CSC 361
Homework 5

Name: _____________________________________

1. Assume the following rules of associativity and precedence for expressions

   Precedence:  Highest  * , / , not
               + , - , &
               - (unary minus)
               = , /=, < , <= , > , >= , >
               and
          Lowest   or, xor

   Associativity  Left to right

   Show the order of evaluation of the following expressions by parenthesizing all
   subexpressions and placing a superscript on the right parenthesis to indicate order. For example,
   for the expression
   a + b * c + d
   the order of evaluation would be represented as
   ( ( a + ( b * c ) ) ) + d )

   a. a - c * -b / a

   b. a * ( b - 1 ) - c / d

   c. a > b xor c and d <= 1
2. Let the function FUN be defined as
   ```c
   int FUN ( int & k )
   {
     k = k - 2;
     return 3*k
   }
   ```

   Suppose FUN is used in a program as follows
   ```c
   I = 100;
   sum1 = ( I/2 ) + FUN( I );
   J = 50;
   sum2 = FUN( J ) + ( J / 2 );
   ```

   What are the values of sum1 and sum2
   a. if the operands in the expression are evaluated left to right?
      sum1 = _____
      sum2 = _____
   b. if the operands in the expression are evaluated right to left?
      sum1 = _____
      sum2 = _____

3. Rewrite the following code segment using a loop structure in C/C++

   ```c
   k = (j + 13) / 27;
   place:
   if ( k > 0 )
      goto out;
   k++;
   n = n - k - 1;
   goto place;
   out:
   k++;  
   ```

   Answer:
4. Suppose that C++ did not have short circuiting in Boolean expressions. Rewrite the following code to avoid dividing by zero:

```cpp
cin << a << b;
if ( a > 0 and b / a < 5 )
    cout << a + b;
else
    cout << b;
```

Answer: