```
/*****************************************************************/
/*      This file can be found in                                */
/*          ~fredh/class/423/text/Chap-B2/example2/trans.y       */
/*****************************************************************/
/*  Information to generate LALR parser function "yyparse()"     */
/*      using "yacc". The grammar that generates the language    */
/*      has the following productions:                           */
/*              E --> Prefix(E)                                  */
/*              E --> V Tail                                     */
/*              Prefix --> F                                     */
/*              Prefix --> (empty)                               */
/*              Tail --> + V                                     */
/*              Tail --> (empty)                                 */
/*  To create the function, "yyparse()" for this grammar, use    */
/*      the following commands:                                  */
/*              yacc trans.y                                     */
/*                      which produces y.tab.c                   */
/*              yacc -v trans.y                                  */
/*                      which produces y.tab.c and y.output      */
/*****************************************************************/

%{
#include "stdio.h"
#include <ctype.h>

%}

%union {
        int ival;
}

%token <ival> LPARENtok         /* token list  */
%token <ival> RPARENtok
%token <ival> PLUStok
%token <ival> Ftok
%token <ival> Vtok
%token <ival> ERRORtok

%%                              /* productions, actions  */
E       :       Prefix LPARENtok E RPARENtok
                        {printf("   E --> Prefix(E)\n"); }
        |       Vtok Tail
                        {printf("   E --> V Tail\n"); }
        ;
Prefix  :       Ftok
                        {printf("   Prefix --> F\n"); }
        |
                        {printf("   Prefix --> (empty)\n"); }
        ;
Tail    :       PLUStok Vtok
                        {printf("   Tail --> +V\n"); }
        |
                        {printf("   Tail --> (empty)\n"); }
        ;

%%
```