**CD LAB ASSIGNMENT**

*Submitted by:*

*M. Lakshmi Abhigna*

*CSE-F*

*AP21110010343*

Aim: YACC Specification for a simple desk calculator

***Lex code:***

%{

#include "calculator.tab.h"

%}

%%

[ \t] ; // Skip whitespace

"+" { return PLUS; }

"-" { return MINUS; }

"\*" { return MULT; }

"/" { return DIV; }

"%" { return MOD; }

"(" { return LPAREN; }

")" { return RPAREN; }

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

. { return yytext[0]; } // Any other character is returned as is

%%

int yywrap() {

return 1; // Indicate end of input

}

***YACC Code:***

%{

#include <stdio.h>

#include <stdlib.h>

%}

%token NUM

%token PLUS MINUS MULT DIV MOD LPAREN RPAREN

%%

calculator: expr { printf("Result=%d\nEntered arithmetic expression is Valid\n", $1); }

;

expr: term { $$ = $1; }

| expr PLUS term { $$ = $1 + $3; }

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

;

term: factor { $$ = $1; }

| term MULT factor { $$ = $1 \* $3; }

| term DIV factor { $$ = $1 / $3; }

| term MOD factor { $$ = $1 % $3; }

;

factor: NUM { $$ = $1; }

| LPAREN expr RPAREN { $$ = $2; }

;

%%

int yylex(void) {

int c = getchar();

if (c == '+' || c == '-' || c == '\*' || c == '/' || c == '%' || c == '(' || c == ')') {

return c;

} else if (c >= '0' && c <= '9') {

ungetc(c, stdin);

int value;

scanf("%d", &value);

yylval = value;

return NUM;

} else if (c == '\n' || c == EOF) {

return 0; // End of input

} else {

return -1; // Invalid input

}

}

int main(void) {

yyparse();

return 0;

}

void yyerror(const char \*s) {

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

exit(EXIT\_FAILURE);

}