1. Write an algorithm that will parse through an array of characters and count the frequency of each instance. How could you do this recursively?

2. Given an array of integers, print out every other integer in the array. Can you do this recursively?

3. Write a C++ recursive function for the given recursive relation.

\[
   t_n = \begin{cases} 
   t_{n-1} + n & \text{if } n \text{ is even} \\
   t_{n-1} + t_{n-2} & \text{if } n \text{ is odd}
   \end{cases} \text{ where } t_0 = 1, \text{ and } t_1 = 2
\]

4. If we are given a “full” binary tree of height 4, how many vertices are there? What is the path length? Can you derive a formula to determine the path length of a full binary tree given that the height of the tree is \( n \)?

5. Draw the recursive tree for the following recursive function:

   ```cpp
   int fun(int n, int r) {
     if ( n < 1 ) return 0;
     if ( n == 1 && r < n) return r;
     if ( n%2 == 0) return fun(n-1,r-2);
     else return fun(n-1,r-1) + fun(n-1,r);
   }
   ```

   Suppose the initial call is `fun(5,3);`

   What is the maximum depth of the stack?

   How many function calls are required?