Recursion:

1. What does the following recursive function do?

   ```java
   public int function(int[] array, int n) {
       if (n == 0)
           return 0;
       else
           return function(array, n-1) + array[n-1];
   }
   ```

   Try running the line below as the computer would:

   ```java
   int[] array = {5, 8, 2};
   int x = function(array, array.length);
   ```

   What is x?

2. Write a recursive function to compute the factorial function: \( n! = n(n - 1)(n - 2) \cdots 1 \).
3. The Fibonacci numbers are a sequence of integers where each entry is the sum of the previous two entries, starting with 1 and 1:

\[
\begin{array}{c|cccccccc}
 n & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\
 F(n) & 1 & 1 & 2 & 3 & 5 & 8 & 13 & 21 \\
\end{array}
\]

Write a recursive function to compute the \( n \)th Fibonacci number.

4. Write a recursive function to compute the greatest common divisor (GCD) of two numbers \( n \) and \( m \). For example, \( \text{GCD}(16,12) = 4 \).

5. Redo each of the previous examples using tail recursion.