Cawangan Melaka
Kampus Jasin

CLASS:
M3CS1103C

COURSE:
FUNDAMENTALS OF DATA STRUCTURE (CSC248)

PROJECT NAME:
THE FUTURE OF HOTEL RESERVATION SYSTEM

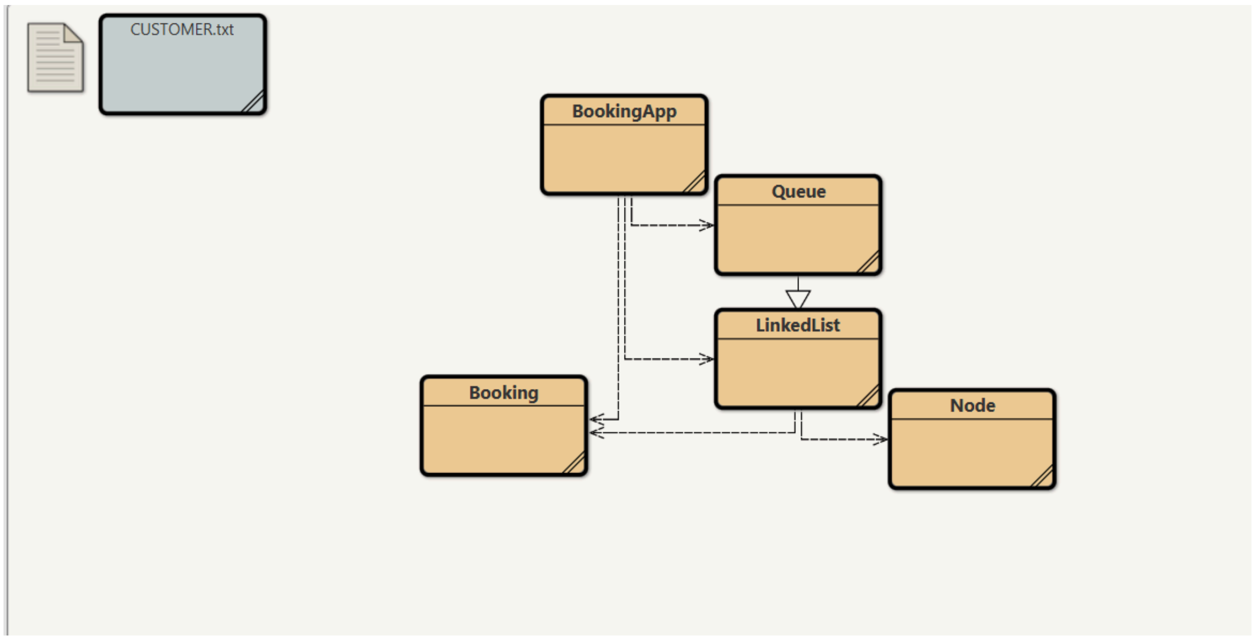
LECTURER:
ROHANA BINTI RAMLI

GROUP MEMBERS:

STUDENT NAME	STUDENT ID
MUHAMMAD IRFAN SYAFIQ BIN ZAIDI	2022488964
MUHAMMAD AMIR BIN KHAIRUL NIZAR	2022832564

TABLE OF CONTENTS

1.0 RELATIONSHIP



2.0 INPUT DATA FILE

```
801223100615;0169517565;single;145;1;paid
980812040352;0176457843;king;155;2;pending
991101090124;0199351428;king;190;3;paid
860129070293;0189815444;single;240;2;paid
720311080710;0155517815;single;300;4;pending
940923060111;0123216845;single;372;3;paid
810512070258;0141891254;king;389;1;pending
961010090541;0118184175;single;439;2;paid
950221060432;0131578184;king;467;2;paid
980808080619;0171871527;single;573;1;pending
010717050303;0186541582;king;610;4;pending
990418090486;0199638257;king;683;5;pending
851210040545;0141482658;single;699;2;paid
930730030637;0162519235;single;710;1;pending
880802040206;0149874982;single;760;3;pending
890225020456;0125547441;king;860;1;pending
821128030801;0115465186;single;908;3;pending
980102010112;0199156165;single;926;4;paid
900622020678;0199871554;king;964;1;paid
010926100351;0163345847;king;987;2;pending
```

3.0 CLASS NODE

```
public class Node
{
   public Object data;
   public Node next;

   public Node (Object d)
   {
      data = d;
   }
}
```

4.0 CLASS LINKED LIST

```
import javax.swing.*;

public class LinkedList
{
   private Node first;
   private Node current;
   private Node last;

   public LinkedList()
   {
      first = null;
      last = null;
      current = null;
   }

   //check linkedList is empty or not
   public boolean isEmpty()
   {
      return (first == null);
   }

   //insert something from the front of linkedList
   public void insertAtFront(Object insertItem)
   {
      Node newNode = new Node(insertItem);

      if (isEmpty())
      {
         first = newNode;
         last = newNode;
      }
      else
      {
         newNode.next = first;
         first = newNode;
      }
   }

   //insert something from the back of linkedList
```

```
public void insertAtBack(Object insertItem)
{
    Node newNode = new Node(insertItem);

    if(isEmpty())
    {
        first = newNode;
        last = newNode;
    }
    else
    {
        last.next = newNode;
        last = newNode;
    }
}

//remove something from the front of linkedList
public Object removeFromFront()
{
    Object removeItem = null;
    if (isEmpty())
    {
        return removeItem;
    }

    removeItem = first.data;
    if (first == last)
    {
        first = null;
        last = null;
    }
    else
    {
        first = first.next;
    }

    return removeItem;
}

//remove something from the back of linkedList
public Object removeFromBack()
{
    Object removeItem = null;
    if (isEmpty())
    {
        return removeItem;
    }

    removeItem = last.data;

    if (first == last)
    {
        first = null;
        last = null;
    }
    else
    {
        current = first;
        while (current.next != last)
        {
            current = current.next;
        }
        last = current;
        last.next = null;
    }

    return removeItem;
}

//retrieve the first thing in the linkedList
public Object getFirst()
{
    if (isEmpty())
    {
        return null;
    }
    else
    {
        current = first;
        return current.data;
    }
}

//retrive the next thing in the linkedList
public Object getNext()
{
    if (current == last)
    {
        return null;
    }
    else
    {
        current = current.next;
        return current.data;
    }
}

//determine the linkedList size
public int length()
{
    int length = 0;
    current = first;

    while(current != null)
    {
        length++;
        current = current.next;
    }

    return length;
}

//remove the node specified by user (roomNumber)
public void remove(int roomNumber)
{
    current = first;
    Node previous = null;

    while (current != null)
    {
        if (current.data instanceof Booking)
        {
            Booking booking = (Booking) current.data;
            if (booking.getRoomNum() == roomNumber)
            {
                if (previous == null)
                {
                    // If the node to be removed is the first node
                    first = current.next;
                }
                else
```

```
                    {
                        // If the node to be removed is not the first node
                        previous.next = current.next;
                        if (current.next == null)
                        {
                            // If the node to be removed is the last node
                            last = previous;
                        }
                    }
                    System.out.println("Booking with room number " + roomNumber + " removed."); //noting
that the roomNumber choosen was removed
                    return; // Exit the method after removing
                }
            }
            previous = current;
            current = current.next;
        }
        // If the room number is not found
        System.out.println("Room number " + roomNumber + " not found.");
    }

    //update the objects data
    public void update(int roomNumber)
    {
        current = first;

        String roomType , reservationStatus;
        int nightsStayed;

        while (current != null)
        {
            if (current.data instanceof Booking)
            {
                Booking booking = (Booking) current.data;
                if (booking.getRoomNum() == roomNumber)
                {
                    // Update the attributes

                    //updates the room type
                    while(true)
                    {
                        roomType = (JOptionPane.showInputDialog("Enter room type (SINGLE/KING):"));

                        if("SINGLE".equalsIgnoreCase(roomType) || "KING".equalsIgnoreCase(roomType))
                        {
                            booking.setRoomType(roomType);
                            break;
                        }
                        else
                        {
                            JOptionPane.showMessageDialog(null,"INVALID INPUT!");
                        }
                    }

                    //updates the nights stayed
                    while(true)
                    {
                        try
                        {
                            nightsStayed = (Integer.parseInt(JOptionPane.showInputDialog("Enter nights
stayed:")));

                            if(nightsStayed >0)
                            {
                                booking.setNightsStayed(nightsStayed);
                                break;
                            }
                            else
                            {
                                JOptionPane.showMessageDialog(null,"INVALID INPUT!");
                            }
                        }
                        catch(NumberFormatException e) //catch user input error, input not an int
                        {
                            JOptionPane.showMessageDialog(null,"INVALID INPUT!");
                        }
                    }

                    //updates the reservation status
                    while(true)
                    {
                        reservationStatus = (JOptionPane.showInputDialog("Enter reservation status
(PAID/PENDING):"));

                        if("PAID".equalsIgnoreCase(reservationStatus) ||
"PENDING".equalsIgnoreCase(reservationStatus))
                        {
                            booking.setReservationStatus(reservationStatus);
                            break;
                        }
                        else
                        {
                            JOptionPane.showMessageDialog(null,"INVALID INPUT!");
                        }
                    }

                    System.out.println("Booking with room number " + roomNumber + " updated."); //noting
that the object was updated
                    return; // Exit the method after updating
                }
            }
            current = current.next;
        }

        // If the room number is not found
        System.out.println("Room number " + roomNumber + " not found.");
      }
}
```

