Day-13_Assignment

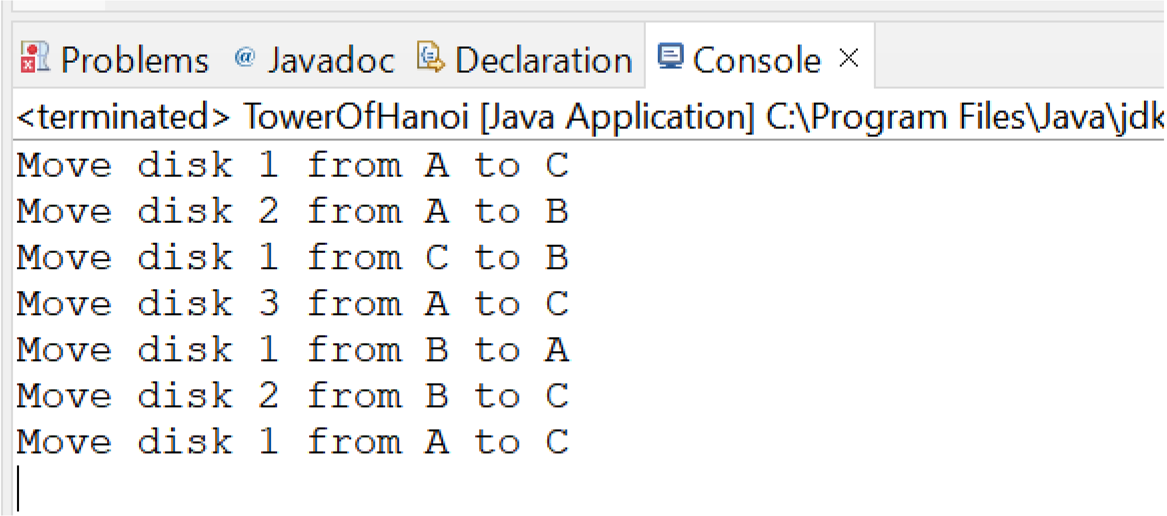
Manjula Nannuri1

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.

```
package com.example;

public class TowerOfHanoi {

public static void towerOfHanoi(int n, char source, char auxiliary, char
destination) {

if (n == 1) {

System.out.println("Move disk 1 from " + source + " to " + destination);

return;

}

towerOfHanoi(n - 1, source, destination, auxiliary);

System.out.println("Move disk " + n + " from " + source + " to " +
destination);

towerOfHanoi(n - 1, auxiliary, source, destination);

}

public static void main(String[] args) {

int n = 3;

towerOfHanoi(n, 'A', 'B', 'C');

}



}
```

OUTPUT:



```
<terminated> TowerOfHanoi [Java Application] C:\Program Files\Java\jdk
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
```

Task 2: Traveling Salesman Problem

Create a function int FindMinCost(int[,] graph) that takes a 2D array representing the graph where graph[i][j] is the cost to travel from city i to city j. The function should return the minimum cost to visit all cities and return to the starting city. Use dynamic programming for this solution.