

```

#include <stdlib.h>
#include <stdio.h>
#include <assert.h>

static FILE *f;
static int ch;
static unsigned int val;
enum { plus, minus, times, divide, mod, lparen, rparen, number, eof,
illegal };

static void SInit( char* filename )
{
    ch = EOF;
    f = fopen( filename, "r+t" );
    if( f != NULL ) ch = getc(f);
}
static void Number()
{
    val = 0;
    while( ('0' <= ch) && (ch <= '9') ) {
        val = val * 10 + ch - '0';
        ch = getc(f);
    }
}
static int SGet()
{
    register int sym;

    while( (ch != EOF) && (ch <= ' ') ) ch = getc(f);
    switch( ch ) {
        case EOF : sym = eof; break;
        case '+' : sym = plus; ch = getc(f); break;
        case '-' : sym = minus; ch = getc(f); break;
        case '*' : sym = times; ch = getc(f); break;
        case '/' : sym = divide; ch = getc(f); break;
        case '%' : sym = mod; ch = getc(f); break;
        case '(' : sym = lparen; ch = getc(f); break;
        case ')' : sym = rparen; ch = getc(f); break;
        case '0' : case '1' : case '2' : case '3' : case '4' :
        case '5' : case '6' : case '7' : case '8' : case '9' :
            sym = number; Number(); break;
        default : sym = illegal;
    }
    return sym;
}

static int sym;
static void Expr();

```

```

static void Factor()
{
    assert( (sym == number) || (sym == lparen) );
    if( sym == number ) {
        sym = SGet();
    } else {
        sym = SGet();
        Expr();
        assert( sym == rparen );
        sym = SGet();
    }
}

static void Term()
{
    Factor();
    while( (sym == times) || (sym == divide) || (sym == mod)) {
        sym = SGet();
        Factor();
    }
}

static void Expr()
{
    if ( ( sym == minus ) || ( sym == plus ) ) {
        sym = SGet();
    }

    Term();

    while( (sym == plus) || (sym == minus) ) {
        sym = SGet();
        Term();
    }
}

int main( int argc, char* argv[] )
{
    register int result;

    if( argc == 2 ) {
        SInit(argv[1]);
        sym = SGet();
        Expr();
        assert( sym == eof );
    } else {
        printf("usage: expreval <filename>\n");
    }
    return 0;
}

```