Make a subdirectory hw7 in your home directory for this assignment. The objective of this assignment is to let you practice with managing multiple files and using the make utility.

1 Postfix Notation

The standard way to write an expression, such as \( a + b \times c \) is called the infix notation. The same expression in the postfix notation is \( a \ b \ c \ * \ + \). You can convert between infix and postfix using an online tool here: [http://www.mathblog.dk/tools/infix-postfix-converter/](http://www.mathblog.dk/tools/infix-postfix-converter/).

Read the provided code `calc.h`, `getch.c`, `getop.c`, `main.c`, and `stack.c`. The program implements a simple calculator that reads input from keyboard in the postfix notation, and performs calculation for floating-point numbers. You can create the executable `calc` by entering the command `make`. Run the program a few times to become familiar with it.

Your tasks in this assignment is to convert the code from floating-point calculation to integer calculation – mostly replacing `double` with `int`. The original code uses `atof()` to convert ASCII to a floating-point number. You can use `atoi()` to convert ASCII to integer. Additionally, you add functionality to the calculator by implementing the following operations.

- Four bitwise operations: AND (\&), OR (\|), XOR (^), and NOT (\~). Note that AND, OR, and XOR are binary operators, but NOT is a unary operator. For example, \( a \ b \ & \) is the bitwise AND of \( a \) and \( b \); \( a \ \sim \) is the bitwise NOT of \( a \).

- Three comparison operations: GREATER-THAN (>), EQUAL-TO (=), and LESS-THAN (<). The resulting values are either zero or one. Note that EQUAL-TO is denoted =, rather than == as in C.

- One logical operation NOT (!). It returns zero for a non-zero number, and returns one for zero. For example, \( 0 \ ! \) is 1, and \( 5 \ ! \) is 0.

- One ternary operation ?: The usual ternary expression \( a \ ? \ b \ : \ c \) in C will be written as \( c \ b \ a \ ? \). If \( a \) is non-zero, the value of the expression is \( b \), otherwise it is \( c \).

Run your program like this. `./calc < testPostfix.txt`