Laboratory 6: CSIT121
SN: 6421490
Sean Overton

Question 1 and Question 2::

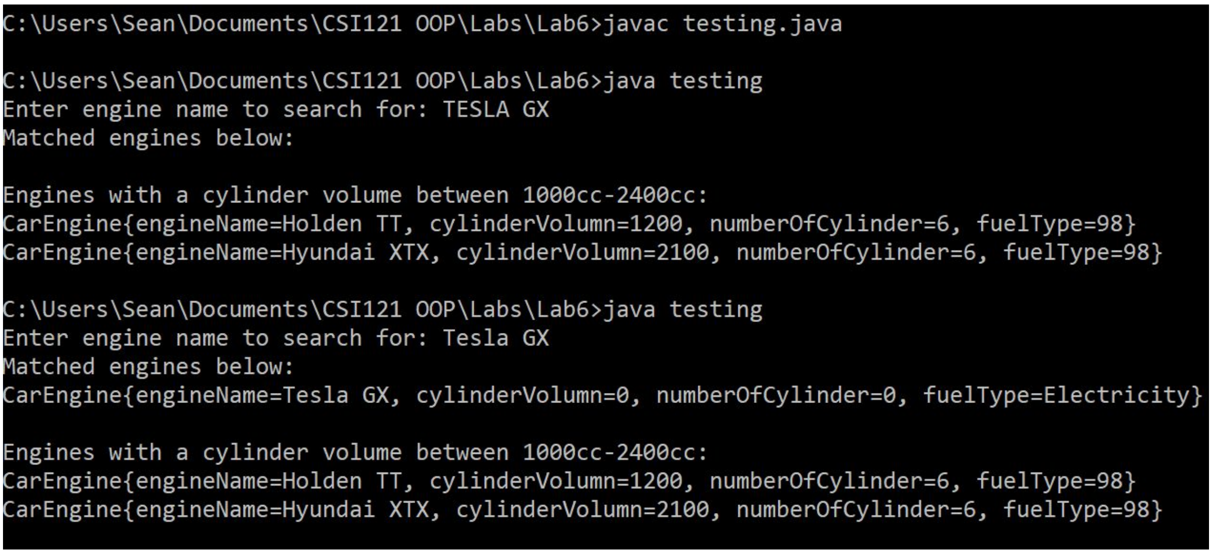
```
import java.util.Arrays;
import java.util.stream.IntStream;
import java.util.stream.Stream;
import java.util.stream.Collectors;
import java.util.TreeMap;

//both of these need checking
class testingStreams{
    public static void main(String[] args){
        //question 1
        int[] values1={12,2,9,8,4,65,7,4,2,66,88,11,33,44,55};

        //method 1
        System.out.printf("Question 1- Average of array: %.2f\n",
            IntStream.of(values1).average().getAsDouble());

        //question 2
        Integer[] values2 = {12,2,9,8,4,65,7,4,2,66,88,11,33,44,55};

        System.out.println("Question 2 - Output:");
        Arrays.stream(values2)
            .collect(Collectors.toSet())
            .forEach(e -> System.out.println(e));
    }
}
```

