Assignment

Part1: Explain in a few words how the Communication Between the Client & the Server over request-response takes place?

Ans:
Clients typically communicate with servers by using the TCP/IP protocol suite. TCP is a connection-oriented protocol, which means a connection is established and maintained until the application programs at each end have finished exchanging messages. It determines how to break application data into packets that networks can deliver, sends packets to and accepts packets from the network layer, manages flow control and handles retransmission of dropped or garbled packets as well as acknowledgement of all packets that arrive.

There are few steps to follow to interact with the servers and clients.

- User enters the URL(Uniform Resource Locator) of the website or file. The Browser then requests the DNS(DOMAIN NAME SYSTEM) Server.
- DNS Server lookup for the address of the WEB Server.
- DNS Server responds with the IP address of the WEB Server.
- Browser sends over an HTTP/HTTPS request to WEB Server's IP (provided by DNS server).
- Server sends over the necessary files of the website.
- Browser then renders the files and the website is displayed. This rendering is done with the help of DOM (Document Object Model) interpreter, CSS interpreter and JS Engine collectively known as the JIT or (Just in Time) Compilers.

For example: When a user requests a website the request goes to Domain server.The domain server gives the IP address of the requested web site and redirects the user to the dedicated server. The server established connection after server-side Authentication and process the request made by user and send the packet to user in the form of HTML, Css, JavaScript file. This file is parsed by the user browser engine and displays the content on the window of the computer or device.

The Open Systems Interconnection (OSI) model describes seven layers that computer systems use to communicate over a network. It was the first standard model for network communications, adopted by all major computer and telecommunication companies in the early 1980s. It is a 7 layer architecture with each layer having specific functionality to perform. All 7 layers work collaboratively to transmit the data from one person to another across the globe.

Part 2:
Given an array and a number "n", find two numbers from the array that sums up to "n".
Sample Input #
arr = {1, 21, 3, 14, 5, 60, 7, 6}
value = 27
Sample Output #
arr = {21, 6} or {6, 21}
Write an efficient solution in pseudo code or algorithm, also compute and give a
small explanation of the Big O complexity of the code snippet.

Ans:

```
package VintageCircleAssignment;
import java.util.Arrays;
public class Part2 {

  public static void main(String[] args) {
//       Given an array and a number "n", find two numbers from the array that sums up to "n".
     int arr[]={1, 21, 3, 14, 5, 60, 7, 6};
     Arrays.sort(arr);
     int value=27;
     int low=0;
     int high=arr.length-1;
     while (low < high){  —-------------------- N-1

        if (arr[low] +  arr[high] > value){  —-------1
           high--;
        } if (arr[low]+ arr[high] < value) {-------------1
           low++;
        }if (arr[low]+ arr[high]== value) {------------1
           System.out.println("Pair of the array that sums up to \"n\" (" + arr[low] + "," +
arr[high] + ")");
           low++;
           high--;

        }
     }
  }
}
```

Here the time complexity is O(N).First sorting the array.Once the array is sorted the two pointers can be taken which mark the beginning and end of the array respectively. Every time we are checking the low index is less than the higher index until the while loop stops. Each time adding two index values and shifting the position of the index and printing the value when the sum of two index values is equal to the value passed by the user.

Part 3:

AirBNB operating in Thiruvananthapuram wishes to offer Bed and Breakfast(B&B) services over
the internet. They have three B&B in Thiruvananthapuram: CoconutValley, AakulamLake and VeliBeach. Each B&B has separate weekday and weekend(Saturday and Sunday) rates. There
are special rates for rewards customers as a part of the loyalty program. Each hotel has a rating
assigned to it.
CoconutValley with a rating of 3 has weekday rates as Rs1100 for regular customers and Rs800 for rewards customers. The weekend rates are 900 for regular customers and 800 for a
rewards customer.
AakulamLake with a rating of 4 has weekday rates as Rs1600 for regular customers and Rs1100 for rewards customers. The weekend rates are 600 for regular customers and 500 for a
rewards customer.
VeliBeach with a rating of 5 has weekday rates as Rs2200 for regular customers and Rs1000
for rewards customers. The weekend rates are 1500 for regular customers and 400 for a rewards customer.
Can you write a program to help an online customer find the cheapest hotel?
The input to the program will be a range of dates for a regular or rewards customer. The output
should be the cheapest available hotel.
*In case of a tie, the hotel with the highest rating should be returned.

