

```

four06% cat -n arith0.c
1 /* arith0.c: simple parser -- no output
2  * grammar:
3  * P ----> E '#'
4  * E ----> T {'+' | '-' | T}
5  * T ----> S {'*' | '/' | S}
6  * S ----> F '^' S | F '(' E ')',
7  * F ----> char | '(' E ')',
8  */
9 #include <stdio.h>
10 #include <stdlib.h>
11 #include <ctype.h>
12 char next;
13 void E(void);
14 void T(void);
15 void S(void);
16 void F(void);
17 void error(int);
18 void scan(void);
19 void enter(char);
20 void leave(char);
21 void spaces(int);
22 int level = 0;
23
24 void main(void)
25 {
26     scan();
27     E();
28     if (next != '#') error(1);
29     else printf("Successful parse\n");
30 }
31
32 void E(void)
33 {
34     enter('E');
35     T();
36     while (next == '+' || next == '-') {
37         scan();
38         T();
39     }
40     leave('E');
41 }
42
43 void T(void)
44 {
45     enter('T');
46     S();
47     while (next == '*' || next == '/') {
48         scan();
49         S();
50     }
51     leave('T');
52 }
53 void S(void)
54 {
55     enter('S');
56     F();
57     if (next == '^') {
58         scan();
59         S();
60     }
61     leave('S');
62 }
63 void F(void)
64 {
65     enter('F');
66     if (isalpha(next)) {
67         scan();
68     }
69     else if (next == '(') {
70         scan();
71         E();
72         if (next == ')') scan();
73         else error(2);
74     }
75     else {
76         error(3);
77     }
78     leave('F');
79 }
80 void scan(void)
81 {
82     while (isspace(next = getchar()))
83         ;
84     void error(int n)
85     {
86         printf("\n*** ERROR: %i\n", n);
87         exit(1);
88     }
89     void enter(char name)
90     {
91         spaces(level++);
92         printf("+-%c: Enter, \t", name);
93         printf("Next == %c\n", next);
94     }
95     void leave(char name)
96     {
97         spaces(--level);
98         printf("+-%c: Leave, \t", name);
99         printf("Next == %c\n", next);
100    }
101    void spaces(int local_level)
102    {
103        while (local_level-- > 0)
104            printf(" ");
105    }
106 }
107

```

```

four06% arith0
a + b + c #
+-E: Enter,      Next == a
+-T: Enter,      Next == a
+-S: Enter,      Next == a
| +-F: Enter,    Next == a
| +-F: Leave,    Next == +
| +-S: Leave,    Next == +
+-T: Leave,     Next == +
+-T: Enter,     Next == b
| +-S: Enter,    Next == b
| +-F: Enter,    Next == b
| +-F: Leave,    Next == +
| +-S: Leave,    Next == +
+-T: Leave,     Next == +
+-T: Enter,     Next == c
| +-S: Enter,    Next == c
| +-F: Enter,    Next == c
| +-F: Leave,    Next == #
| +-S: Leave,    Next == #
+-T: Leave,     Next == #
+-E: Leave,     Next == #
Successful parse
four06% arith0
a ^ b ^ c #
+-E: Enter,      Next == a
+-T: Enter,      Next == a
| +-S: Enter,    Next == a
| +-F: Enter,    Next == a
| +-F: Leave,    Next == ^
| +-S: Enter,    Next == b
| +-F: Enter,    Next == b
| +-F: Leave,    Next == ^
| +-S: Enter,    Next == c
| +-F: Enter,    Next == c
| +-F: Leave,    Next == #
| +-S: Leave,    Next == #
| +-S: Leave,    Next == #
+-T: Leave,     Next == #
+-E: Leave,     Next == #
Successful parse

```

```

four06% arith0
( a * b #
+-E: Enter,      Next == (
+-T: Enter,      Next == (
+-S: Enter,      Next == (
| +-F: Enter,    Next == (
| +-E: Enter,    Next == a
| +-T: Enter,    Next == a
| +-S: Enter,    Next == a
| +-F: Enter,    Next == a
| +-F: Leave,    Next == *
| +-S: Leave,    Next == *
| +-S: Enter,    Next == b
| +-F: Enter,    Next == b
| +-F: Leave,    Next == #
| +-S: Leave,    Next == #
| +-T: Leave,    Next == #
| +-E: Leave,    Next == #
*** ERROR: 2
four06% arith0
(a + b) * c #
+-E: Enter,      Next == (
+-T: Enter,      Next == (
+-S: Enter,      Next == (
| +-F: Enter,    Next == (
| +-E: Enter,    Next == a
| +-T: Enter,    Next == a
| +-S: Enter,    Next == a
| +-F: Enter,    Next == a
| +-F: Leave,    Next == +
| +-S: Leave,    Next == +
| +-T: Leave,    Next == +
| +-T: Enter,    Next == b
| +-S: Enter,    Next == b
| +-F: Enter,    Next == b
| +-F: Leave,    Next == *
| +-S: Leave,    Next == *
| +-E: Leave,    Next == )
| +-F: Leave,    Next == )
| +-S: Leave,    Next == *
+-S: Leave,     Next == c
+-S: Enter,     Next == c
| +-F: Enter,    Next == c
| +-F: Leave,    Next == #
| +-S: Leave,    Next == #
| +-T: Leave,    Next == #
+-E: Leave,     Next == #
Successful parse

```