Question 3:

```
import java.util.Scanner;
import java.util.HashMap;
import java.util.Map;

class CarEngine {
    private String engineName;
    private int cylinderVolumn; // example 2.4L
    private int numberOfCylinder;
    private String fuelType;

    public CarEngine(String engineName, int cylinderVolumn, int numberOfCylinder, String fuelType) {
        this.engineName = engineName;
        this.cylinderVolumn = cylinderVolumn;
        this.numberOfCylinder = numberOfCylinder;
        this.fuelType = fuelType;
    }

    public String getEngineName() {
        return engineName;
    }

    public void setEngineName(String engineName) {
```

```java
        this.engineName = engineName;
    }

    public int getCylinderVolumn() {
        return cylinderVolumn;
    }

    public void setCylinderVolumn(int cylinderVolumn) {
        this.cylinderVolumn = cylinderVolumn;
    }

    public int getNumberOfCylinder() {
        return numberOfCylinder;
    }

    public void setNumberOfCylinder(int numberOfCylinder) {
        this.numberOfCylinder = numberOfCylinder;
    }

    public String getFuelType() {
        return fuelType;
    }

    public void setFuelType(String fuelType) {
        this.fuelType = fuelType;
    }

    @Override
    public String toString() {
        return "CarEngine{" + "engineName=" + engineName + ", cylinderVolumn=" + cylinderVolumn + ", 
numberOfCylinder=" + numberOfCylinder + ", fuelType=" + fuelType + '}';
    }
}

class testing {
    public static void main(String[] args){
        //creates 5 car engine objects
        CarEngine a0 = new CarEngine("Honda GT",2500,6,"98");
        CarEngine a1 = new CarEngine("Hyundai XTX",2100,6,"98");
        CarEngine a2 = new CarEngine("Holden TT",1200,6,"98");
        CarEngine a3 = new CarEngine("Toyota TX",900,6,"98");
        CarEngine a4 = new CarEngine("Tesla GX",0,0,"Electricity");

        //creates hash map and stores all car engine objects
        Map<String,CarEngine> ma = new HashMap<>();
        ma.put(a0.getEngineName(),a0);
        ma.put(a1.getEngineName(),a1);
        ma.put(a2.getEngineName(),a2);
        ma.put(a3.getEngineName(),a3);
        ma.put(a4.getEngineName(),a4);

        // ma.values().stream() convert a hashmap to a stream.

//      Write the code to search the object by the engine name (using Lambda and Stream)
        System.out.print("Enter engine name to search for: ");
        Scanner input = new Scanner(System.in);
        String engineStr = input.nextLine();
        //loop through hashmap keys and find a match then getEngine instance and print
        System.out.println("Matched engines below: ");
        ma.values().stream().filter(x ->
x.getEngineName().equals(engineStr)).forEach(System.out::println);

//        Write the code to search for the engine that the cylinder volume between 1000 cc. to 2400 cc.
and
//       display them to the console. (using Lambda and Stream)
        System.out.println();
        System.out.println("Engines with a cylinder volume between 1000cc-2400cc: ");
        ma.values().stream().filter(x -> x.getCylinderVolumn() > 1000).filter(x ->
x.getCylinderVolumn() < 2400).forEach(System.out::println);
    }
}
```

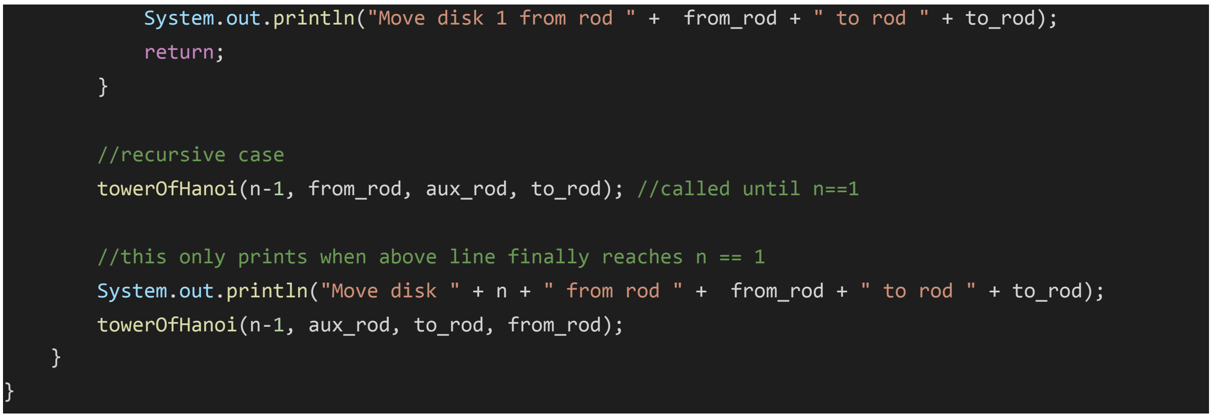
Compilation and testing:



```
C:\Users\Sean\Documents\CSI121 OOP\Labs\Lab6>javac testing.java

C:\Users\Sean\Documents\CSI121 OOP\Labs\Lab6>java testing
Enter engine name to search for: TESLA GX
Matched engines below:

Engines with a cylinder volume between 1000cc-2400cc:
CarEngine{engineName=Holden TT, cylinderVolumn=1200, numberOfCylinder=6, fuelType=98}
CarEngine{engineName=Hyundai XTX, cylinderVolumn=2100, numberOfCylinder=6, fuelType=98}

C:\Users\Sean\Documents\CSI121 OOP\Labs\Lab6>java testing
Enter engine name to search for: Tesla GX
Matched engines below:
CarEngine{engineName=Tesla GX, cylinderVolumn=0, numberOfCylinder=0, fuelType=Electricity}

Engines with a cylinder volume between 1000cc-2400cc:
CarEngine{engineName=Holden TT, cylinderVolumn=1200, numberOfCylinder=6, fuelType=98}
CarEngine{engineName=Hyundai XTX, cylinderVolumn=2100, numberOfCylinder=6, fuelType=98}
```

