Before attempting these problems, carefully read the web pages on loop invariants, links to which can be found on the course web page.

1. Complete the following program so that it is consistent with its documentation. Recall that, in Java, the remainder obtained when dividing $a$ by $b$ is given by the expression $a \% b$. Hence, to say that $a$ is divisible by (or, equivalently, is a multiple of) $b$ is to say that $a \% b = 0$.

```java
/* pre: x > 0 & y > 0
* post: value returned is the least common (positive) multiple of x and y
*/
public int lcm(int x, int y) {
    int k = ___________;  

    /* loop invariant: k is a positive multiple of x such that 
    * no smaller positive multiple of x is divisible by y. 
    * Or, to state it more technically: for some integer q > 0, 
    * k = qx and none of x, 2x, 3x, ..., (q-1)x is divisible by y 
    */
    while ( ________________________________ ) {

    }

    return _________________;
}
```
2. Develop a solution for the three-color partitioning problem. Given is an array $a[]$ each of whose elements can be classified as being exactly one of RED, WHITE, or BLUE. Using $N$ as a shorthand for $a.length$, the postcondition is as follows:

\[
\begin{array}{cccc}
0 & j & k & N \\
\hline
a & \text{all RED} & \text{all WHITE} & \text{all BLUE} \\
\hline
\end{array}
\]

In words, this says that $0 \leq j \leq k \leq N$ and that every element in $a[0..j - 1]$ is RED, every element in $a[j..k - 1]$ is WHITE, and every element in $a[k..N - 1]$ is BLUE.

The loop invariant should be as follows:

\[
\begin{array}{cccc}
0 & i & j & k & N \\
\hline
a & \text{all RED} & \text{?} & \text{all WHITE} & \text{all BLUE} & \text{\&\&} 0 \leq i \leq j \leq k \leq N \\
\hline
\end{array}
\]

In words, this says that $0 \leq i \leq j \leq k \leq N$ and that every element in $a[0..i - 1]$ is RED, every element in $a[j..k - 1]$ is WHITE, and every element in $a[k..N - 1]$ is BLUE. (It says nothing about the values of elements in $a[i..j - 1]$.) Notice that we obtained the invariant by replacing the first occurrence of $j$ in the postcondition by a fresh variable $i$.

Complete the program on the following page by supplying an expression wherever there is an underline and a code segment for each of the three branches of the if statement. The program is not to modify the array except by swapping array elements using the `swap()` method: the invocation `swap(a, t, s)` causes the values in locations $t$ and $s$ of $a[]$ to be swapped.
/* precondition: every element in a[0..N-1] is either RED, WHITE, or BLUE */

i = _____; j = _____; k = _____;

/* invariant: 0<=i<=j<=k<=N && every element in a[0..i-1] is RED &&
  every element in a[j..k-1] is WHITE &&
  every element in a[k..N-1] is BLUE */

while ( ___________ ) {

    if (isRed(a[ _____ ])) {

    }

    else if (isWhite(a[ _____ ])) {

    }

    else { // isBlue(a[ _____ ])

    }

/* post: 0<=j<=k<=N && every element in a[0..j-1] is RED &&
       every element in a[j..k-1] is WHITE &&
       every element in a[k..N-1] is BLUE */