5.0 CLASS QUEUE

```
public class Queue extends LinkedList
{
   public Queue()
   {}

   //enqueue something in the queue
   public void enqueue(Object elem)
   {
      insertAtBack(elem);
   }

   //deque something in the queue
   public Object dequeue()
   {
      return removeFromFront();
   }

   //gets the first thing in the queue
   public Object getFront()
   {
      return getFirst();
   }

   //gets the last thing in the queue
   public Object getEnd()
   {
      Object obj = removeFromFront();
      insertAtBack(obj); // reinsert
      return obj;
   }
}
```

6.0 CLASS BOOKING

```
//import java.text.DecimalFormat;

public class Booking
{
   // attributes
   //private double singlePrice = 85.50 , kingPrice = 125.75;
   private int roomNum , nightsStayed;
   private  String guestIC , guestContactNum , roomType , reservationStatus;

   //DecimalFormat df = new DecimalFormat("0.00");

   // normal constructor
   public Booking(String guestIC , String guestContactNum , String roomType , int roomNum , int
nightsStayed , String reservationStatus)
   {
      this.guestIC = guestIC;
      this.guestContactNum = guestContactNum;
      this.roomType = roomType;
      this.roomNum = roomNum;
      this.nightsStayed = nightsStayed;
      this.reservationStatus = reservationStatus;
   }

   // setter
   public void setBooking(String guestIC , String guestContactNum , String roomType , int roomNum ,
int nightsStayed , String reservationStatus)
   {
      this.guestIC = guestIC;
      this.guestContactNum = guestContactNum;
      this.roomType = roomType;
      this.roomNum = roomNum;
      this.nightsStayed = nightsStayed;
      this.reservationStatus = reservationStatus;
   }

   public void setRoomType(String roomType)
   {
      this.roomType = roomType;
   }
```

```
    public void setNightsStayed(int nightsStayed)
    {
        this.nightsStayed = nightsStayed;
    }

    public void setReservationStatus(String reservationStatus)
    {
        this.reservationStatus = reservationStatus;
    }

    // getter
    public String getGuestIC()
    {   return guestIC;}

    public String getGuestContactNum()
    {   return guestContactNum;}

    public String getRoomType()
    {   return roomType;}

    public int getRoomNum()
    {   return roomNum;}

    public int getNightsStayed()
    {   return nightsStayed;}

    public String getReservationStatus()
    {   return reservationStatus;}

    public String reportPerCust()
    {
        return (String.format("|%-13s|%-18s|%-10s|%-9s|%-14s|%-19s|", guestIC , guestContactNum , 
roomType , roomNum , nightsStayed , reservationStatus));
    }
}
```

