/* C4: Use YACC to convert: Infix expression to Postfix expression. */

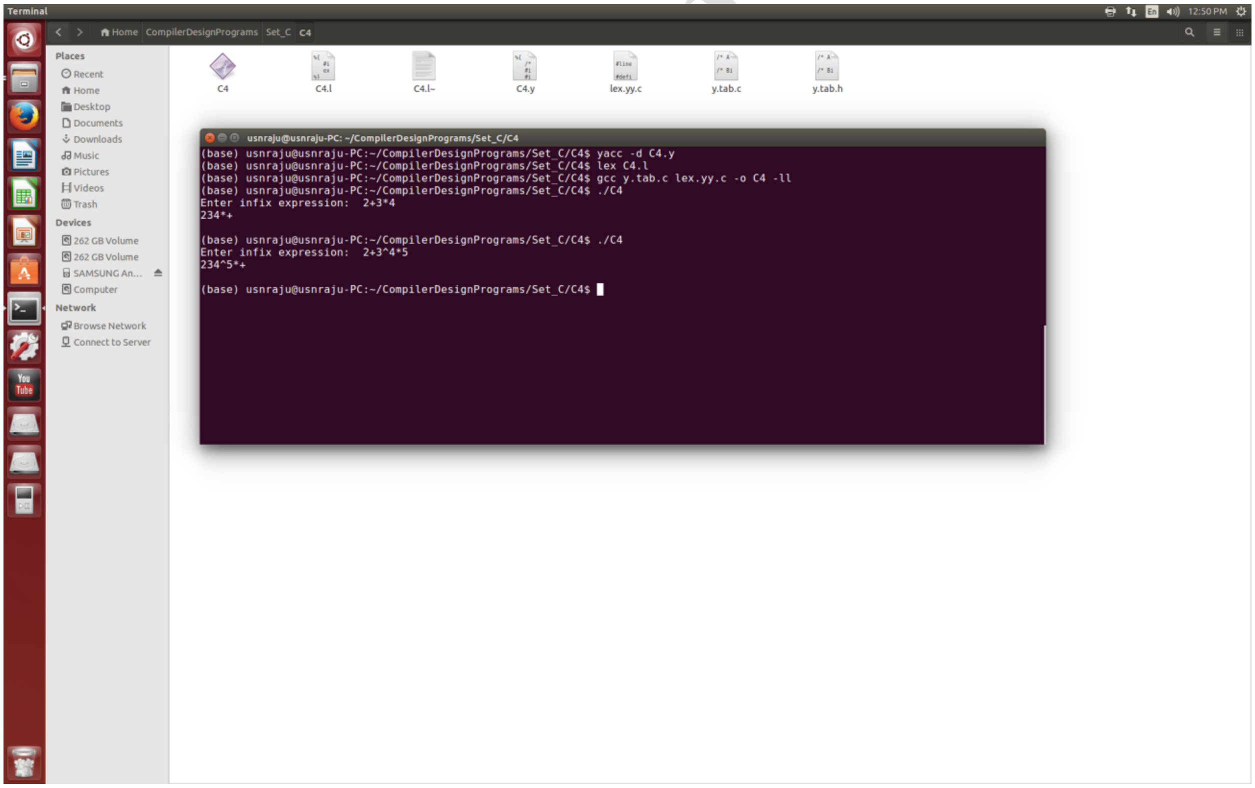
File: C4.y

```
%{
   /* Definition section */
   #include <ctype.h>
   #include<stdio.h>
   #include<stdlib.h>
%}
%token digit
/* Rule Section */
%%
/*All these grammar rules are established for operator precedence and
associativity*/
/*S prints new line after evaluating E*/
S: E {printf("\n\n");}
 ;
/*E can be evaluated to E+T or E-T or just T*/
E:  E '+'  T { printf ("+");}
 |  E '-'  T { printf ("-");}
 |  T
 ;
/*T can be evaluted to T*P or T/P or just P*/
T:  T '*'  P { printf("*");}
 |  T '/'  P { printf("/");}
 |  P
 ;
/*P can be evaluated to F^P or just F*/
P:  F '^'  P { printf ("^");}
 |  F
 ;
/*F can evaluated to E or a number*/
F: '(' E ')'
  | digit {printf("%d", $1);}
  ;
%%

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

File: C4.l

```
%{
   #include "y.tab.h"
   extern int yylval;
%}
%%
/*If the token is a Integer number,return it.*/
[0-9]+ {yylval=atoi(yytext); return digit;}
/*If the token is a space or tab,ignore it.*/
[\t]   ;
/*If the token is a new line,return 0*/
[\n]   return 0;
/*If the token didn't match with any of the above,return the first
character*/
 .     return yytext[0];
%%
```



```
(base) usnraju@usnraju-PC:~/CompilerDesignPrograms/Set_C/C4$ yacc -d C4.y
(base) usnraju@usnraju-PC:~/CompilerDesignPrograms/Set_C/C4$ lex C4.l
(base) usnraju@usnraju-PC:~/CompilerDesignPrograms/Set_C/C4$ gcc y.tab.c lex.yy.c -o C4 -ll
(base) usnraju@usnraju-PC:~/CompilerDesignPrograms/Set_C/C4$ ./C4
Enter infix expression:  2+3*4
234*+

(base) usnraju@usnraju-PC:~/CompilerDesignPrograms/Set_C/C4$ ./C4
Enter infix expression:  2+3^4*5
234^5*+

(base) usnraju@usnraju-PC:~/CompilerDesignPrograms/Set_C/C4$ 
```