Before attempting these problems, you should read the document on loop invariants to which there is a link on the course web page.

1. You have a jar containing \( k \) red marbles and \( m \) blue marbles, where \( k + m > 0 \). You also have an unlimited supply of blue marbles off to the side. Repeat the following procedure until exactly one marble remains in the jar:

Choose (any) two marbles from the jar;
if both are blue {
    put one back into the jar;
    discard the other one;
}
else if both are red {
    put one back into the jar;
    discard the other one;
    put a (new) blue marble into the jar;
}
else {  // one marble of each color was chosen
    discard the red one;
    put the blue one back into the jar;
}

(a) Give a convincing argument that the procedure necessarily terminates.

**Hint:** It is not the case that the number of marbles in the jar necessarily decreases on each iteration. However, if you assign one weight to each red marble and another weight to each blue marble, you can show that the total weight of the marbles in the jar decreases on each iteration. It is up to you to come up with appropriate weights. **End of hint.**

(b) Make the strongest statement you can regarding the color of the last marble remaining in the jar. Is it always the same, regardless of the values of \( k \) and \( m \)? Or, failing that, can it be determined from \( k \) and \( m \)? In either case, give a convincing argument, making use of the loop invariant concept.

If, on the other hand, the color of the last marble cannot be determined from \( k \) and \( m \), give an example demonstrating this. (That is, for some particular values of \( k \) and \( m \), show that one execution of the procedure yields a blue marble and another execution yields a red marble.)

*Note:* For a slight deduction in points, you may assume not only that \( k + m > 0 \) but, more strongly, that both \( k > 0 \) and \( m > 0 \). If you make this assumption, be sure to mention it.
In class, we developed a program for solving the two-color version of the Dutch National Flag problem. The program was developed using a proposed loop invariant (derived from the post-condition) as a guide. Here you are asked to develop an alternative solution to the same problem. As in class, your program is not to modify the array except by swapping array elements. (Assume that there is a method \textit{swap} such that the effect of making the call \textit{swap}(a,k,j) is to swap the values in \textit{a}[k] and \textit{a}[j].)

The pre-condition is that every element in the array \textit{a[]} satisfies exactly one among the two predicates \textit{isRed()} and \textit{isBlue()}. The postcondition is as indicated in this picture:

\[
\begin{array}{c|cc|c}
0 & k & N \\
\hline
\text{a} & \text{all RED} & \text{all BLUE} & \& \ 0\leq k \leq N \\
\end{array}
\]

(We use \textit{N} as an abbreviation for the length of \textit{a}, which in Java is written \textit{a.length}.) In words, this says that \(0 \leq k \leq N\) and that every element in the array segment \textit{a}[0..k-1] is Red and every element in \textit{a}[k..N-1] is Blue.

More formally, we could express this in the language of predicate logic as follows:

\[
(\forall i \mid 0 \leq i < k : \text{isRed}(a[i])) \land (\forall i \mid k \leq i < N : \text{isBlue}(a[i])) \land 0 \leq k \leq N
\]

The loop invariant of your program should be as suggested by this picture:

\[
\begin{array}{c|c|cc|c}
0 & k & m & N \\
\hline
\text{a} & \text{all RED} & \text{?} & \text{all BLUE} & \& \ 0\leq k \leq m \leq N \\
\end{array}
\]

In words, this says that \(0 \leq k \leq m \leq N\) and that every element in \textit{a}[0..k-1] is Red and every element in \textit{a}[m..N-1] is Blue. More formally, we can express this by

\[
(\forall i \mid 0 \leq i < k : \text{isRed}(a[i])) \land (\forall i \mid m \leq i < N : \text{isBlue}(a[i])) \land 0 \leq k \leq m \leq N
\]

Arrive at your solution by correctly replacing each question mark in the following with either an expression or a sequence of statements, whichever is appropriate. The initialization of \textit{k} and \textit{m} should establish the invariant by making the \textit{?}-region cover the entire array. Each iteration of the loop should decrease by at least one the length of the \textit{?}-region.
k = ?;  m = ?;

while ( ? ) {
    if (isRed(a[k]))
        { ? }  
    else /* isBlue(a[k]) */
        { ? }  
}