7.0 CLASS BOOKING APP

```
import java.util.*; //adt
import java.io.*; //input output
import javax.swing.*; //joptionpane

public class BookingApp
{
   public static void main(String args[]) throws Exception
   {
      File file = new File("C:\\Users\\skkrrrtttt\\OneDrive\\Documents\\# WORKS\\CSC 248\\FINAL
PROJECT\\FILE INPUT\\CUSTOMER.txt"); //input file and file destination
      Scanner scanFile = new Scanner(file); //scan the file

      //ADT
      LinkedList linkedList = new LinkedList(); //linked list for booking
      Queue queuePaid = new Queue();          //queue for paid customer
      Queue queuePending = new Queue();        //queue for pending customer

      //LINKEDLIST
      while(scanFile.hasNext()) //loops until hasNext is null to break out of loop
      {
         //scan file
         String indata = scanFile.nextLine();
         StringTokenizer st = new StringTokenizer(indata, ";");

         //temporary attribute to store in object b, to pass in arguments in Booking normal constructor
         String guestIC = st.nextToken();
         String guestContactNum = st.nextToken();
         String roomType = st.nextToken();
         int roomNum = Integer.parseInt(st.nextToken());
         int nightsStayed = Integer.parseInt(st.nextToken());
         String reservationStatus = st.nextToken();

         Booking b = new Booking (guestIC , guestContactNum , roomType , roomNum , nightsStayed ,
reservationStatus); //arguments pass in Booking normal constructor

         //Question i) & ii)
         //insert from front or back
```

```
          //determine object b reservation status, either "PAID" or "PENDING"
          if ("PAID".equalsIgnoreCase(b.getReservationStatus()))
          {
             linkedList.insertAtFront(b); //object b is inserted from the front of linked list
          }
          else if ("PENDING".equalsIgnoreCase(b.getReservationStatus()))
          {
             linkedList.insertAtBack(b); //object b is inserted from the back of linked list
          }
       }

       scanFile.close(); //close scanFile

       //attributes for conditions
       String userRespons = "YES";
       String remUpdIns = "REMOVE"; // or "UPDATE"
       int roomNum;

       while(userRespons.equalsIgnoreCase("YES"))
       {
          //display table BOOKING LIST
          System.out.println("|======================================BOOKING
LIST========================================|\n");
          System.out.println(String.format("|%-13s|%-18s|%-10s|%-9s|%-14s|%-19s|","GUEST
IC","GUEST CONTACT NUM","ROOM TYPE","ROOM NUM","NIGHTS
STAYED","RESERVATION STATUS"));

System.out.println("|==============================================================
=============================|");

          //Question iii)
          // getFirst & getNext , count & traversal
          Booking b = (Booking) linkedList.getFirst();
          while (b != null)
          {
             System.out.println(b.reportPerCust()); //prints a row of object b attribute
             b = (Booking) linkedList.getNext();    //gets the next node to store in object b
          }


System.out.println("|==============================================================
=============================|\n");

          //display linkedlist size
```

```
        System.out.println("Size of Linked List: " + linkedList.length());


 // iv)
        //display if list is empty
        System.out.println("Is the the list empty : " + linkedList.isEmpty() + "\n");

        do
        {
           //prompt user to modify data or not for linked list
           if(!userRespons.equalsIgnoreCase("YES") && !userRespons.equalsIgnoreCase("NO"))
           {
              userRespons = JOptionPane.showInputDialog("INVALID INPUT! \n\nWant to
REMOVE/UPDATE/INSERT data for linked list? \n(YES/NO) : ");
           }
           else
           {
              userRespons = JOptionPane.showInputDialog("Want to REMOVE/UPDATE/INSERT data
for linked list? \n(YES/NO) : ");
           }
        }
        while(!userRespons.equalsIgnoreCase("YES") && !userRespons.equalsIgnoreCase("NO"));

        if(userRespons.equalsIgnoreCase("YES"))
        {
           do
           {
              //prompt user to choose either remove or update data
              if(!remUpdIns.equalsIgnoreCase("REMOVE") &&
!remUpdIns.equalsIgnoreCase("UPDATE") && !remUpdIns.equalsIgnoreCase("INSERT"))
              {
                 remUpdIns = JOptionPane.showInputDialog("INVALID INPUT! \n\nEnter 'REMOVE' to
delete a data \nOR \nEnter 'UPDATE' to modify a data \nOR \nEnter 'INSERT' to add a data :");
              }
              else
              {
                 remUpdIns = JOptionPane.showInputDialog("Enter 'REMOVE' to delete a data \nOR
\nEnter 'UPDATE' to modify a data \nOR \nEnter 'INSERT' to add a data :");
              }
           }
           while(!remUpdIns.equalsIgnoreCase("REMOVE") &&
!remUpdIns.equalsIgnoreCase("UPDATE") && !remUpdIns.equalsIgnoreCase("INSERT"));
```

```
//prompt user to enter room number to remove or update data
            while(true) //infite loop until it breaks out
            {
              if(remUpdIns.equalsIgnoreCase("REMOVE") || remUpdIns.equalsIgnoreCase("UPDATE"))
              {
                try
                {
                  roomNum = Integer.parseInt(JOptionPane.showInputDialog("Enter room number
(1-1000) :"));

                  if(roomNum >0 && roomNum <=1000) //checks for valid room number
                  {
                    //excecute user requirements, either remove or update data

                    if(remUpdIns.equalsIgnoreCase("REMOVE")) //Question v) REMOVE
                    {
                      linkedList.remove(roomNum); //removes the choosen node
                    }
                    else if(remUpdIns.equalsIgnoreCase("UPDATE")) //Question vi) UPDATE
                    {
                      linkedList.update(roomNum); //updates data of the choosen node
                    }

                    break; //break out of infinite loop
                  }
                  else
                  {
                    JOptionPane.showMessageDialog(null,"INVALID INPUT!"); //display error message
because of invalid room number entered
                  }
                }
                catch(NumberFormatException e) //handles exception. catch user input error, input not an
int
                {
                  JOptionPane.showMessageDialog(null,"INVALID INPUT!"); //display error message
because of invalid room number entered
                }
              }
              else if(remUpdIns.equalsIgnoreCase("INSERT")) //user input
              {
                String guestIC = JOptionPane.showInputDialog("Enter guest IC number : ");
                String guestContactNum = JOptionPane.showInputDialog("Enter guest contact number :
");
                String roomType = "single";
                //Int roomNum;
                int nightsStayed;
                String reservationStatus = "paid";

                String insFroBac = "front";

                do
                {
                  //prompt user to enter valid room type
                  if(!roomType.equalsIgnoreCase("SINGLE") &&
!roomType.equalsIgnoreCase("KING"))
                  {
                    roomType = JOptionPane.showInputDialog("INVALID INPUT! \n\nEnter room type
(SINGLE/KING) : ");
                  }
                  else
                  {
                    roomType = JOptionPane.showInputDialog("Enter room type (SINGLE/KING) : ");
                  }
                }
                while(!roomType.equalsIgnoreCase("SINGLE") &&
!roomType.equalsIgnoreCase("KING"));

                while(true)
                {
                  try
                  {
                    //prompt user to enter valid room number
                    roomNum = Integer.parseInt(JOptionPane.showInputDialog("Enter room number
(1-1000) :"));

                    if(roomNum >0 && roomNum <=1000)
                    {
                      break;
                    }
                    else
                    {
                      JOptionPane.showMessageDialog(null,"INVALID INPUT!");
                    }
                  }
                  catch(NumberFormatException e)
```

```
                    {
                        JOptionPane.showMessageDialog(null,"INVALID INPUT!");
                    }
                }

                while(true)
                {
                    try
                    {
                        //prompt user to enter valid nights stayed
                        nightsStayed = Integer.parseInt(JOptionPane.showInputDialog("Enter nights stayed
:"));

                        if(roomNum >0)
                        {
                            break;
                        }
                        else
                        {
                            JOptionPane.showMessageDialog(null,"INVALID INPUT!");
                        }
                    }
                    catch(NumberFormatException e)
                    {
                        JOptionPane.showMessageDialog(null,"INVALID INPUT!");
                    }
                }

                do
                {
                    //prompt user to enter valid reservaion status
                    if(!reservationStatus.equalsIgnoreCase("PAID") &&
!reservationStatus.equalsIgnoreCase("PENDING"))
                    {
                        reservationStatus = JOptionPane.showInputDialog("INVALID INPUT! \n\nEnter
reservation status (PAID/PENDING) : ");
                    }
                    else
                    {
                        reservationStatus = JOptionPane.showInputDialog("Enter reservation status
(PAID/PENDING) : ");
                    }
                }
```

```
                        while(!reservationStatus.equalsIgnoreCase("PAID") &&
!reservationStatus.equalsIgnoreCase("PENDING"));

                        b = new Booking (guestIC , guestContactNum , roomType , roomNum , nightsStayed ,
reservationStatus);

                        do
                        {
                            //prompt user to enter data either from front or back of linked list
                            if(!insFroBac.equalsIgnoreCase("FRONT") &&
!insFroBac.equalsIgnoreCase("BACK"))
                            {
                                insFroBac = JOptionPane.showInputDialog("INVALID INPUT! \n\nWhere to insert
data (FRONT/BACK) : ");
                            }
                            else
                            {
                                insFroBac = JOptionPane.showInputDialog("Where to insert data (FRONT/BACK) :
");
                            }
                        }
                        while(!insFroBac.equalsIgnoreCase("FRONT") &&
!insFroBac.equalsIgnoreCase("BACK"));

                        if(insFroBac.equalsIgnoreCase("FRONT"))
                        {
                            linkedList.insertAtFront(b); //object b is inserted from the front of linked list
                            System.out.println("New item INSERTED at the FRONT");
                        }
                        else if(insFroBac.equalsIgnoreCase("BACK"))
                        {
                            linkedList.insertAtBack(b); //object b is inserted from the back of linked list
                            System.out.println("New item INSERTED at the BACK");
                        }

                        break;
                    }
                }
            }
        }

        //QUEUE

        //Question i) enqueue
        Booking tempCust = (Booking) linkedList.getFirst(); //gets the first node in linkedList to store in
tempCust

        while(tempCust != null) //loops while tempCust is not null
        {
           //determine the object b status reservation either "PAID" or "PENDING" customer
           if("PAID".equalsIgnoreCase(tempCust.getReservationStatus()))
           {
              queuePaid.enqueue(tempCust.reportPerCust()); //enqueue the object into paid customer queue
           }
           else if("PENDING".equalsIgnoreCase(tempCust.getReservationStatus())) //
tempCust.getResrvationaStatus().we
           {
              queuePending.enqueue(tempCust.reportPerCust()); //enqueue the object into pending customer
queue
           }

           tempCust = (Booking) linkedList.getNext(); //gets the next node in linked list to assign in
tempCust
        }

        userRespons = "YES";

        while(userRespons.equalsIgnoreCase("YES"))
        {
           do
           {
              //prompt user to enqueue new data
              if(!userRespons.equalsIgnoreCase("YES") && !userRespons.equalsIgnoreCase("NO"))
              {
                 userRespons = JOptionPane.showInputDialog("INVALID INPUT! \n\nWant to enqueue new
data ? \n(YES/NO) : ");
              }
              else
              {
                 userRespons = JOptionPane.showInputDialog("Want to enqueue new data ? \n(YES/NO) : ");
              }
           }
           while(!userRespons.equalsIgnoreCase("YES") && !userRespons.equalsIgnoreCase("NO"));

           if(userRespons.equalsIgnoreCase("YES"))
           {
              String guestIC = JOptionPane.showInputDialog("Enter guest IC number : ");
```

