	Parti 1 W =
	Practical No. 7
4	T.d P. 11. F. /
1	Title: Portfin Enpression Emplustion
1	Objective: Students will learn and implement
	20.10
	Parser for englisting portion empression
	- full sections for such a starm scarner
	and a parser and their working in
	- Parser for evaluating portion empression - Fules sections for such a state scanner and a parser and their working in synchronization.
*	Para intia.
1	Description:
	i) la to a IFX dila big + l part & Part
	i) Create a LEX file first & Besto fin Regular Empression for the digit.
	Emplession got we argu.
	ii) In the rules section, write rules for digit identification and identification of different operators.
	identification and identification of different
	chee at as a
	operatos.
	in Also in other sevier sertion of IEX life writer
	iii) Also, in the rules section of LEX file, write rules for rewline and for consideration of other details for apart from identification of digits and other operators.
	others dotails by what know identification
	of digith and withe oberatory
	of pugues over the special second
	iv) Then, create a XACC file.
	v) In the Declaration part of YAC (file add rules for start symbol which will print the value of the partin supression at the stack top, write rules for different.
	see start sumfol which will print the
	value of the partlin empression at the
	starle top wante gules for different.
	the state of authorise
	multiplication, division, etc.
	17:61: 1: 11:101.
	muripularian, dirugion, ele.
	The rule ferren of the THE Me milute
	1 FV 1 119 CC 1: Intermediary vel
	vi) After rules section of the YACC file include len yy c file as intermediary bet " LEX and YACC files.
	Vii) Then write the functions PUSH, POP & Topete.
	VIII) Then sall main antino at the last him
	viii) Then wall main noutine at the last by including suparu (1 fun' to ashieve the required parsing functionality.
	manifold has in during lid.
	to the total housements.

## Code for LEX:

```
%{
#include "y.tab.h"
%}
%%
[0-9]+ { yylval = atoi(yytext); return ID; }
[\t\n ] { /* Ignore whitespace */ }
       { return yytext[0]; }
%%
int yywrap() {
    return 1;
}
Code for YACC:
%{
    #include<stdio.h>
    #include<assert.h>
    int yylex(void); // Declaration of yylex function
    void yyerror(const char *s); // Declaration of yyerror function
    void push(int val);
    int pop();
    int top();
%}
%token ID
%%
S
      : E {printf("= %d\n",top());}
     : E E '+' {push(pop() + pop());}
Ε
     | E E '-' {int temp = pop(); push(pop() - temp);}
     | E E '*' {push(pop() * pop());}
```

```
| E E '/' {int temp = pop(); push(pop() / temp);}
     | ID {push(yylval);}
%%
#include "lex.yy.c"
int st[100];
int i = 0;
void push(int val)
{
    assert(i < 100);
    st[i++] = val;
}
int pop()
{
    assert(i > 0);
    return st[--i];
}
int top()
{
    assert(i > 0);
    return st[i-1];
}
void yyerror(const char *s) {
    fprintf(stderr, "Error: %s\n", s);
}
int main()
{
    yyparse();
    return 0;
}
```

## Output:

x	Condusion: Thus, we have simplemented parses
	Conducion: Thus, we have implemented parger for evaluating porting appression.