```
/*
 * hw7.c
 *
 *  Created on: Nov 3, 2014
 *      Author: scot
 */
/*
 * main.c
 *
 *  Created on: Nov 3, 2014
 *      Author: Scot Matson
 *      Assn: 7
 *      Cour: CS49C
 *      Sect: 1
 */
#include <stdio.h>
#include <stdlib.h>     /* for atof() */
#include "calcf.h"
#define MAXOP   100     /* max size of operand or o
perator */


/* reverse polish calculator */
int RPNCalc (FILE *fpi, FILE *fpo) {


    int type;
    double op1, op2, str;
    char s[MAXOP];

    while ((type = getop(s, fpi)) != EOF) {
        switch(type) {
        case NUMBER:
            push(atof(s));
            break;
        case '+':
            push(pop() + pop());
            break;
        case '*':
            push(pop() * pop());
            break;
        case '-':
            op2 = pop();
            push(pop() - op2);
            break;
        case '/':
            op2 = pop();
            if (op2 != 0.0) {
                push(pop() / op2);
            }
            else {
                fprintf(fpo, "error: zero divisor\n
");
            }
            break;
        case '\n':
            op2 = pop();
            fprintf(fpo, "\t%.16g\n", op2);
            push(op2);
            break;
        case '=':
            op2 = pop();
            fprintf(fpo, "\t%.16g\n", op2);
            push(op2);
            break;
        case 'X':
            op1 = pop();
            op2 = pop();
            push(op1);
            push(op2);
            break;
        case 'S':
            str = pop();
            break;
        case 'R':
            push(str);
            break;
        default:
            fprintf(fpo, "error: unknown command %s
\n", s);
            break;
        }
    }
    return 0;
```

```
}
```