Java Coding Questions

Question 1 : Write a Java program that reads the size of an array from the user and performs basic operations on it.
Description: Take input for array size and elements. Then find the original array, minimum value, maximum value, sum, and average.
Example:
Input: 5
Array Elements: 3 7 1 9 4
Output:
Original Array: [3, 7, 1, 9, 4] Min: 1 Max: 9 Sum: 24 Average: 4.8

Answer)

```
import java.util.Scanner;

public class Program1 {

public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        System.out.println("Enter Array size : ");

        int size = sc.nextInt();

        int [] arr = new int[size];

        for(int i=0;i<size;i++) {

                System.out.println("Enter Array element for the position of "+i+" : ");

                arr[i]= sc.nextInt();

        }

        m1(arr);

        sc.close();

}

public static void m1(int [] arr) {

        int min=arr[0],max=arr[0],sum=0;

        double average=0;

        for(int i:arr ) {

        if(i<min) {

                min = i;

        }

        if(i>max) {
```

```
                max = i;
            }
                sum+=i;
                System.out.print(i+" ");
            }
            average = (double)sum/arr.length;
            System.out.println("\nMin : "+min);
            System.out.println("Max : "+max);
            System.out.println("Sum : "+sum);
            System.out.println("Average : "+average);

        }

        }
```

Question 2 : Print a Pyramid Pattern with Top and Bottom

Description: Write a Java program that takes an integer input n and prints a symmetric pyramid pattern where the pyramid has both a top and a bottom part. The pattern should be made using * characters and have a height of 2 * n - 1 lines. Inputs are provided dynamically by the user during program execution.

Example:

Input: n = 4

Output:

```
   *
  ***
 *****
*******
 *****
  ***
   *
```

Answer)

```
import java.util.Scanner;

public class Program2 {
```

```
public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter how many rows you want ");
        int rows = sc.nextInt();
        m1(rows);
}
public static void m1(int rows) {
        // Ascending Order
        for(int i=1;i<= rows;i++) {
                for(int j=i;j<rows;j++) {
                        System.out.print(" ");
                }
                for(int k=1;k<=(2*i-1);k++) {
                        System.out.print("*");
                }
                System.out.println();
        }
        // Descending Order
        for(int i=rows-1;i>=1;i--) {
                for(int j=i;j<rows;j++) {
                        System.out.print(" ");
                }
                for(int k=1;k<=(2*i-1);k++) {
                        System.out.print("*");
                }
                System.out.println();
        }
}
}
```

Question 3 : Write a program to traverse a list and print all elements.
Description: Use both for-each loop and iterator to demonstrate list traversal. Inputs are provided dynamically by the user during program execution.
Example: Input List: [10, 20, 30, 40]
Output: 10 20 30 40

Answer)

```
import java.util.ArrayList;
import java.util.Collections;
import java.util.Iterator;
import java.util.List;
import java.util.Scanner;


public class Program3 {
public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter Array size : ");
        int size = sc.nextInt();
        List<Integer> li = new ArrayList<Integer>();
        for(int i=0;i<size;i++) {
                System.out.println("Enter Array element for the position of "+i+" : ");
                int n= sc.nextInt();
                li.add(n);
        }
        m1(li);
        sc.close();
}
public static void m1(List<Integer> arr) {
         Collections.sort(arr);


         for(int i:arr) {
                  System.out.print(i+" ");
         }
         System.out.println();
```

```
        Iterator<Integer> it = arr.iterator();
        while(it.hasNext())
        {
                System.out.print(it.next()+" ");
        }

}
                        }
```

Question 4 : Rotate Digits to Form Maximum Number
Description: Given an integer, rotate its digits and return the maximum number that can be formed. Inputs are provided dynamically by the user during program execution.
Example:
Input: 3142
Output: 4321

Answer)

```
import java.util.ArrayList;
import java.util.Collections;
import java.util.Iterator;
import java.util.List;
import java.util.Scanner;

public class Program4 {
public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter Array size : ");
        int size = sc.nextInt();
        List<Integer> li = new ArrayList<Integer>();
        for(int i=0;i<size;i++) {
                System.out.println("Enter Array element for the position of "+i+" : ");
                int n= sc.nextInt();
                li.add(n);
        }
        m1(li);
```

```
        sc.close();


}
public static void m1(List<Integer> arr) {
        Collections.sort(arr,Collections.reverseOrder());


        for(int i:arr) {
                System.out.print(i+" ");
        }
        System.out.println();
        Iterator<Integer> it = arr.iterator();
        while(it.hasNext())
        {
                System.out.print(it.next()+" ");
        }


}
}
```

