

***/\* C1: Write a program to design LALR parsing using YACC \*/***

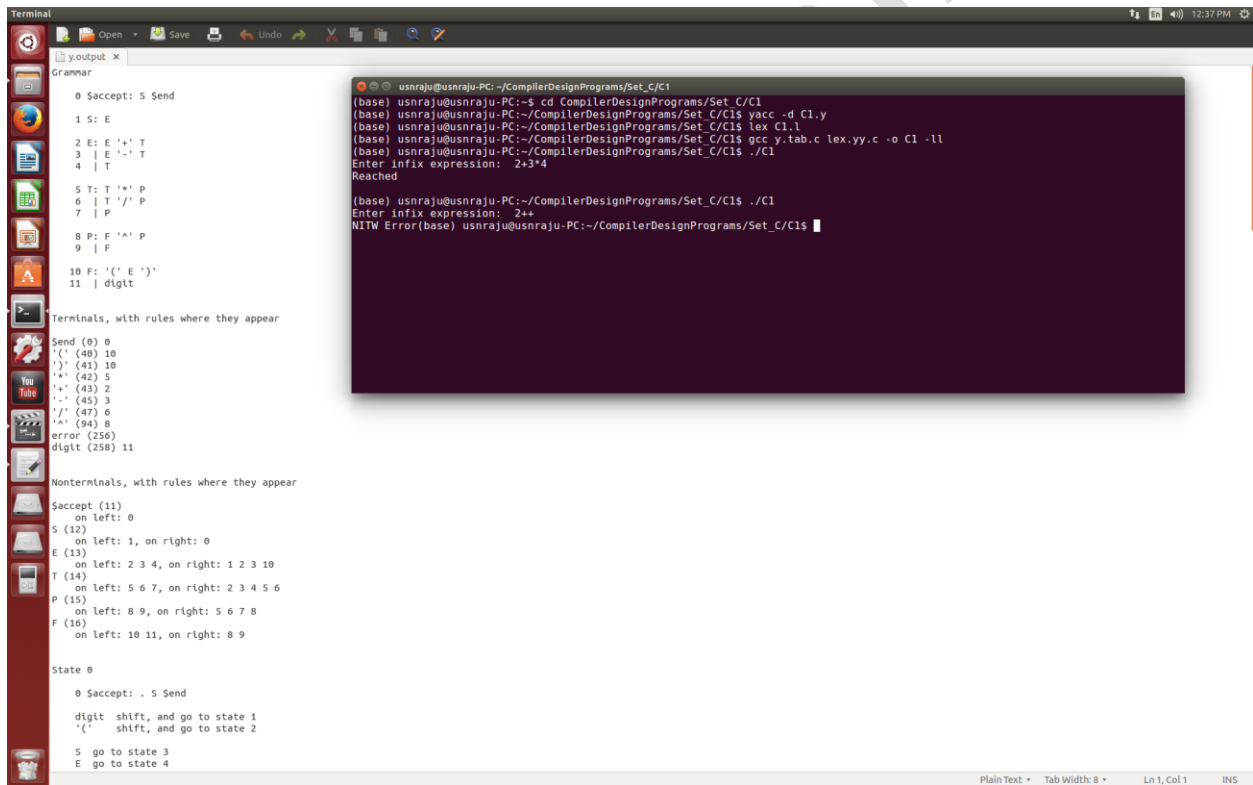
***File: C1.y***

```
%{
    /* Definition section */
    #include <ctype.h>
    #include<stdio.h>
    #include<stdlib.h>
}%
%token digit
/* Rule Section */
%%
/*After Evaluating the expression E,S prints Reached*/
S: E {printf("Reached\n\n");}
;
/*The expression parser uses three different symbols T,F and P, to
set the
precedence and associativity of operators*/
E: E '+' T
| E '-' T
| T
;
T: T '*' P
| T '/' P
| P
;
P: F '^' P
| F
;
F: '(' E ')'
| digit
;
%%

//driver code
int main()
{
    printf("Enter infix expression: ");
    yyparse();
}
yyerror()
{
    printf("NITW Error");
}
```

## File: C1.l

```
%{
#include "y.tab.h"
extern int yylval;
}%
%%
//If the token is an Integer number,then return it's value.
[0-9]+ {yylval=atoi(yytext); return digit;}
//If the token is space or tab,then just ignore it.
[\t] ;
//If the token is new line,return 0.
[\n] return 0;
//For any other token, return the first character read since the last
match.
. return yytext[0];
%%
```



```
Terminal
y.output x
Grammar
0 $accept: S $end
1 S: E
2 E: E '*' T
3 | E '-' T
4 | T
5 T: T '*' P
6 | T '/' P
7 | P
8 P: F '^' P
9 | F
10 F: '(' E ')'
11 | digit

Terminals, with rules where they appear
Send (0) 0
'(' (40) 10
')' (41) 10
'*' (42) 5
'-' (43) 2
'/' (44) 3
'^' (45) 6
'(' (46) 8
error (256) 11
digit (256) 11

Nonterminals, with rules where they appear
$accept (11)
on left: 0
S (12)
on left: 1, on right: 0
E (13)
on left: 2 3 4, on right: 1 2 3 10
T (14)
on left: 5 6 7, on right: 2 3 4 5 6
P (15)
on left: 8 9, on right: 5 6 7 8
F (16)
on left: 10 11, on right: 8 9

State 0
0 $accept: . S $end
digit shift, and go to state 1
'(' shift, and go to state 2
S go to state 3
E go to state 4

(base) usnraju@usnraju-PC: ~/CompilerDesignPrograms/Set_C/C1
(base) usnraju@usnraju-PC:~/CompilerDesignPrograms/Set_C/C1$ yacc -d C1.y
(base) usnraju@usnraju-PC:~/CompilerDesignPrograms/Set_C/C1$ lex C1.l
(base) usnraju@usnraju-PC:~/CompilerDesignPrograms/Set_C/C1$ gcc y.tab.c lex.yy.c -o C1 -ll
Enter infix expression: 2+3*4
Reached
(base) usnraju@usnraju-PC:~/CompilerDesignPrograms/Set_C/C1$ ./C1
Enter infix expression: 2+
NITW Error(base) usnraju@usnraju-PC:~/CompilerDesignPrograms/Set_C/C1$
```