	Partial W/ O
	Practical No. 8:
X	Title: lesk salulator and related parar
	resolution.
	, , , , , , , , , , , , , , , , , , , ,
1	Myeuwe: Students will learn and implement
	i) Parses for desk calculator and soluted
4	Estas selavery
	ii) Rules resting for mil a server
	leaves and the sum a granner and a
	Objective: Students will learn and implement i) Parses for desk inhulator and related ii) Pulls section for such a scanner and a parser and their working in synchronization.
X	lescription:
1	
	i late - 154 11 1:0 1 1 DE la -M 11 14
/	i) Create a LEX file first & RE for the signit
	ii) In the rules section, write rules for number identification, rewline and other details apport from this.
1	ide tilis tim sousing and other details whent
	Manyagnor, new une and over ocialis apox
- 1-	from Thy.
	ivi) Then weste a YACC file. In declaration part
	of the YACC lile include header lile like sthis to
	etupe h my deline. YYSTYPF as double.
	in) Then seeste a XACC file. In declaration port of the XACC file include header file like stdio.h, etype.h and define. YYSTYPE as double.
	in the state of the section of the
_	iv) Persone ID as a number, and precedence for taken related arithmetic operations like +, -, * and fete:
_	token related arithmetic operations like
	+, -, * and lett.
	V T. the sules section of YACC sile, add rules
	Jos the rules section of YMC file, add rules for start symbol which will print the value of the answer, write rules for catching the error like if the expression is entered.
	for Mari significant and for the fore it things
	of the answer, will must got among
-	the estor like if the empression is emercan
	is begand they are to at it
	is wrong then ask user to enter it once more.
	· A. de · A
	VI) this write ruly for different arithmetic
	operations like +, + and /, unary - and for number also in the orules fection of the YAC file.
	and for number also in the auto bestion
	of the YALL is
	of the .
	The said of the said of
	VII) Then instructe Sea. 44. I file which arts as a sometor let w LEX and YALL files. Lastly, sall main proutine for printing the value of the enpression through ypparse ().
	a nonnector Net'w LEX and YALL files.
	fastly, call main groutine for printing
	the value of the entresion therush
	in makes 11
	y yy pa ch ().
100	t

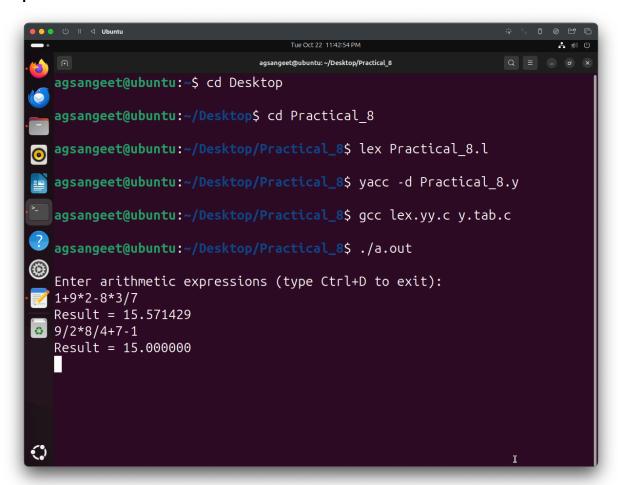
Code for LEX:

```
%{
#include <stdio.h>
#include <stdlib.h>
#include "y.tab.h" // Include generated header from yacc
%}
%%
 [0-9]+(\.[0-9]+)?
                           { yylval.dval = atof(yytext); return NUMBER; }
 [ \t]+
                           ; // Ignore whitespace
                           { return '\n'; } // Return newline as a token
 \n
 "("
                           { return LPAREN; }
 ")"
                           { return RAREN; } // Ensure this matches your
token name
 "+"
                           { return PLUS; }
 0 \subseteq 0
                           { return MINUS; }
 "*"
                           { return MULTIPLY; }
 "/"
                           { return DIVIDE; }
                           { return yytext[0]; } // Any other character
%%
 int yywrap() {
    return 1;
 }
Code for YACC:
%{
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
void yyerror(const char *s); // Declare the error handling function
int yylex(void); // Declare lexer function
%}
%union {
    double dval; // Store double values from expressions
```

```
}
%token <dval> NUMBER
%token LPAREN RAREN PLUS MINUS MULTIPLY DIVIDE
%left PLUS MINUS
%left MULTIPLY DIVIDE
%nonassoc UMINUS // Unary minus precedence
%nonassoc ERROR // Error recovery
%type <dval> expression
%%
// Grammar rules
calculation:
   /* empty */
    calculation expression '\n' { printf("Result = %1f\n", $2); }
expression:
    NUMBER { $$ = $1; }
    expression PLUS expression { $$ = $1 + $3; }
    expression MINUS expression { $$ = $1 - $3; }
    expression MULTIPLY expression { $$ = $1 * $3; }
    expression DIVIDE expression {
        if ($3 == 0) {
            yyerror("Error: Divide by zero");
            YYERROR; // Trigger a syntax error
        } else {
            $$ = $1 / $3;
        }
    | LPAREN expression RAREN { $$ = $2; }
    | MINUS expression %prec UMINUS { $$ = -$2; }
%%
// Error handling function
void yyerror(const char *s) {
    fprintf(stderr, "Syntax error: %s\n", s);
```

```
int main(void) {
    printf("Enter arithmetic expressions (type Ctrl+D to exit):\n");
    return yyparse(); // Start the parser
}
```

Output:



1	Condusion:	Thus,	he have	implemented the
	parser dos	desk	Salukatos	implemented the