Assignment 1 (Basic Java)

Instructions

1. The first two questions are DSA based questions, they will help you get familiarized with the data structures present in java and also enhance your problem-solving skills
2. The third question is an open-ended LLD (Low-level design) problem, it will help you get your design concepts clear
3. Feel free to comment anywhere in the doc if you are not able to understand anything
4. There is no timeline for submission, attempt with full honesty

Assume there is a person class as follows for the first two questions:

```
class Person{
        int age;
        String name;
        long height;

        // Getters and Setters of all the fields
        Public int getAge()
                Return age;
        Public void setAge(int age)
          This.age = age;


        Public String getName(){
                Return name;
        }
        Public voidsetName(String name)
          This.name = name;

        Public long getHeight()
                Return age;
        Public void setHeight(long height)
          This.height = height;



}
```

Q1. You are given a Binary Tree where a node of the tree looks like this

```
        class Node{
                Person person;
                Node next;
        }
```

You have to implement a function where you are given the root node, you have to return a hashmap in which the key will be the person object and the value would be no of occurrences of that person in the entire tree

public HashMap<Person, Integer> findPersonMap(Node root){

    // Write your code here: Tree traversal

}

Q2. You are given an array of Persons. You have to do the following operation until you are left with 1 element in the array. In the end, you have to return the minimum sum of all the operation values

The operation to do → Pick any two Person from the array, calculate the sum of their heights and then delete that person and add one person with the newly added height
The sum of two persons' heights is the operation value
For example, let's assume we have heights of people as :
1, 2, 3, 4

Way 1:

Pick 2, 3 Operation Value - 5, arr becomes - {1, 4, 5}
Pick 4, 5, Operation Value - 9, arr becomes - {1, 9}
Pick 1, 9 - Operation Value - 10, arr becomes {10}

Ans = 5 + 9 + 10 = 24

Way 2:
Pick 1, 2 Operation Value - 3, arr becomes - {3, 3, 4}
Pick 3, 3 Operation Value - 6, arr becomes - {4, 6}
Pick 4, 6 Operation Value - 10, arr becomes - {10}

Ans = 3 + 6 + 10 = 19

19 is the minimum operation sum value that we can achieve so for this example, we should 19

Note :

1. In every step, you are removing two-person and adding one person, so definitely at one point in time, your array will be going to have only 1 element. You just have to minimize the height of that last person
2. Also, you don't have to actually remove and add to/from the array. The function just expects an integer to be returned

public int findMinimumOperationSum(List<Person> personList){
    // write your code here
}

Q3. Implement a library management system. It’s an open-ended question, you can implement whatever comes to your mind. Try to incorporate the following features in your code.

1. Use OOPS concepts as much as you can
2. Make your code more modular and readable

3.