Question 5 : Convert a Roman numeral to an integer.
Description: Use character comparison logic to convert Roman to decimal. Inputs are provided dynamically by the user during program execution.
Example:
Input: "IX"
Output: 9

Answer)

```
import java.util.HashMap;
public class Program5 {


public static void main(String[] args) {
        String s ="LVIII";
        HashMap<Character,Integer> roamn = new HashMap<Character,Integer>();
                roamn.put('I', 1);
```

```
        roamn.put('V', 5);
        roamn.put('X', 10);
        roamn.put('L', 50);
        roamn.put('C', 100);
        roamn.put('D', 500);
        roamn.put('M', 1000);

        // String[] sarr = s.split("");
    int prevValue =0;
        int sum = 0;

        for (int i=s.length()-1;i >=0;i--) {
      int currentvue = roamn.get(s.charAt(i));
            if(currentvue < prevValue)
            {
                sum-= currentvue;
            }else{
       sum+= currentvue;
      }
      prevValue = currentvue;
        }
        System.out.println(sum);
}
}
```

Question 6: Demonstrate Object-Oriented Programming concepts with a simple example.
Description: Cover all topics/features of OOPs (Abstraction, encapsulation, interface, Multiple-Inheritance and polymorphism.) Inputs are provided dynamically by the user during program execution.

Answer)

```
import java.util.Scanner;
abstract class Vehicle {
   String brand;
```

```
    public Vehicle(String brand) {
        this.brand = brand;
    }

    public abstract void start();
}


interface Electric {
    void charge();
}


class Car extends Vehicle implements Electric {
    private String model;
    private int year;

    public Car(String brand, String model, int year) {
        super(brand);
        this.model = model;
        this.year = year;
    }


    public String getModel() {
        return model;
    }

    public void setModel(String model) {
        this.model = model;
    }
```

```
    public int getYear() {
        return year;
    }

    public void setYear(int year) {
        this.year = year;
    }


    public void start() {
        System.out.println(brand + " " + model + " is starting...");
    }


    public void charge() {
        System.out.println(brand + " " + model + " is charging...");
    }
}


class Bike extends Vehicle {
    public Bike(String brand) {
        super(brand);
    }

    public void start() {
        System.out.println(brand + " bike is starting...");
    }
}
```

```
public class Program6 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.println("Enter Car Brand:");
        String carBrand = sc.nextLine();
        System.out.println("Enter Car Model:");
        String carModel = sc.nextLine();
        System.out.println("Enter Car Year:");
        int carYear = sc.nextInt();
        sc.nextLine();
        System.out.println("Enter Bike Brand:");
        String bikeBrand = sc.nextLine();


        Vehicle car = new Car(carBrand, carModel, carYear);
        Vehicle bike = new Bike(bikeBrand);

        System.out.println("\n--- Car Details ---");
        car.start();
        ((Car) car).charge();

        System.out.println("\n--- Bike Details ---");
        bike.start();

        sc.close();
    }
}
```

Question 7 : Write a program to count frequency of each element in an array.
Description: Use a map to store the frequency of each number. Inputs are provided dynamically by the user during program execution.

Example:
Input: [1, 2, 2, 3, 1, 4]
Output: 1: 2
2: 2
3: 1
4: 1
Answer)

```
import java.util.ArrayList;
import java.util.Iterator;
import java.util.List;
import java.util.Map;
import java.util.Scanner;
import java.util.TreeMap;

public class Program7 {
public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter Array size : ");
        int size = sc.nextInt();
        List<Integer> li = new ArrayList<Integer>();
        for(int i=0;i<size;i++) {
                System.out.println("Enter Array element for the position of "+i+" : ");
                int n= sc.nextInt();
                li.add(n);
        }
        m1(li);
        sc.close();
}
public static void m1(List<Integer> arr) {
TreeMap<Integer,Integer> tm = new TreeMap<Integer,Integer>();
        for(int i:arr) {
                int count =0;
```

```
                for(int j:arr) {
                        if( i == j) {
                                count++;
                        }
                }
                tm.put(i,count);
        }
        System.out.println();
        Iterator<Map.Entry<Integer, Integer>> it = tm.entrySet().iterator();
        while(it.hasNext())
        {
                Map.Entry<Integer,Integer> mp = it.next();
                System.out.println(mp.getKey()+": "+mp.getValue());
        }
}
}
```

