

```

/* lpard.c: parser for simple lisp, with debug output
 * grammar:  sexpr ----> alphanum | "(" sexpr tail
 *          tail ----> ")"
 */
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
void sexpr(void);
void tail(void);
char gettoken(void);
void err(int);
void nestBlanks(int n);
void enter(char* s);
void leave(char* s);

char tok; /* the next token */
int debug = 1; /* produce debug output */
int nest; /* used for debug output */

void main(void) {
    tok = gettoken();
    sexpr();
}

/* sexpr: recognize a lisp s-expression */
void sexpr(void) {
    if (isalnum(tok))
        ; /* nothing for now */
    else if (tok == '(')
        tail();
    else err(0);

    if (debug) enter("sexpr");

    if (debug) leave("sexpr");
}

/* tail: recognize tail end of a lisp s-expression */
void tail(void) {
    tok = gettoken();
    if (tok == ')')
        ; /* nothing for now */
    else {
        sexpr();
        tail();
    }

    if (debug) enter("tail");

    if (debug) leave("tail");
}

/* err: error message */
void err(int i)
{
    printf("Error number: %ld\n", i);
    exit(1);
}

/* gettoken: fetch next token (here just a char) */
char gettoken(void)
{
    char ch;
    while (isspace(ch = getchar()) && ch != EOF)
        ;
    return ch;
}

/* nestBlanks: initial output for debug output */
void nestBlanks(int n)
{
    int i;
    for (i = 0; i < n-1; i++)
        printf("| ");
}

/* enter: called on function entry (debug output) */
void enter(char* s)
{
    if (debug) {
        nest++; nestBlanks(nest);
        printf("+Enter %s, \t tok: %c\n", s, tok);
    }
}

/* leave: called last when leaving (debug output) */
void leave(char* s)
{
    if (debug) {
        nestBlanks(nest); nest--;
        printf("+Leave %s, \t tok: %c\n", s, tok);
    }
}

```

[illegible]