CS 1713, Stack Implementation and RPN Calculator, Thu Apr 23 1998

```
runner% cat stack_rpn.c
/* stack_rpn.c: Reverse Polish calculator.*/
#include <stdio.h>
#include <math.h>
#include <stdlib.h>
#include "stack_double.h"
void error(int );
void main(void)
{
    char c;
    double op1, op2;
    while ((c = getchar()) != '\n') {
        if (isdigit(c)) {
            if (!full()) push((double)(c - '0'));
            else error(1);
        }
        else {
            if (!empty()) op2 = pop();
            else error(2);
            if (!empty()) op1 = pop();
            else error(2);
            switch (c) {
            case '+': push(op1 + op2); break;
            case '-': push(op1 - op2); break;
            case '*': push(op1 * op2); break;
            case '/': push(op1 / op2); break;
            default: error(3); break;
            }
        }
    }
    if (!empty()) printf("Final value: %.2f\n", pop());
    else error(4);
    exit(0);
}
void error(int err_no)
{
    switch (err_no) {
    case 1:printf("Stack overflow\n"); exit(1); break;
    case 2:printf("Too few operands\n");exit(1); break;
    case 3:printf("Illegal operator\n");exit(1); break;
    case 4:printf("No final value\n");exit(1); break;
    }
}
-----------------------------------------------------
runner% cat stack_double.h
/* stack_double.h: stack header file */
typedef double Stacktype;
Stacktype pop(void);
void push(Stacktype);
int empty(void);
int full(void);
-----------------------------------------------------
runner% cat stack.c
/* stack.c: stack implementation */
#include <stdio.h>
#include "stack_double.h"
#define S_SIZE 12

static Stacktype s[S_SIZE];
static int sp = 0;

/* pop: remove top element from stack */
Stacktype pop(void)
{
    if (!empty()) return s[--sp];
    else {
        fprintf(stderr,"Underflow\n");
        return 0;
    }
}
/* push: add new element to stacktop */
void push(Stacktype c)
{
    if (!full()) s[sp++] = c;
    else
        fprintf(stderr,"Overflow\n");
}
/* empty: check if stack is empty */
int empty(void) { return sp == 0; }
/* full: check if stack is full */
int full(void)  { return sp == S_SIZE; }
-----------------------------------------------------
runner% cat makefile
stack: stack_rpn.c  stack.c  stack_double.h
    cc -g -o stack_rpn stack_rpn.c stack.c -lm
lint:
    lint -m -u stack_rpn.c stack.c -lm
-----------------------------------------------------
runner$ stack_rpn
34+
Final value: 7.00
runner$ stack_rpn
234+*
Final value: 14.00
runner$ stack_rpn
23+4*
Final value: 20.00
runner$ stack_rpn
235*+
Final value: 17.00
runner$  stack_rpn
23
Final value: 3.00
runner$ stack_rpn
234+++
Too few operands
```