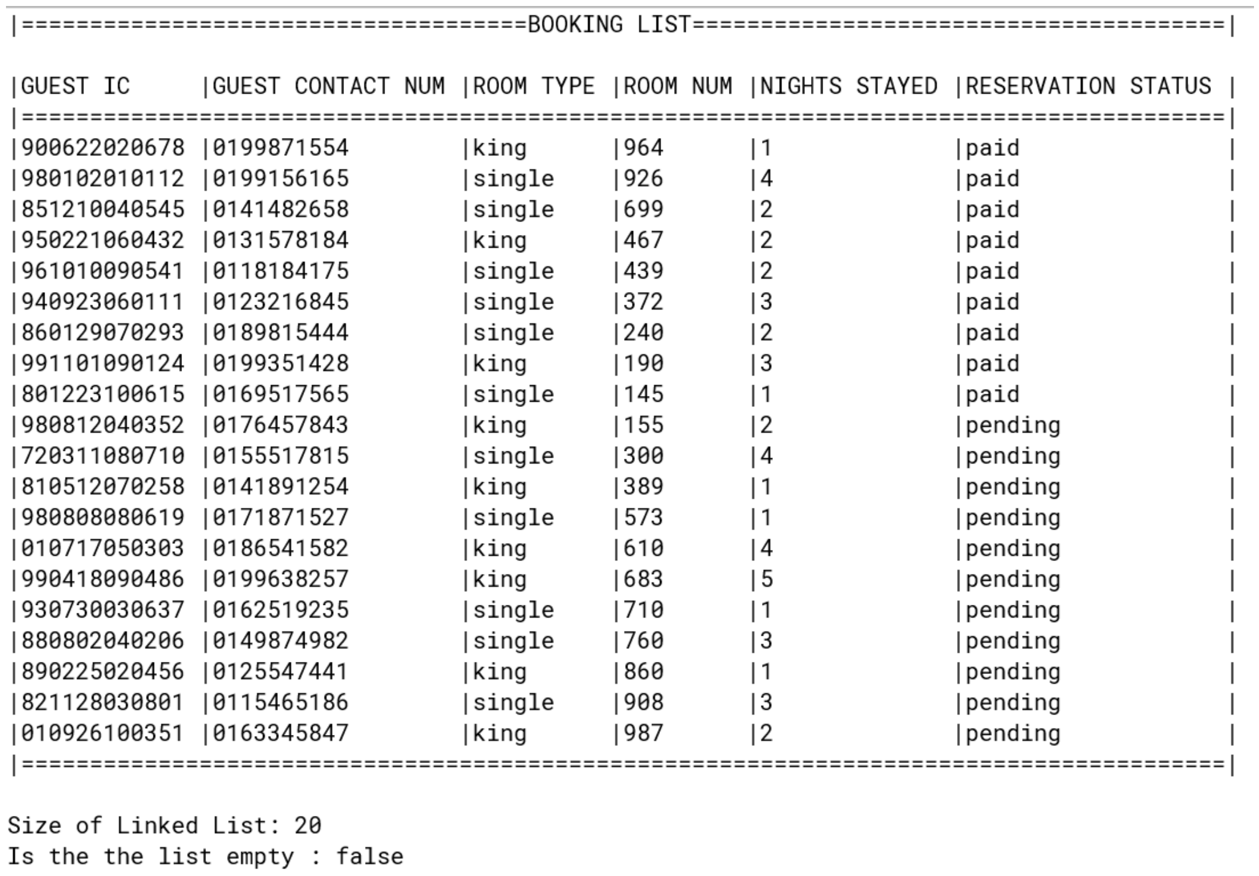
```
            String guestContactNum = JOptionPane.showInputDialog("Enter guest contact number : ");
            String roomType = "single";

//Int roomNum;
            int nightsStayed;
            String reservationStatus = "paid";

            do
            {
               //prompt user to enter valid room type
               if(!roomType.equalsIgnoreCase("SINGLE") && !roomType.equalsIgnoreCase("KING"))
               {
                  roomType = JOptionPane.showInputDialog("INVALID INPUT! \n\nEnter room type
(SINGLE/KING) : ");
               }
               else
               {
                  roomType = JOptionPane.showInputDialog("Enter room type (SINGLE/KING) : ");
               }
            }
            while(!roomType.equalsIgnoreCase("SINGLE") && !roomType.equalsIgnoreCase("KING"));

            while(true)
            {
               try
               {
                  //prompt user to enter valid room number
                  roomNum = Integer.parseInt(JOptionPane.showInputDialog("Enter room number (1-1000)
:"));

                  if(roomNum >0 && roomNum <=1000)
                  {
                     break;
                  }
                  else
                  {
                     JOptionPane.showMessageDialog(null,"INVALID INPUT!");
                  }
               }
               catch(NumberFormatException e)
               {
                  JOptionPane.showMessageDialog(null,"INVALID INPUT!");
               }
            }
            while(true)
            {
               try
               {
                  //prompt user to enter valid nights stayed
                  nightsStayed = Integer.parseInt(JOptionPane.showInputDialog("Enter nights stayed :"));

                  if(roomNum >0)
                  {
                      break;
                  }
                  else
                  {
                     JOptionPane.showMessageDialog(null,"INVALID INPUT!");
                  }
               }
               catch(NumberFormatException e)
               {
                  JOptionPane.showMessageDialog(null,"INVALID INPUT!");
               }
            }

            do
            {
               //prompt user to enter valid reservaion status
               if(!reservationStatus.equalsIgnoreCase("PAID") &&
!reservationStatus.equalsIgnoreCase("PENDING"))
               {
                  reservationStatus = JOptionPane.showInputDialog("INVALID INPUT! \n\nEnter
reservation status (PAID/PENDING) : ");
               }
               else
               {
                  reservationStatus = JOptionPane.showInputDialog("Enter reservation status
(PAID/PENDING) : ");
               }
            }
            while(!reservationStatus.equalsIgnoreCase("PAID") &&
!reservationStatus.equalsIgnoreCase("PENDING"));

            Booking b = new Booking (guestIC , guestContactNum , roomType , roomNum , nightsStayed ,
reservationStatus);
```

```
            //enqueuing new data from user input
            if("PAID".equalsIgnoreCase(b.getReservationStatus()))
            {
               queuePaid.enqueue(b.reportPerCust());
            }
            else if("PENDING".equalsIgnoreCase(b.getReservationStatus()))
            {
               queuePending.enqueue(b.reportPerCust());
            }

            System.out.println("New customer with room number " + b.getRoomNum() + " has been added
in queue");
         }
      }

      //Question ii) dequeue , iii) display size , iv) isEmpty

      //Paid Customer Queue

      System.out.println("\nSize of Paid Customer Queue : " + queuePaid.length()); //paid customer queue
size
      System.out.println("Is the queue empty : " + queuePaid.isEmpty() + "\n");  //checks if paid customer
queue size empty or not

      System.out.println("|=====================================PAID
CUSTOMER======================================|\n");
      System.out.println(String.format("|%-13s|%-18s|%-10s|%-9s|%-14s|%-19s|","GUEST IC","GUEST
CONTACT NUM","ROOM TYPE","ROOM NUM","NIGHTS STAYED","RESERVATION STATUS"));

System.out.println("|============================================================
============================|");

      while(!queuePaid.isEmpty()) //loops while queuePaid is not empty
      {
         System.out.println(queuePaid.dequeue()); //prints out the obj info
      }


System.out.println("|============================================================
============================|\n");

      //Pending Customer Queue
```

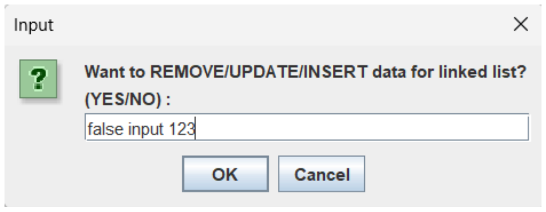
```
        System.out.println("Size of Pending Customer Queue : " + queuePending.length()); //pending
customer queue size
        System.out.println("Is the queue empty : " + queuePending.isEmpty() + "\n");     //checks if pending
customer queue size empty or not

        System.out.println("|======================================PENDING
CUSTOMER======================================|\n");
        System.out.println(String.format("|%-13s|%-18s|%-10s|%-9s|%-14s|%-19s|","GUEST IC","GUEST
CONTACT NUM","ROOM TYPE","ROOM NUM","NIGHTS STAYED","RESERVATION STATUS"));

System.out.println("|================================================================
==============================|");

        while(!queuePending.isEmpty()) //loops while queuePending is not empty
        {
          System.out.println(queuePending.dequeue()); //prints out the obj info
        }


System.out.println("|================================================================
==============================|");
    }
}
```