Question 8 : Guess a number between 1 to 100. You have 5 chances.
Description: Generate a random number using Math.random() and let user guess it in max 5 tries. The user is given 5 chances to guess the number correctly. After each incorrect guess, provide a hint if the guess was **low** or **high**. If the user fails to guess in 5 attempts, the correct number should be revealed.
Example:
Random Number: 56
Guesses: 40, 60, 55, 56
Output: Correct! You guessed in 4 attempts.

Answer)

```
import java.util.*;


public class GuessNumber {
   public static void main(String[] args) {
      Scanner sc = new Scanner(System.in);
      int number = (int)(Math.random() * 100) + 1;
      int attempts = 5;
```

```
        while(attempts-- > 0) {
            System.out.print("Guess the number: ");
            int guess = sc.nextInt();
            if(guess == number) {
                System.out.println("Correct! You guessed it.");
                return;
            } else if(guess < number)
                System.out.println("Too low.");
            else
                System.out.println("Too high.");
        }
        System.out.println("Out of attempts! The number was: " + number);
    }
}
```

Question 9 : Group Anagrams Together Using Map
Description: Given a list of strings, group all anagrams together. Return a list of lists of grouped anagrams. Inputs are provided dynamically by the user during program execution.
Example:
Input: ["bat", "tab", "tap", "pat", "cat"]
Output: [["bat", "tab"], ["tap", "pat"], ["cat"]]

Answer)

```
import java.util.ArrayList;
import java.util.Arrays;
import java.util.List;
import java.util.Scanner;
import java.util.TreeMap;

public class Program9 {
public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
```

```
        System.out.println("Enter String Array size : ");
        int size = sc.nextInt();
        sc.nextLine();
        List<String> li = new ArrayList<String>();
        for(int i=0;i<size;i++) {
                System.out.println("Enter Array element for the position of "+i+" : ");
                String s= sc.nextLine();
                li.add(s);
        }
        m1(li);
        sc.close();
}

public static void m1(List<String> arr) {
         TreeMap<String, List<String>> tm = new TreeMap<>();

           for (String word : arr) {

              char[] carr = word.toCharArray();
              Arrays.sort(carr);
              String sortedWord = new String(carr);



              if (!tm.containsKey(sortedWord)) {
                 tm.put(sortedWord, new ArrayList<>());
              }
              tm.get(sortedWord).add(word);
           }



           System.out.println("Anagram Groups:"+tm);
```

```
}


}
```

Question 10 : Stream Pipeline Output with Dynamic Input

Description: Write a Java 8 program that takes dynamic string inputs from the user. The user will enter n strings. Use Stream API to:

Filter strings starting with the letter "a",
Sort them alphabetically,
Convert them to uppercase,
Collect and print the final list.

Example:

Input: Enter number of strings: 5
Enter strings: apple banana avocado cherry apricot
Output: [APPLE, APRICOT, AVOCADO]

Answer)

```
import java.util.Arrays;

import java.util.List;

import java.util.Scanner;

import java.util.stream.Collectors;


public class Program10 {

public static void main(String[] args) {

	Scanner sc = new Scanner(System.in);

	System.out.println("Enter the length of the Strings");

	int n = sc.nextInt();

	sc.nextLine();

	String [] sarr = new String[n];

	for(int i=0;i<n;i++) {

		System.out.println("Enter the Strings");

		sarr[i]=sc.nextLine();

	}

	m1(sarr);

	sc.close();

}
```

```
public static void m1(String [] sarr) {


List<String> li = Arrays.stream(sarr)
                                        .filter(s->s.startsWith("a")|| s.startsWith("A"))
                                        .sorted()
                                        .map(String::toUpperCase)
                                        .collect(Collectors.toList());
System.out.println(li);
}
}
```