

```

%{
#include "listaSimbolos.h"
Lista tablaSimb;
int contCadenas=0;
%}

%union{
char *lexema;
}

%token <lexema> STRING ID NUM

%%

program : {tablaSimb=creaLS();} ID APAR CPAR ALLAVE declarations statement_list CLLAVE {imprimirTablaS();
liberaTS(); };

declarations : declarations VAR tipo var_list PC
              | declarations CONST tipo const_list PC
              ;
var_list : ID {if !(perteneceTablaS($1)) insertaTablaIdentificador($1,VARIABLE);
              else printf("Variable %s ya declarada \n",$1);}
          | var_list COMA ID {if !(perteneceTablaS($3)) insertaTablaIdentificador($3,VARIABLE);
                              else printf("Variable %s ya declarada \n",$3);}
          ;
const_list : ID ASIG expression {if !(perteneceTablaS($1)) insertaTablaIdentificador($1,CONSTANTE);
                                else printf("Constante %s ya declarada \n",$1);}
            | const_list COMA ID ASIG expression {if !(perteneceTablaS($3)) insertaTablaIdentificador($3,CONSTANTE);
                                                    else printf("Constante %s ya declarada \n",$3);}
            ;

```

```

statement : ID ASIG expression PC {if !(perteneceTablaS($1))
                                printf("Variable %s no declarada \n",$1);
                                else if (esConstante($1))
                                printf("Asignación a constante\n");}

print_item : STRING {insertaTablaString($1,CADENA,contCadenas++);}

read_list : ID {if !(perteneceTablaS($1)) printf("Variable %s no declarada \n",$1);
               else if esConstante($1) printf("Asignación a constante\n");}
  | read_list COMA ID {if !(perteneceTablaS($3)) printf("Variable %s no declarada \n",$3);
                      else if esConstante($3) printf("Asignación a constante\n");}

expression : ID {if !(perteneceTablaS($1)) printf("Variable %s no declarada \n",$1); }

%%

```

OJO: Este fichero no contiene código compilable, sino pseudocódigo, cuya intención es que entendáis lo que hay que hacer. Cada uno puede implementarlo según su criterio y **definiendo las funciones que considere convenientes**. Para ello, podéis hacer uso de las funciones declaradas y definidas en **listaSimbolos.h** y **listaSimbolos.c**, y resultará muy útil el fichero **testLS.c**, cuya finalidad es únicamente la de clarificar cómo usar dichas funciones para implementar las vuestras. Por tanto, el proyecto final debe incluir los ficheros **listaSimbolos.h** y **listaSimbolos.c**, aunque no el fichero **testLS.c**.