1 Explain what happens when a negative value is passed to the `Factorial()` method programmed in class. (3)

2 Modify `Factorial()` in order to handle the error encountered in Question #1 elegantly. (3)

3 Evaluate `mystery(2, 25)` and `mystery(3, 11)` for the following recursive function. (5)

```
public static int mystery(int a, int b) {
  if (b == 0)
    return 0;
  if (b % 2 == 0)
    return mystery(a + a, b / 2);
  return mystery(a + a, b / 2) + a;
}
```

4 Explain what is wrong with the following recursive function. (5)

```
public static String work(int n) {
  String s = work(n - 3) + n + work(n - 2) + n;
  if (n <= 0)
    return "";
  return s;
}
```

5 *Quicksort.* Implement the *Quicksort Algorithm* as detailed below. **Input:** An array, `A`, of integer values. (10)

1. Pick any element from `A` to act as the `pivot` value.
2. Reorder the array so that all elements in `A` less than or equal to `pivot` are in a lower index than `pivot` and all elements in `A` greater than or equal to `pivot` are in a higher index than `pivot`.
3. Recursively apply steps #1 & 2 on the subarray of elements less than `pivot` and on the subarray of elements greater than `pivot`.

Output: An array, `A`, of integer values in ascending order.

6 *Partitions.* A *partition* of a positive integer, n, is its representation as a sum of positive integers, $n = p_1 + p_2 + \cdots + p_k$. Write a method that prints out all possible partitions of a given positive integer, n. Consider only partitions where $p_1 \leq p_2 \leq \cdots \leq p_k$. (20)