

# Bison Lisp Grammar

My LP's home work about Lisp syntax parse.

Save it like lisp-grammar.y and run

```
# bison lisp-grammar.y
```

it will create the file lisp-grammar.tab.c with the code of the Lisp parse.

So, run ...

```
# gcc lisp-grammar.y -o parse
```

--- cut here ---

```
/*
 * lisp-grammar.y - Lisp Grammar
 *
 * Copyright (C) 2007 Ragner Magalhaes
 *
 * This program is free software; you can redistribute it and/or modify
 * it under the terms of the GNU General Public License as published by
 * the Free Software Foundation; version 2 of the License only.
 */

%{
#define YYSTYPE double
%}

%token NUMBER SYMBOL STRING SHARP_QUOTE
%start input

%%

input : /* empty */
      | input line
      ;
```

```

line : '\n'
| s_exp '\n' { printf( "%g\n", $1 ); }
;

s_exp : atom
| list
;

atom : NUMBER
| SYMBOL
| STRING
;

list : '(' ')'
| '(' s_exp_list ')'
| '(' s_exp_list '.' s_exp ')'
| '\" s_exp
| SHARP_QUOTE s_exp
;

s_exp_list : s_exp
| s_exp_list s_exp
;

%%

#include
#include

main()
{
  yyparse();
}

yyerror( char * str )
{
  printf( "lisp: PUUUUUU %s\n", str );
}

```

```
int yylex( void )
{
int ic;

while (ic = getchar(), ic == ' ' || ic == '\t') { ; }
if (ic == EOF)
return 0;
else if (isalpha(ic))
return STRING;
else if ( isdigit( ic ) )
return NUMBER;
else if ( ic == '\"')
return SHARP_QUOTE;
else switch (ic) {
case '+':
case '-':
case '*':
case '/':
return SYMBOL;
}
return ic;
}
```