Ans:

1.Main Class

```
package VintageCircleAssignment.Part3;

import VintageCircleAssignment.Part3.ServiceClass.ServiceClass;


public class MainClass {
  public static void main(String[] args) throws Exception {

   Taking Input
     Scanner sc =new Scanner(System.in);
     System.out.println("Please Enter data in this format 'Rewards: 26Mar2009, 27Mar2009,
28Mar2009'");
     String input = sc.nextLine();

//      Creating Objects of Service Class.
     ServiceClass service=new ServiceClass();
//      Data is added to HashMap
     service.SetHotels();
```

```
        String hotel =service.UserInput(input);
        System.out.println(hotel);
    }
}
```

2.Model Class

```
package VintageCircleAssignment.Part3.ModelClass;

public class ModelClass {
    int Bed;
    int Breakfast;
    int Rating;
    int Regprice;
    int RewPrice;
    String Hotelname;
    String Regular;
    String Rewards;

    public ModelClass() {
    }

    public ModelClass(int bed, int breakfast, String hotelname) {
        Bed = bed;
        Breakfast = breakfast;
        Hotelname = hotelname;
    }

    public int getBed() {
        return Bed;
    }

    public void setBed(int bed) {
        Bed = bed;
    }

    public int getBreakfast() {
        return Breakfast;
    }

    public void setBreakfast(int breakfast) {
        Breakfast = breakfast;
    }

    public int getRating() {
        return Rating;
    }

    public void setRating(int rating) {
        Rating = rating;
    }

    public int getRegprice() {
        return Regprice;
    }

    public void setRegprice(int regprice) {
        Regprice = regprice;
    }

    public int getRewPrice() {
        return RewPrice;
    }

    public void setRewPrice(int rewPrice) {
        RewPrice = rewPrice;
    }

    public String getHotelname() {
        return Hotelname;
    }

    public void setHotelname(String hotelname) {
        Hotelname = hotelname;
    }

    public String getRegular() {
        return Regular;
    }

    public void setRegular(String regular) {
        Regular = regular;
    }

    public String getRewards() {
        return Rewards;
    }

    public void setRewards(String rewards) {
        Rewards = rewards;
    }

}
```