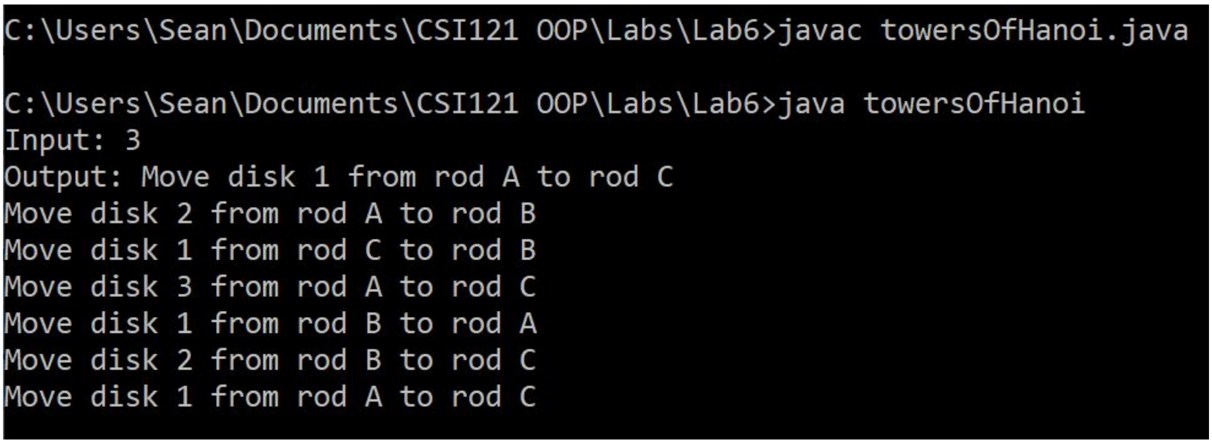
Question 4:

```
import java.util.ArrayList;
import java.util.Scanner;

class towersOfHanoi{
    public static void main(String[] args){
        //recursion must be used
        Scanner input = new Scanner(System.in);
        System.out.print("Input: ");
        int n = input.nextInt();// Number of disks

        System.out.print("Output: ");
        towerOfHanoi(n, "A", "C", "B");

    }

    //the recursive method used to solve problem
    public static void towerOfHanoi(int n, String from_rod, String to_rod, String aux_rod){
        //base case
        if (n == 1) {
```



```
            System.out.println("Move disk 1 from rod " +  from_rod + " to rod " + to_rod);
            return;
        }

        //recursive case
        towerOfHanoi(n-1, from_rod, aux_rod, to_rod); //called until n==1

        //this only prints when above line finally reaches n == 1
        System.out.println("Move disk " + n + " from rod " +  from_rod + " to rod " + to_rod);
        towerOfHanoi(n-1, aux_rod, to_rod, from_rod);
    }
}
```

Compilation and testing:



```
C:\Users\Sean\Documents\CSI121 OOP\Labs\Lab6>javac towersOfHanoi.java

C:\Users\Sean\Documents\CSI121 OOP\Labs\Lab6>java towersOfHanoi
Input: 3
Output: Move disk 1 from rod A to rod C
Move disk 2 from rod A to rod B
Move disk 1 from rod C to rod B
Move disk 3 from rod A to rod C
Move disk 1 from rod B to rod A
Move disk 2 from rod B to rod C
Move disk 1 from rod A to rod C
```