```
runner% cat -n eval1.c
     1  /* arith.c: simple parser -- bold  evaluates */
     2  #include <stdio.h>
     3  #include <stdlib.h>
     4  #include <ctype.h>
     5  #include <math.h>  /* for pow */
     6  char next;
     7  double E(void);
     8  double T(void);
     9  double S(void);
    10  double F(void);
    11  void error(void);
    12  void scan(void);
    13  double evaluate(double arg1, char op, double arg2);
    14
    15  void main(void)
    16  {
    17     double res;
    18     scan();
    19     res = E();
    20     if (next != '#') error();
    21     else
    22        printf("Result is: %.5f\n", res);
    23  }
    24
    25  double E(void)
    26  {
    27     char save;
    28     double res, arg1, arg2;
    29     res = T();
    30     while (next == '+' || next == '-') {
    31        save = next;
    32        arg1 = res;
    33        scan();
    34        arg2 = T();
    35        res = evaluate(arg1, save, arg2);
    36     }
    37     return res;
    38  }
    39
    40  double T(void)
    41  {
    42     char save;
    43     double res, arg1, arg2;
    44     res = S();
    45     while (next == '*' || next == '/') {
    46        save = next;
    47        arg1 = res;
    48        scan();
    49        arg2 = S();
    50        res = evaluate(arg1, save, arg2);
    51     }
    52     return res;
    53  }
    54
    55  double S(void)
    56  {
    57     char save;
    58     double res, arg1, arg2;
    59     res = F();
    60     if (next == '^') {
    61        save = next;
    62        arg1 = res;
    63        scan();
    64        arg2 = S();
    65        res = evaluate(arg1, save, arg2);
    66     }
    67     return res;
    68  }
    69
    70  double F(void)
    71  {
    72     char save;
    73     double res;
    74     if (isdigit(next)) {
    75        save = next;
    76        scan();
    77        return (double)(save - '0');
    78     }
    79     else if (next == '(') {
    80        scan();
    81        res = E();
    82        if (next == ')') scan();
    83        else error();
    84        return res;
    85     }
    86     else {
    87        error();
    88        return 0;
    89     }
    90  }
    91  void scan(void)
    92  {
    93     while (isspace(next = getchar()))
    94        ;
    95  }
    96  void error(void)
    97  {
    98     printf("\n*** ERROR ***\n");
    99     exit(1);
   100  }
   101  double evaluate(double arg1, char op, double arg2)
   102  {
   103     switch (op) {
   104     case '+': return arg1 + arg2;
   105     case '-': return arg1 - arg2;
   106     case '*': return arg1 * arg2;
   107     case '/': return arg1 / arg2;
   108     case '^': return pow(arg1, arg2);
   109     default: error(); return 0;
   110     }
   111  }
```

```
runner% cc -o eval1 -lm eval1.c   (Need -lm because of pow)

runner% eval1
2+3*4#
Result is: 14.00000

runner% eval1
3*4+5#
Result is: 17.00000

runner% eval1
(2+3)*4#
Result is: 20.00000

runner% eval1
(3*(2+4)/(5+1))-2#
Result is: 1.00000

runner% eval1
(5+3)^(2+1)^2#
Result is: 134217728.00000

runner% eval1
2+3*4^5*6+7#
Result is: 18441.00000

runner% eval1
2 + (3*(4^5)*6) + 7#          (same with parentheses)
Result is: 18441.00000

runner% eval1
((3^2-4*1*2)^(1/2)-3)/(2*1)#
Result is: -1.00000

runner% eval1
((2-3)^((4+1)*5)/6-(2-4)*7)-8#
Result is: 5.83333

runner% eval1
3^4^2#
Result is: 43046721.00000

runner% eval1
3^(4^2)#                      (same with parentheses)
Result is: 43046721.00000

runner% eval1
(3^4)^2#
Result is: 6561.00000

runner% eval1
9 - 5 - 3 #
Result is: 1.00000
```

```
runner% eval1
8/3/5#
Result is: 0.53333

runner% eval1
8/(3/5)#
Result is: 13.33333

runner% eval1
1+1/(2+1/(2+1/(2+1/(2+1/(2+1/(2+1/(2+1/(2+1/(2+1/(2+1
/(2+1/(2+1/(2+1/(2+1/(2+1/(2+1/(2+1/(2+1/(2+1/(2
))))))))))))))))))))#
Result is:      1.4142135623730949
(Note:sqrt(2) = 1.414213562373095048801688724209...)

runner% eval1
2+1/(1+1/(2+1/(1+1/(1+1/(4+1/(1+1/(1+1/(6+1/(1+1/(1+1
/(8+1/(1+1/(1+1/(2*5+1/(1+1/(1+1/(2*6+1/(1+1/(1+1/(2*
7+1  ))))))))))))))))))))#
Result is: 2.7182818284590455
(Note:e =  2.71828182845904523526028747 1352...)

runner% eval1
3+1/(7+1/(5*3+1/(1+1/((4*(9*8+1))  ))))#
Result is: 3.1415926530119025

runner% eval1
3+1/(7+1/(5*3+1/(1+1/((4*(9*8+1))+1/(1+1/(1+1/(1+1/(2
+1/(1+1/(3+1/(1+1/(2*7+1/(2+1) ))))))))))))#
Result is: 3.1415926535897931
(Note:pi = 3.14159265358979323846264338327 9...)

four06% eval1
1+1/(1+1/(1+1/(1+1/(1+1/(1+1/(1+1/(1+1/(1+1/(1+1/(1+1/(1+1
/(1+1/(1+1/(1+1/(1+1/(1+1/(1+1/(1+1/(1+1/(1+1/(1+1/(1+1/(1+1
/(1+1/(1+1/(1+1/(1+1/(1+1/(1+1/(1+1/(1+1/(1+1/(1+1/(1+1/(1+1
))))))))))))))))))))))))))))))))#
Result is:   1.6180339887498940
(Note: φ =    1.6180339887498948482045868343 65 ...
       =    (0.5)*(1 + sqrt(5)) )
```