8.0 OUTPUT BOOKING APP

8.1 LINKED LIST



```
|=====================================BOOKING LIST=======================================|

|GUEST IC     |GUEST CONTACT NUM |ROOM TYPE |ROOM NUM |NIGHTS STAYED |RESERVATION STATUS |
|========================================================================================|
|900622020678 |0199871554        |king      |964      |1             |paid               |
|980102010112 |0199156165        |single    |926      |4             |paid               |
|851210040545 |0141482658        |single    |699      |2             |paid               |
|950221060432 |0131578184        |king      |467      |2             |paid               |
|961010090541 |0118184175        |single    |439      |2             |paid               |
|940923060111 |0123216845        |single    |372      |3             |paid               |
|860129070293 |0189815444        |single    |240      |2             |paid               |
|991101090124 |0199351428        |king      |190      |3             |paid               |
|801223100615 |0169517565        |single    |145      |1             |paid               |
|980812040352 |0176457843        |king      |155      |2             |pending            |
|720311080710 |0155517815        |single    |300      |4             |pending            |
|810512070258 |0141891254        |king      |389      |1             |pending            |
|980808080619 |0171871527        |single    |573      |1             |pending            |
|010717050303 |0186541582        |king      |610      |4             |pending            |
|990418090486 |0199638257        |king      |683      |5             |pending            |
|930730030637 |0162519235        |single    |710      |1             |pending            |
|880802040206 |0149874982        |single    |760      |3             |pending            |
|890225020456 |0125547441        |king      |860      |1             |pending            |
|821128030801 |0115465186        |single    |908      |3             |pending            |
|010926100351 |0163345847        |king      |987      |2             |pending            |
|========================================================================================|

Size of Linked List: 20
Is the the list empty : false
```

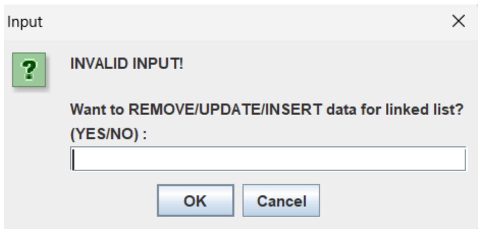


Input

Want to REMOVE/UPDATE/INSERT data for linked list?
(YES/NO) :

false input 123

→

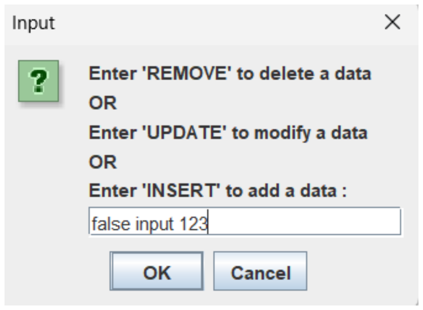


Input

INVALID INPUT!

Want to REMOVE/UPDATE/INSERT data for linked list?
(YES/NO) :

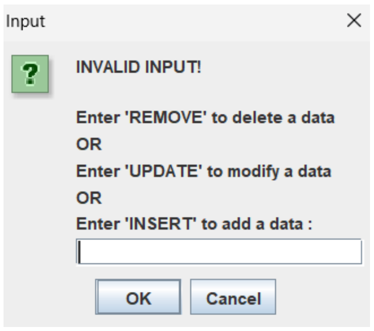
REMOVE


Input

Enter 'REMOVE' to delete a data
OR
Enter 'UPDATE' to modify a data
OR
Enter 'INSERT' to add a data :

false input 123

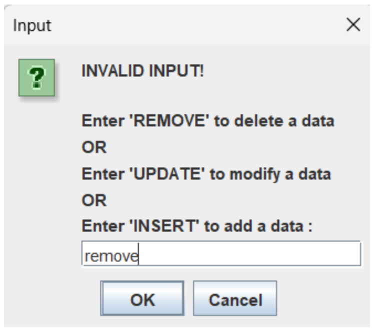
→


Input

INVALID INPUT!

Enter 'REMOVE' to delete a data
OR
Enter 'UPDATE' to modify a data
OR
Enter 'INSERT' to add a data :

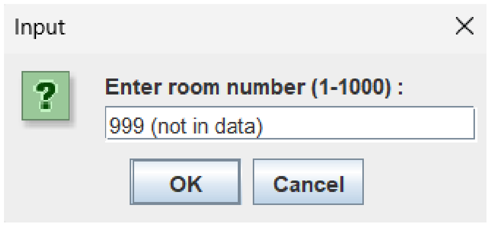
→


Input

INVALID INPUT!

Enter 'REMOVE' to delete a data
OR
Enter 'UPDATE' to modify a data
OR
Enter 'INSERT' to add a data :

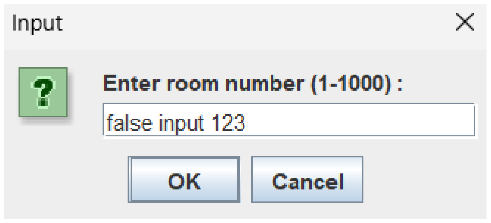
remove


Input

Enter room number (1-1000) :

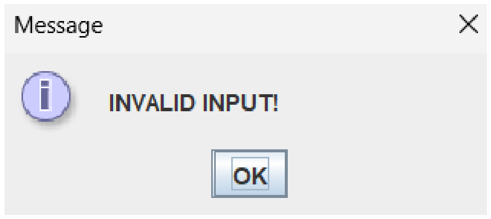
999 (not in data)

```
Room number 999 not found.
|===================================BOOKING LIST=====================================|

|GUEST IC     |GUEST CONTACT NUM |ROOM TYPE |ROOM NUM |NIGHTS STAYED |RESERVATION STATUS |
|===================================================================================|
```

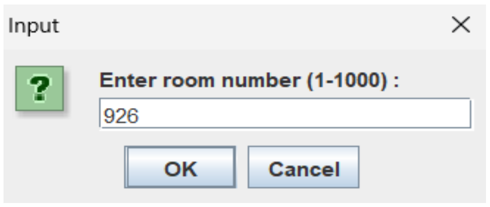

Input

Enter room number (1-1000) :

false input 123


Message

INVALID INPUT!

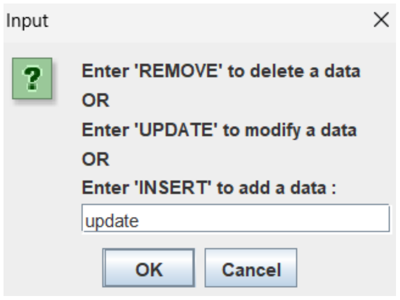

Input

Enter room number (1-1000) :

926

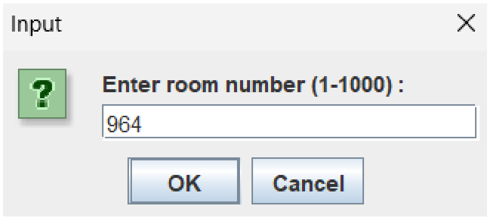
```
Booking with room number 926 removed.
|=====================================BOOKING LIST=====================================|

|GUEST IC     |GUEST CONTACT NUM |ROOM TYPE |ROOM NUM |NIGHTS STAYED |RESERVATION STATUS |
|=======================================================================================|
|900622020678 |0199871554        |king      |964      |1             |paid               |
|851210040545 |0141482658        |single    |699      |2             |paid               |
|950221060432 |0131578184        |king      |467      |2             |paid               |
|961010090541 |0118184175        |single    |439      |2             |paid               |
|940923060111 |0123216845        |single    |372      |3             |paid               |
|860129070293 |0189815444        |single    |240      |2             |paid               |
|991101090124 |0199351428        |king      |190      |3             |paid               |
|801223100615 |0169517565        |single    |145      |1             |paid               |
|980812040352 |0176457843        |king      |155      |2             |pending            |
|720311080710 |0155517815        |single    |300      |4             |pending            |
|810512070258 |0141891254        |king      |389      |1             |pending            |
|980808080619 |0171871527        |single    |573      |1             |pending            |
|010717050303 |0186541582        |king      |610      |4             |pending            |
|990418090486 |0199638257        |king      |683      |5             |pending            |
|930730030637 |0162519235        |single    |710      |1             |pending            |
|880802040206 |0149874982        |single    |760      |3             |pending            |
|890225020456 |0125547441        |king      |860      |1             |pending            |
|821128030801 |0115465186        |single    |908      |3             |pending            |
|010926100351 |0163345847        |king      |987      |2             |pending            |
|=======================================================================================|

Size of Linked List: 19
Is the the list empty : false
```

