CSC 212 Tutorial #2 Solution
Recursion

Problem 1

```
public static int count(int[] arr, int x) {
    return countRec(arr, x, 0);
}

private static int countRec(int arr[], int x, int pos) {
    if (pos == arr.length)
        return 0;
    if (arr[pos] == x)
        return 1 + countRec(arr, x, pos + 1);
    return countRec(arr, x, pos + 1);
}
```

Problem 2

```
public static <T> boolean isPalindrome(T[] arr, int n) {
    return isPalindromeRec(arr, 0, n - 1);
}

private static <T> boolean isPalindromeRec(T[] arr, int leftIndex,
    int rightIndex) {
    if (leftIndex == rightIndex || leftIndex > rightIndex)
        return true;
    if (!arr[leftIndex].equals(arr[rightIndex]))
        return false;
    return isPalindromeRec(arr, leftIndex + 1, rightIndex - 1);
}
```

Problem 3

```
public static void solveTowersOfHanoi(int n) {
    solveTowersOfHanoiRec(n, 'a', 'c', 'b');
}

private static void solveTowersOfHanoiRec(int n, char src, char
   dest, char spare) {
    if (n == 1) {
        System.out.printf("Move disk 1 from peg %c to %c\n", src,
           dest);
```

```
            return;
        }
        solveTowersOfHanoiRec(n - 1, src, spare, dest);
        System.out.printf("Move disk %d from peg %c to %c\n", n, src,
            dest);
        solveTowersOfHanoiRec(n - 1, spare, dest, src);
    }
```