3.Service Class

```
package VintageCircleAssignment.Part3.ServiceClass;

import VintageCircleAssignment.Part3.ModelClass.ModelClass;


import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Date;
import java.util.HashMap;
import java.util.List;

public class ServiceClass {

  ModelClass modelClass=new ModelClass();

  public HashMap<String, List<Integer>> CoconutValley = new HashMap<>();

  public HashMap<String, List<Integer>> AakulamLake = new HashMap<>();
  public  HashMap<String, List<Integer>> Velibeach = new HashMap<>();


  private static int CocountValleyRating;
  private static int AakulamLekeRating;
  private static int VelibeachRating;


  public void SetHotels() {
     setCoconutValley();
     setAakulamLake();
     setVelibeach();
  }

  public void setCoconutValley() {
     List<Integer> CocountValleyRewords = new ArrayList<>();
     CocountValleyRewords.add(800);
     CocountValleyRewords.add(800);

     List<Integer> CocountValleyRegugar = new ArrayList<>();
     CocountValleyRegugar.add(1100);
     CocountValleyRegugar.add(900);


     CoconutValley.put("Rewards", CocountValleyRewords);
     CoconutValley.put("Regular", CocountValleyRegugar);
     CocountValleyRating = 3;
```

```
}


public void setAakulamLake() {


    List<Integer> AuakalamLakeRewords = new ArrayList<>();
    AuakalamLakeRewords.add(1100);
    AuakalamLakeRewords.add(500);

    List<Integer> AuakalamRewgular = new ArrayList<>();
    AuakalamRewgular.add(1600);
    AuakalamRewgular.add(600);

    AakulamLake.put("Rewards", AuakalamLakeRewords);
    AakulamLake.put("Regular", AuakalamRewgular);
    AakulamLekeRating = 4;

}

public void setVelibeach() {

    List<Integer> VelibeachRewaords = new ArrayList<>();
    VelibeachRewaords.add(1000);
    VelibeachRewaords.add(400);

    List<Integer> VelibeachRewgular = new ArrayList<>();
    VelibeachRewgular.add(2200);
    VelibeachRewgular.add(1500);

    Velibeach.put("Rewards", VelibeachRewaords);
    Velibeach.put("Regular", VelibeachRewgular);
    VelibeachRating=5;



}


public String UserInput(String input) throws Exception {
    // Regular: 16Mar2009(mon), 17Mar2009(tue), 18Mar2009(wed)
    String arr[] = input.split(",");
    String CustomerType = "";
    if (arr.length > 0) {
        String tempArray[] = arr[0].split(":");
        CustomerType = tempArray[0];
```

```
         arr[0] = tempArray[1];
      }
      String Hotels = minCostHotels(CustomerType, arr);
      return Hotels;

   }



   private String minCostHotels(String customerType, String[] arr)throws Exception{
      int numweekdays = 0;
      int numweekends = 0;
      int CocountValleyCost=0;
      int AakulamLakeCost=0;
      int VelibeachCost=0;

      for (int i = 0; i < arr.length; i++) {
         String dates = arr[i];

         SimpleDateFormat formatter2=new SimpleDateFormat("ddMMMyyyy");
         Date date2=formatter2.parse(dates);

         System.out.println(dates);
//          System.out.println("\t"+date2);

         String dayName=date2.toString();

         String arraysofday[]= dayName.split(" ");

         for (int j = 0; j <1; j++) {

            if (arraysofday[0].equalsIgnoreCase("Sat")
||arraysofday[0].equalsIgnoreCase("Sun")){
               numweekends++;
            }else {
               numweekdays++;
            }
         }
      }

      List<Integer> valueofCocountVelly = CoconutValley.get(customerType);
      List<Integer> valueofAakulamlake = AakulamLake.get(customerType);
      List<Integer> valuesofVeliBeach = Velibeach.get(customerType);
      try {
         CocountValleyCost = numweekdays *  valueofCocountVelly.get(0) + numweekends *
valueofCocountVelly.get(1);
         AakulamLakeCost = numweekdays * valueofAakulamlake.get(0) + numweekends *
valueofAakulamlake.get(1);
```

```
        VelibeachCost = numweekdays * valuesofVeliBeach.get(0) + numweekends *
valuesofVeliBeach.get(1);
      }catch (Exception e){
        System.out.println(e);
      }

      String hotel = minCost(CocountValleyCost, AakulamLakeCost, VelibeachCost);

      System.out.println( "CocountvalleyPrice:"+CocountValleyCost+"\n" +
"AakulamLakePrice:" + AakulamLakeCost +"\n"+ "VelibeachPrice:" + VelibeachCost);

      return hotel;
  }


  private String minCost(int cocountValleyCost, int aakulamLakeCost, int velibeachCost) {

      int minCost = Math.min(cocountValleyCost, Math.min(aakulamLakeCost,
velibeachCost));

      if (minCost == cocountValleyCost && minCost == aakulamLakeCost) {

        return aakulamLakeCost > cocountValleyCost ? "AakulamLake" : "cocountValley";
      }
      else if (minCost == cocountValleyCost && minCost == velibeachCost) {

        return cocountValleyCost > velibeachCost ? "CocountValley" : "VeliBeach";
      }
      else if (minCost == aakulamLakeCost && minCost == velibeachCost) {

        return aakulamLakeCost > velibeachCost ? "AakulamLake" : "Velibeach";
      }
      else {

        if (minCost == cocountValleyCost) {
          int rating = CocountValleyRating;
          System.out.println(rating+"(rating)");

          return "CoconutValley is Cheapest" ;


        } else if (minCost == aakulamLakeCost) {

          int rating =AakulamLekeRating;
          System.out.println(rating+"(rating)");
          return "AakulamLake is Cheapest";

        } else {
            int rating = VelibeachRating;
            System.out.println(rating+"(rating)");
            return "VeliBeachCost is Cheapest";
        }
    }


  }

}
```

Submitted By:
Vishal Kumar Singh
Vishalsingh961266@gmail.com