```c
#include <stdio.h>
#include <string.h> // for strlen function
#include <math.h>   // for pow function
#include <ctype.h>  // for isdigit function

// Function to compute the result of the operator applied to two operands
double compute(char symbol, double op1, double op2) {
    switch(symbol) {
        case '+': return op1 + op2;
        case '-': return op1 - op2;
        case '*': return op1 * op2;
        case '/': return op1 / op2;
        case '$': // Could be used for exponentiation (same as ^)
        case '^': return pow(op1, op2);
        default: return 0; // Error case
    }
}

int main() {
    char postfix[20] = "56+437-*/"; // Example postfix expression
    double st[20], op1, op2;
    int top = -1, i;

    // Iterate through each character in the postfix expression
    for (i = 0; postfix[i] != '\0'; i++) {
        if (isdigit(postfix[i])) {
            // Push operands (digits) to the stack
            st[++top] = postfix[i] - '0';
        } else {
            // Pop two operands from the stack
            op2 = st[top--];
            op1 = st[top--];
            // Perform the operation and push the result back onto the stack
            st[++top] = compute(postfix[i], op1, op2);
        }
    }

    // The final result is the only value left in the stack
    printf("Result is %lf\n", st[top]);
    return 0;
}
```

```c
#include <stdio.h>
void hanoi(int n, char S, char T, char D) {
    if (n == 0)
    return;
    hanoi(n - 1, S, D, T);
    printf("Move disc %d from %c to %c\n", n, S, D);
    hanoi(n - 1, T, S, D);
}

int main() {
    int n;
    printf("Enter number of discs\n");
    scanf("%d", &n);
    hanoi(n, 'A', 'B', 'C');

    return 0;
}
```