1. Consider the following naïve recursion code for Fibonacci numbers. Trace the code and draw the recursion tree for F(5)

```
public static long F(int n) {
    if (n == 0) return 0;
    if (n == 1) return 1;
    return F(n-1) + F(n-2);
}
```

2. Consider the following dynamic programming code for Fibonacci numbers. Trace the code for \( n \) being 5, and write down the values of the elements in array \( F \):

```java
long[] F = new long[n+1];
F[0] = 0; F[1] = 1;
for (int i = 2; i <= n; i++)
    F[i] = F[i-1] + F[i-2];
```
3. Consider the following recursion code with memorization for Fibonacci numbers

```java
static long[] memo = new long[100];
public static long F(int n)
{
    if (n == 0) return 0;
    if (n == 1) return 1;
    if (memo[n] == 0)
        memo[n] = F(n-1) + F(n-2);
    return memo[n];
}
```

a. Trace the code and draw the recursion tree for F(5)

b. Consider the code given earlier and keep all the statements except change the if body to the following, which just switches the positions for F(n-1) and F(n-2):

```
    memo[n] = F(n-2) + F(n-1);
```

Now trace the slightly changed code and draw the new recursion tree for F(5):
4. a. Consider the following DP code for coin change, draw the 2D array $C[i, j]$, and trace the algorithm for an change amount of 8 and a coin set $\{1, 4, 5\}$.

$$
c(i, j) = \begin{cases} 
0 & \text{if } j = 0 \\
\frac{i}{d_i} & \text{if } i = 1 \\
\infty & \text{if } j < 0 \\
\min(c(i - 1, j), 1 + c(i, j - d_i)) & \text{otherwise}
\end{cases}
$$

b. The mathematical definition above for the function $c(i, j)$ has a problem for the base case $i = 1$. It works when the first coin value $d_1$ is 1. But if $d_1$ is not 1, how to process the base case needs to be changed. Consider an change amount of 8 and the coin set $\{2, 4, 5\}$. You can start a trace by using the definition above for $c(i, j)$ and identify the problem. Try to develop a solution to fix the problem, change the $c(i, j)$ definition for the base case $i = 1$, and trace your revised definition on amount $= 8$ and coins $= \{2, 4, 5\}$.

5. Consider the following pair of mutually recursive functions, what is the value of $g(2)$?

```java
public static int f(int n) {
    if (n == 0) return 0;
    return f(n-1) + g(n-1);
}

public static int g(int n) {
    if (n == 0) return 0;
    return g(n-1) + f(n);
}
```

6. Trace Permutations.java
   [https://introcs.cs.princeton.edu/java/23recursion/Permutations.java.html](https://introcs.cs.princeton.edu/java/23recursion/Permutations.java.html)