

**Practical7\_1099.y**

%{

#include <stdio.h>

#include <stdlib.h>

int yylex(void);

void yyerror(const char \*s);

%}

%token NUMBER

%%

input:

/\* empty \*/

| input expr '\n' { printf("Result = %d\n", $2); }

;

expr:

expr expr '+' { $$ = $1 + $2; }

| expr expr '\*' { $$ = $1 \* $2; }

| expr expr '-' { $$ = $1 - $2; }

| expr expr '/' {

if ($2 == 0) {

yyerror("Division by zero");

YYABORT;

}

$$ = $1 / $2;

}

| NUMBER { $$ = $1; }

;

%%

void yyerror(const char \*s) {

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

}

int main() {

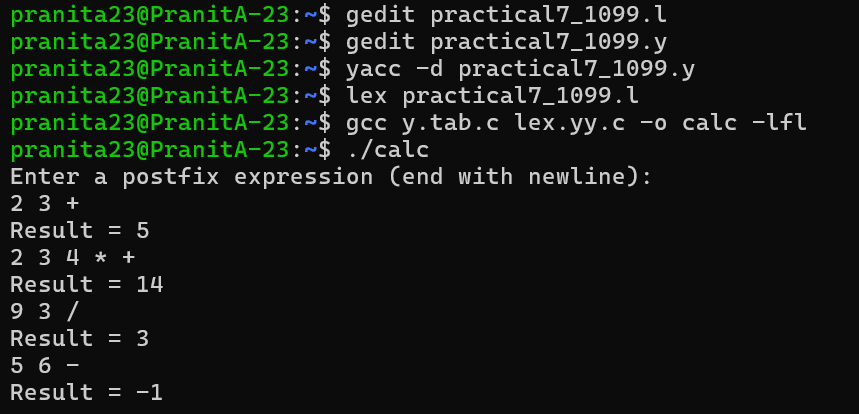
printf("Enter a postfix expression (end with newline):\n");

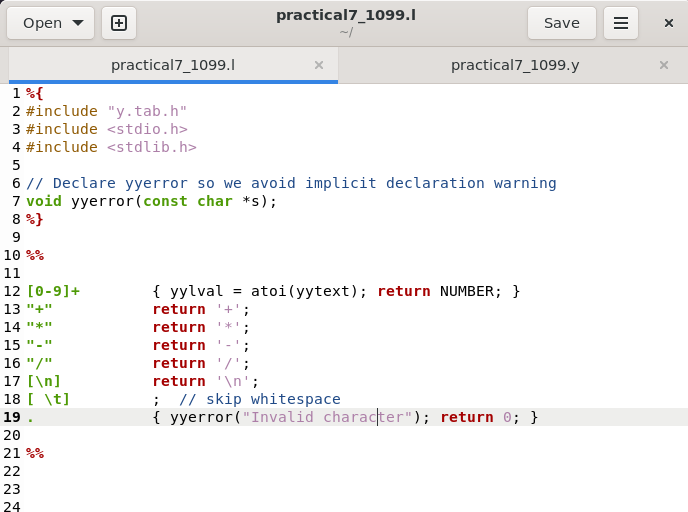
yyparse();

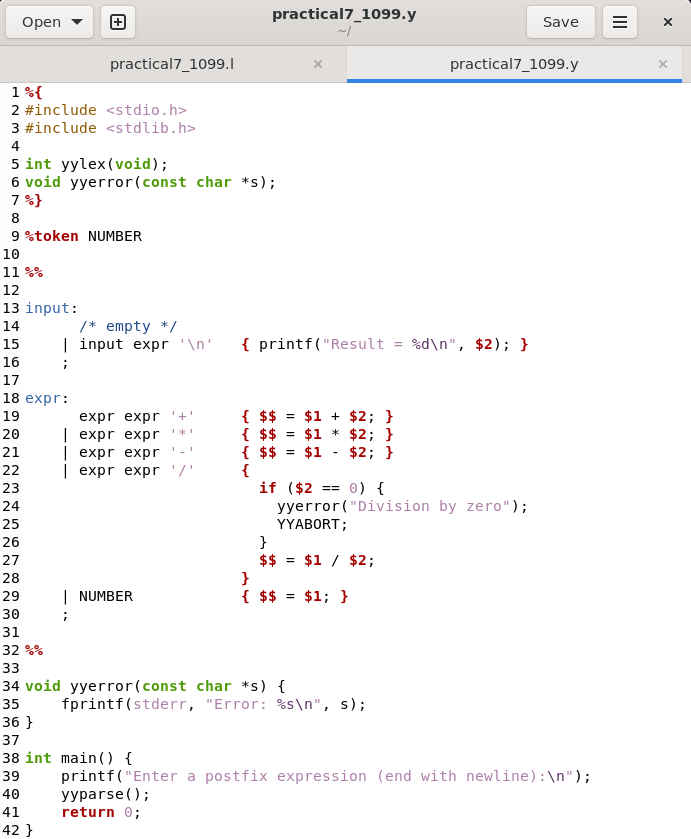
return 0;

}

**ScreenShot**







1. **Name- Pranita Dadhe ; PRN-22070521099**  [↑](#footnote-ref-0)