**Practical - 9**

**AIM:** Write a program to validate the IF…. ELSE statement using Lex and YACC.

**YACC CODE**

%{

#include <stdio.h>

#include <stdlib.h>

int yylex(void);

void yyerror(const char \*s);

%}

%token ID NUM IF OPBR CLBR LE GE EQ NE OR AND ELSE

%right '='

%left AND OR

%left '<' '>' LE GE EQ NE

%left '+' '-'

%left '\*' '/'

%right UMINUS

%left '!'

%%

S : ST { printf("Input accepted.\n"); exit(0); }

;

ST : IF '(' E2 ')' OPBR ST1 ';' CLBR ELSE ST1 ';'

| IF '(' E2 ')' OPBR ST1 ';' CLBR

;

ST1 : ST

| E

;

E : ID '=' E

| E '+' E

| E '-' E

| E '\*' E

| E '/' E

| E '<' E

| E '>' E

| E LE E

| E GE E

| E EQ E

| E NE E

| E OR E

| E AND E

| ID

| NUM

;

E2 : E '<' E

| E '>' E

| E LE E

| E GE E

| E EQ E

| E NE E

| E OR E

| E AND E

| ID

| NUM

;

%%

void yyerror(const char \*str) {

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

}

int yywrap(void) {

return 1;

}

int main(void) {

printf("Enter the expression: ");

yyparse();

return 0;

}

**LEX CODE**

%{

#include <stdio.h>

#include "y.tab.h"

%}

alpha [A-Za-z]

digit [0-9]

%%

[ \t\n] ; /\* Skip whitespace \*/

"if" { return IF; }

"else" { return ELSE; }

"{" { return OPBR; }

"}" { return CLBR; }

{digit}+ { return NUM; }

{alpha}({alpha}|{digit})\* { return ID; }

"<=" { return LE; }

">=" { return GE; }

"==" { return EQ; }

"!=" { return NE; }

"||" { return OR; }

"&&" { return AND; }

. { return yytext[0]; }

%%