UPDATE


Input

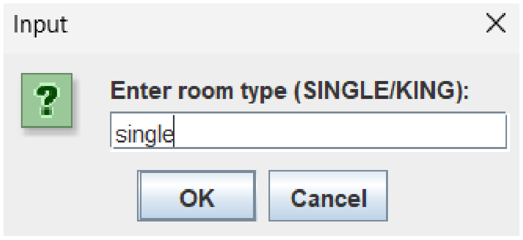
Enter 'REMOVE' to delete a data
OR
Enter 'UPDATE' to modify a data
OR
Enter 'INSERT' to add a data :

update


Input

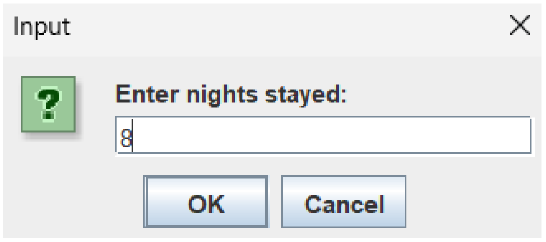
Enter room number (1-1000) :

964


Input

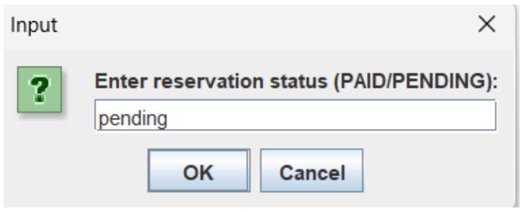
Enter room type (SINGLE/KING):

single


Input

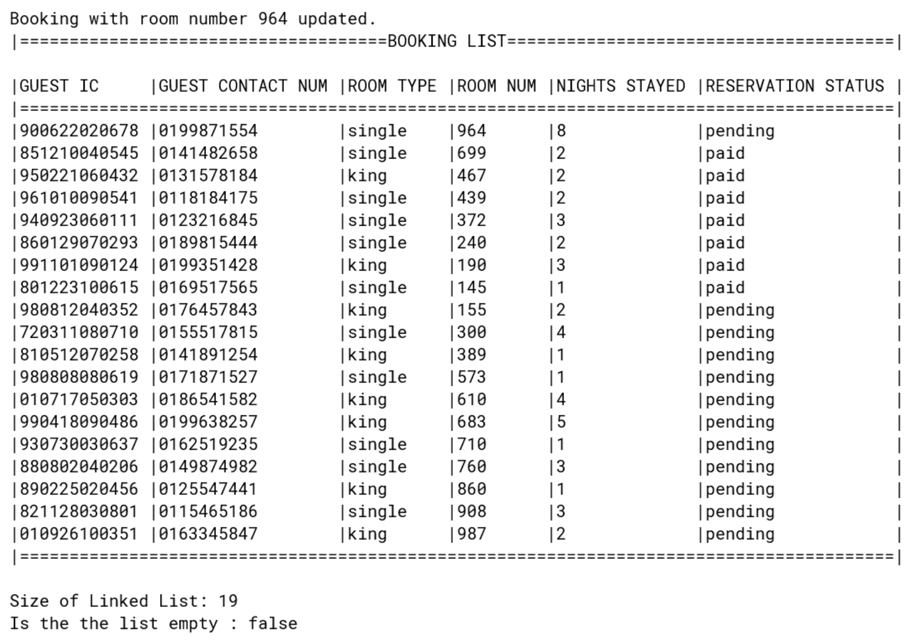
Enter nights stayed:

8

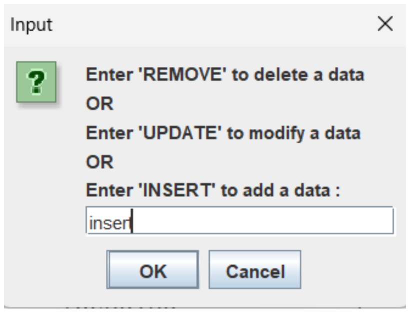

Input

Enter reservation status (PAID/PENDING):

pending

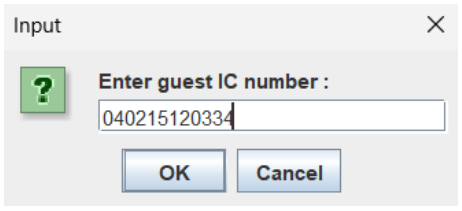

```
Booking with room number 964 updated.
|==================================BOOKING LIST==================================|

|GUEST IC     |GUEST CONTACT NUM |ROOM TYPE |ROOM NUM |NIGHTS STAYED |RESERVATION STATUS |
|=================================================================================|
|900622020678 |0199871554        |single    |964      |8             |pending            |
|851210040545 |0141482658        |single    |699      |2             |paid               |
|950221060432 |0131578184        |king      |467      |2             |paid               |
|961010090541 |0118184175        |single    |439      |2             |paid               |
|940923060111 |0123216845        |single    |372      |3             |paid               |
|860129070293 |0189815444        |single    |240      |2             |paid               |
|991101090124 |0199351428        |king      |190      |3             |paid               |
|801223100615 |0169517565        |single    |145      |1             |paid               |
|980812040352 |0176457843        |king      |155      |2             |pending            |
|720311080710 |0155517815        |single    |300      |4             |pending            |
|810512070258 |0141891254        |king      |389      |1             |pending            |
|980808080619 |0171871527        |single    |573      |1             |pending            |
|010717050303 |0186541582        |king      |610      |4             |pending            |
|990418090486 |0199638257        |king      |683      |5             |pending            |
|930730030637 |0162519235        |single    |710      |1             |pending            |
|880802040206 |0149874982        |single    |760      |3             |pending            |
|890225020456 |0125547441        |king      |860      |1             |pending            |
|821128030801 |0115465186        |single    |908      |3             |pending            |
|010926100351 |0163345847        |king      |987      |2             |pending            |
|=================================================================================|

Size of Linked List: 19
Is the the list empty : false
```

INSERT AT FRONT


Input

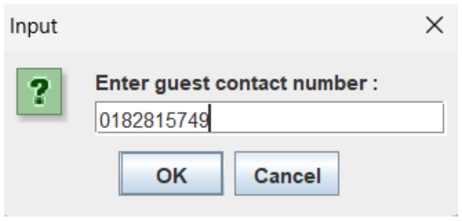
Enter 'REMOVE' to delete a data
OR
Enter 'UPDATE' to modify a data
OR
Enter 'INSERT' to add a data :

insert


Input

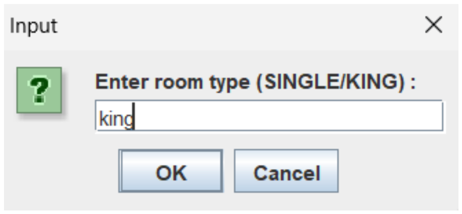
Enter guest IC number :

040215120334


Input

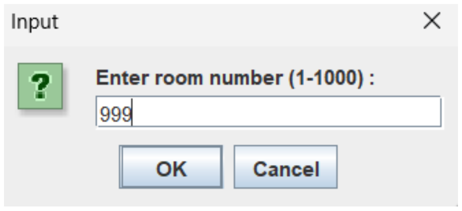
Enter guest contact number :

0182815749


Input

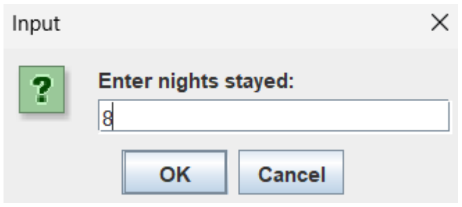
Enter room type (SINGLE/KING) :

king


Input

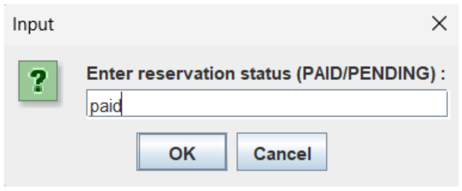
Enter room number (1-1000) :

999


Input

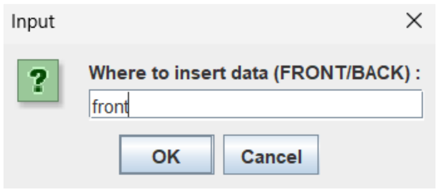
Enter nights stayed:

