	Practical No. 9
*	Title: Parser for "For" Loop statements.
	Out of the state o
*	Objective: Students will learn and implement
_	) No and Comment of the state o
-	i) Parser for "For" Loop statements
Н	ii) Rules sections for such a seamer and a parson and their working in synthronization.
	and their morking in synthronization.
	Dericitet!
*	Description:
	Brate I EV I'M I'm I DE I
	the dit has the first & Breste a RE for
	i) Greate a LEX file first & Breate a RE for the digit and for the alphabet.
	ii) To the and and and a to be less in
	identilization is the first in the trule
	by a distinguished about in the state of the
	feyword agranous offerences of the
	we see man equal to great than
	equal no paper to the data to the Think
	in the rules section, write orules for number identification, identifies identification. FOR pegusord, conditional operations operators like less than equal to preates than equal to, rot equal to, OR, AND, other details aparts from this
	iii) The weste a YACC dide injustrator In the
	dessiration part of the YACC file include.  header file sine etdio h, etdlit h.
	harding fact stains the stalit he
	reader gre sure from the
	De Las TD. Num, LAS, LE, GE, ER NE.
7	Destore ID, Num, for, LE, CrE, EQ, NE, OR, AND os topen. Destore precedence for considered topens.
	la mideral tokens.
	you would be to
	A NUMBER OF STREET
+	IT the all lasting I have a
+	las street section of A (Cfile and rules
+	for run sympol, definition, body, F and
+	El wind are in a way party for FOR"
+	for start symbol, definition, lody, F and E2 which are in a way party for FOR"
+	·
+	VI) (han include them less gy a file which
+	all as a sometor between IFX and
-	YHIC files. Eastly, wall main routines
3	JAC files. Eastly, wall main moratines for cherking the FOR of enfercion through yyporse ().
	through elyporse ().

## **Code for LEX:**

```
alpha [A-Za-z]
digit [0-9]
%%
[\t \n]
for
                              return FOR;
{digit}+
                              return NUM;
                              return ID;
{alpha}({alpha}|{digit})*
"<="
                              return LE;
">="
                              return GE;
"=="
                              return EQ;
" ! = "
                              return NE;
"||"
                              return OR;
"&&"
                              return AND;
                              return yytext[0];
%%
```

## **Code for YACC:**

```
%{
#include <stdio.h>
#include <stdlib.h>

void yyerror(char*);
int yylex(void);

%}

%token ID NUM FOR LE GE EQ NE OR AND
%right "="
%left OR AND
%left '>' '<' LE GE EQ NE
%left '+' '-'
%left '*' '/'
%right UMINUS
%left '!'</pre>
```

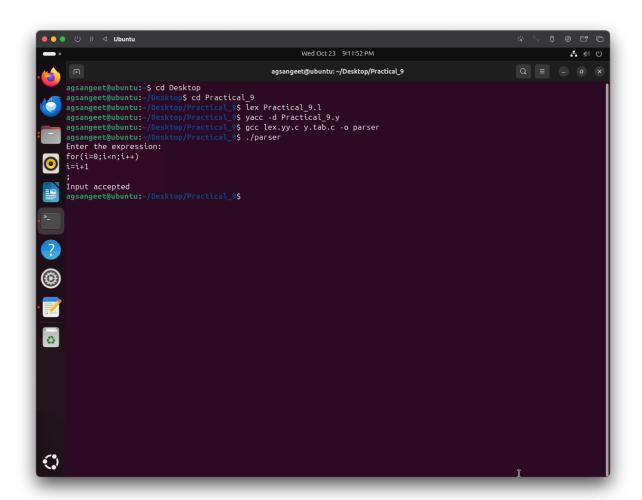
```
S
        : ST {printf("Input accepted\n"); exit(0);}
        : FOR '(' E ';' E2 ';' E ')' DEF
ST
   : '{' BODY '}'
DEF
          | E';'
          ST
BODY : BODY BODY
          | E ';'
          | ST
          ;
Е
        : ID '=' E
         | E '+' E
         | E '-' E
         | E '*' E
         | E '/' E
         | E '<' E
         | E '>' E
         | E LE E
         | E GE E
         | E EQ E
         | E NE E
         | E OR E
         | E AND E
         E '+' '+'
         | 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
;

#include "lex.yy.c"
int main() {
    printf("Enter the expression:\n");
    yyparse();
    return 0;
}

void yyerror(char* errorText){
    printf("%s",errorText);
}
```

## **Output:**



*	Condus	ion;	Thus, we have	implemented the
	barres	Sas	SFOR" loop.	
		1	30 37	