Task 1:
Tower of Hanoi Solver Create a program that solves the Tower of Hanoi puzzle for n disks. The solution should use recursion to move disks between three pegs (source, auxiliary, and destination) according to the game's rules. The program should print out each move required to solve the puzzle.
ANS:

```
package com.Day19;
public class TowerOfHanoi {

   // Function to solve the Tower of Hanoi puzzle
   public static void towerOfHanoi(int n, char source, char auxiliary,
char destination) {
      // Base case: when only one disk is to be moved
      if (n == 1) {
         System.out.println("Move disk 1 from " + source + " to " +
destination);
         return;
      }
      // Move n-1 disks from source to auxiliary, so they are out of the way
      towerOfHanoi(n - 1, source, destination, auxiliary);
      // Move the nth disk from source to destination
      System.out.println("Move disk " + n + " from " + source + " to " +
destination);
      // Move the n-1 disks that we left on auxiliary to destination
      towerOfHanoi(n - 1, auxiliary, source, destination);
   }
   public static void main(String[] args) {
      int n = 3; // Number of disks
      towerOfHanoi(n, 'A', 'B', 'C'); // Solve the puzzle for n disks
   }
}
```

OUTPUT:

```
Move disk 1 from A to C
Move disk 2 from A to B
Move disk 1 from C to B
Move disk 3 from A to C
Move disk 1 from B to A
Move disk 2 from B to C
```

Move disk 1 from A to C