8


Input

Enter reservation status (PAID/PENDING) :

paid

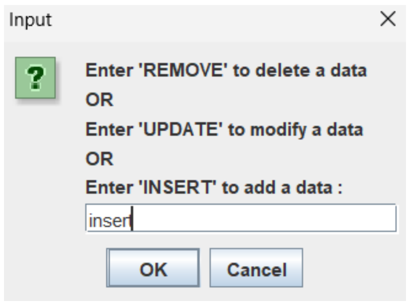

Input

Where to insert data (FRONT/BACK) :

front

```
New item INSERTED at the FRONT
|=====================================BOOKING LIST=======================================|

|GUEST IC     |GUEST CONTACT NUM |ROOM TYPE |ROOM NUM |NIGHTS STAYED |RESERVATION STATUS |
|========================================================================================|
|04021520334  |0182815749        |king      |999      |8             |paid               |
|900622020678 |0199871554        |single    |964      |8             |pending            |
|851210040545 |0141482658        |single    |699      |2             |paid               |
|950221060432 |0131578184        |king      |467      |2             |paid               |
|961010090541 |0118184175        |single    |439      |2             |paid               |
|940923060111 |0123216845        |single    |372      |3             |paid               |
|860129070293 |0189815444        |single    |240      |2             |paid               |
|991101090124 |0199351428        |king      |190      |3             |paid               |
|801223100615 |0169517565        |single    |145      |1             |paid               |
|980812040352 |0176457843        |king      |155      |2             |pending            |
|720311080710 |0155517815        |single    |300      |4             |pending            |
|810512070258 |0141891254        |king      |389      |1             |pending            |
|980808080619 |0171871527        |single    |573      |1             |pending            |
|010717050303 |0186541582        |king      |610      |4             |pending            |
|990418090486 |0199638257        |king      |683      |5             |pending            |
|930730030637 |0162519235        |single    |710      |1             |pending            |
|880802040206 |0149874982        |single    |760      |3             |pending            |
|890225020456 |0125547441        |king      |860      |1             |pending            |
|821128030801 |0115465186        |single    |908      |3             |pending            |
|010926100351 |0163345847        |king      |987      |2             |pending            |
|========================================================================================|

Size of Linked List: 20
Is the the list empty : false
```

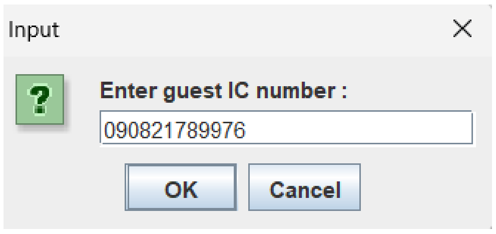
INSERT AT BACK



Input

Enter 'REMOVE' to delete a data
OR
Enter 'UPDATE' to modify a data
OR
Enter 'INSERT' to add a data :

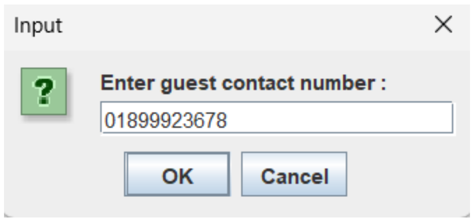
insert



Input

Enter guest IC number :

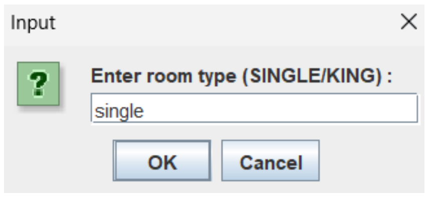
090821789976



Input

Enter guest contact number :

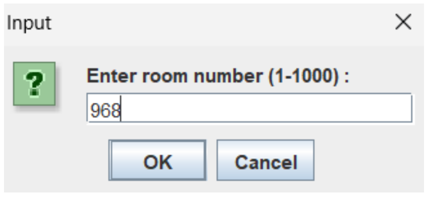
01899923678



Input

Enter room type (SINGLE/KING) :

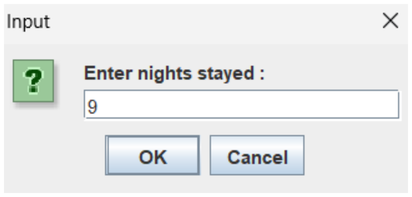
single



Input

Enter room number (1-1000) :

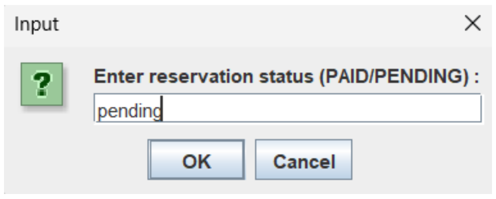
968



Input

Enter nights stayed :

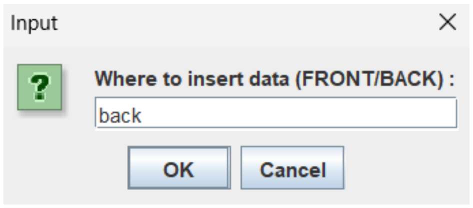
9



Input

Enter reservation status (PAID/PENDING) :

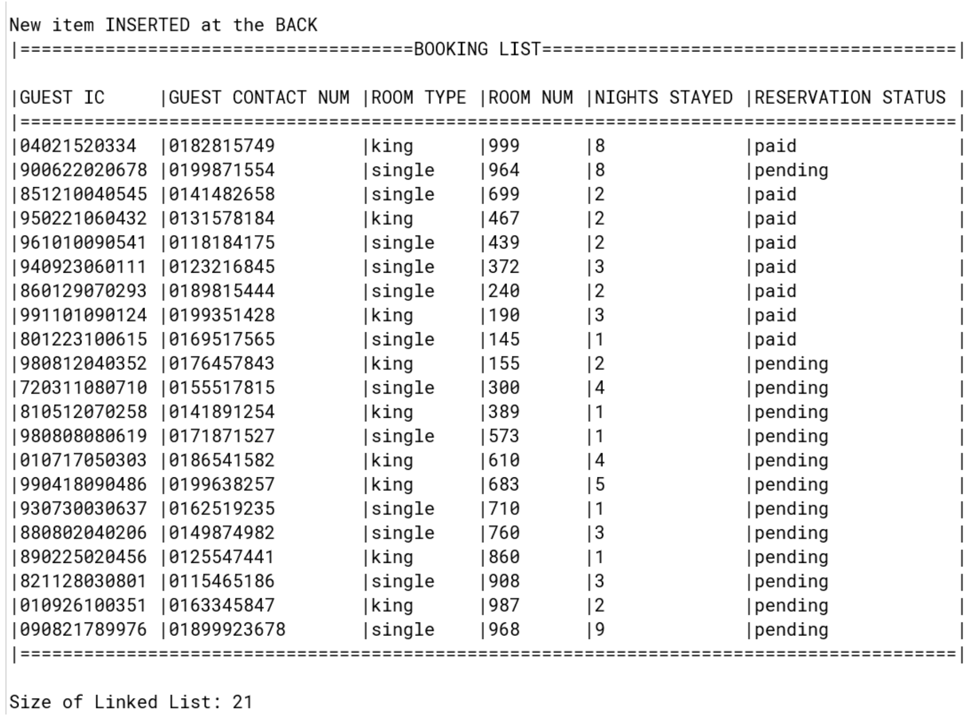
pending



Input

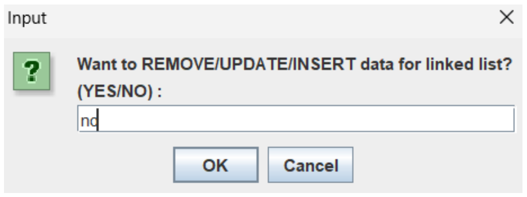
Where to insert data (FRONT/BACK) :

back



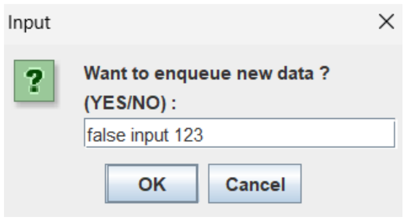
```
New item INSERTED at the BACK
|===================================BOOKING LIST=====================================|

|GUEST IC     |GUEST CONTACT NUM |ROOM TYPE |ROOM NUM |NIGHTS STAYED |RESERVATION STATUS |
|====================================================================================|
|04021520334  |0182815749        |king      |999      |8             |paid               |
|900622020678 |0199871554        |single    |964      |8             |pending            |
|851210040545 |0141482658        |single    |699      |2             |paid               |
|950221060432 |0131578184        |king      |467      |2             |paid               |
|961010090541 |0118184175        |single    |439      |2             |paid               |
|940923060111 |0123216845        |single    |372      |3             |paid               |
|860129070293 |0189815444        |single    |240      |2             |paid               |
|991101090124 |0199351428        |king      |190      |3             |paid               |
|801223100615 |0169517565        |single    |145      |1             |paid               |
|980812040352 |0176457843        |king      |155      |2             |pending            |
|720311080710 |0155517815        |single    |300      |4             |pending            |
|810512070258 |0141891254        |king      |389      |1             |pending            |
|980808080619 |0171871527        |single    |573      |1             |pending            |
|010717050303 |0186541582        |king      |610      |4             |pending            |
|990418090486 |0199638257        |king      |683      |5             |pending            |
|930730030637 |0162519235        |single    |710      |1             |pending            |
|880802040206 |0149874982        |single    |760      |3             |pending            |
|890225020456 |0125547441        |king      |860      |1             |pending            |
|821128030801 |0115465186        |single    |908      |3             |pending            |
|010926100351 |0163345847        |king      |987      |2             |pending            |
|090821789976 |01899923678       |single    |968      |9             |pending            |
|====================================================================================|

Size of Linked List: 21
```

8.2 QUEUE


Input

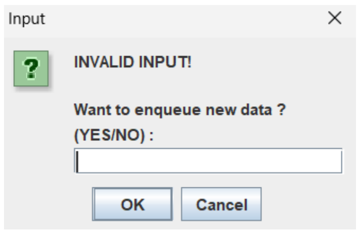
Want to REMOVE/UPDATE/INSERT data for linked list? (YES/NO) :

no


Input

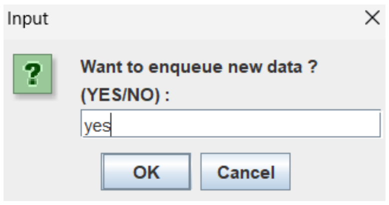
Want to enqueue new data ? (YES/NO) :

false input 123


Input

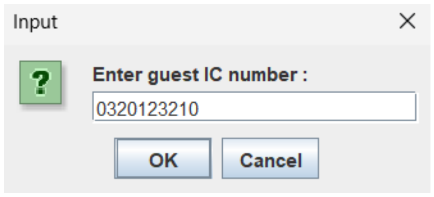
INVALID INPUT!

Want to enqueue new data ? (YES/NO) :


Input

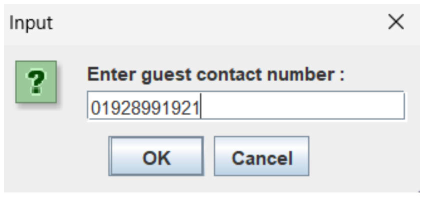
Want to enqueue new data ? (YES/NO) :

yes


Input

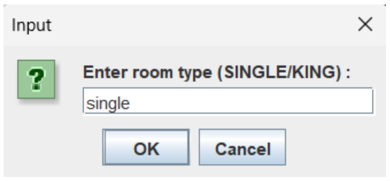
Enter guest IC number :

0320123210


Input

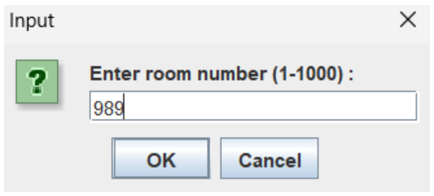
Enter guest contact number :

01928991921


Input

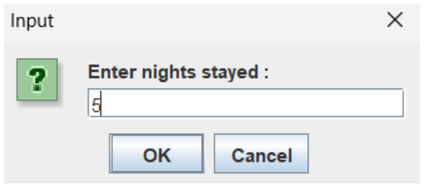
Enter room type (SINGLE/KING) :

single


Input

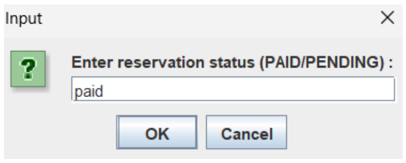
Enter room number (1-1000) :

989


Input

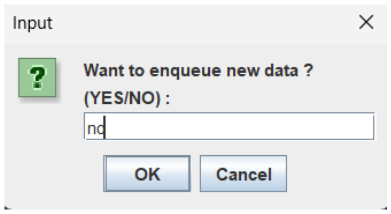
Enter nights stayed :

5


Input

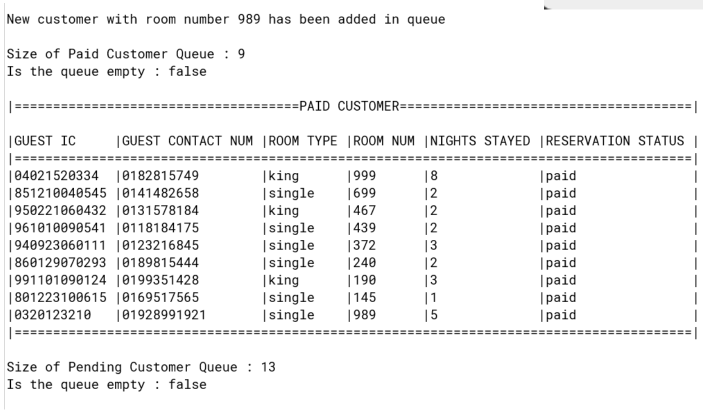
Enter reservation status (PAID/PENDING) :

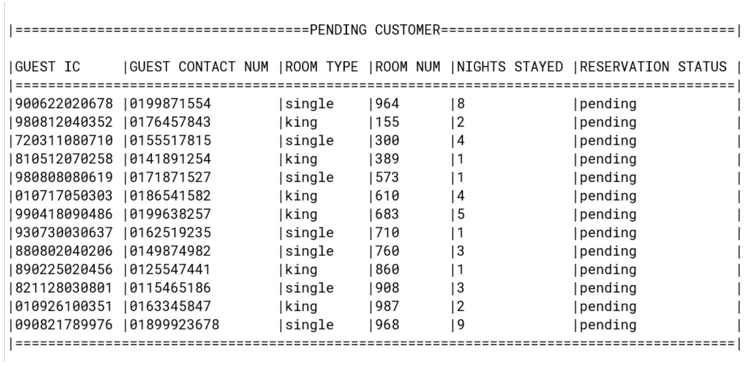
paid


Input

Want to enqueue new data ? (YES/NO) :

no


```
New customer with room number 989 has been added in queue

Size of Paid Customer Queue : 9
Is the queue empty : false

|=====================================PAID CUSTOMER======================================|

|GUEST IC     |GUEST CONTACT NUM |ROOM TYPE |ROOM NUM |NIGHTS STAYED |RESERVATION STATUS |
|========================================================================================|
|04021520334  |0182815749        |king      |999      |8             |paid               |
|851210040545 |0141482658        |single    |699      |2             |paid               |
|950221060432 |0131578184        |king      |467      |2             |paid               |
|961010090541 |0118184175        |single    |439      |2             |paid               |
|940923060111 |0123216845        |single    |372      |3             |paid               |
|860129070293 |0189815444        |single    |240      |2             |paid               |
|991101090124 |0199351428        |king      |190      |3             |paid               |
|801223100615 |0169517565        |single    |145      |1             |paid               |
|0320123210   |01928991921       |single    |989      |5             |paid               |
|========================================================================================|

Size of Pending Customer Queue : 13
Is the queue empty : false
```


```
|====================================PENDING CUSTOMER====================================|

|GUEST IC     |GUEST CONTACT NUM |ROOM TYPE |ROOM NUM |NIGHTS STAYED |RESERVATION STATUS |
|========================================================================================|
|900622020678 |0199871554        |single    |964      |8             |pending            |
|980812040352 |0176457843        |king      |155      |2             |pending            |
|720311080710 |0155517815        |single    |300      |4             |pending            |
|810512070258 |0141891254        |king      |389      |1             |pending            |
|980808080619 |0171871527        |single    |573      |1             |pending            |
|010717050303 |0186541582        |king      |610      |4             |pending            |
|990418090486 |0199638257        |king      |683      |5             |pending            |
|930730030637 |0162519235        |single    |710      |1             |pending            |
|880802040206 |0149874982        |single    |760      |3             |pending            |
|890225020456 |0125547441        |king      |860      |1             |pending            |
|821128030801 |0115465186        |single    |908      |3             |pending            |
|010926100351 |0163345847        |king      |987      |2             |pending            |
|090821789976 |01899923678       |single    |968      |9             |pending            |
|========================================================================================|
```