Assignment1_E1900344

by I Nyoman Surya Pradipta

Submission date: 05-Nov-2020 08:27PM (UTC+0800)

Submission ID: 1436071701

File name: E1900344.docx (1.2M)

Word count: 1291

Character count: 7962

ASSIGNMENT 1

LEVEL 3

BIT 106

INTRODUCTION TO PROGRAMMING IN JAVA

HELP University

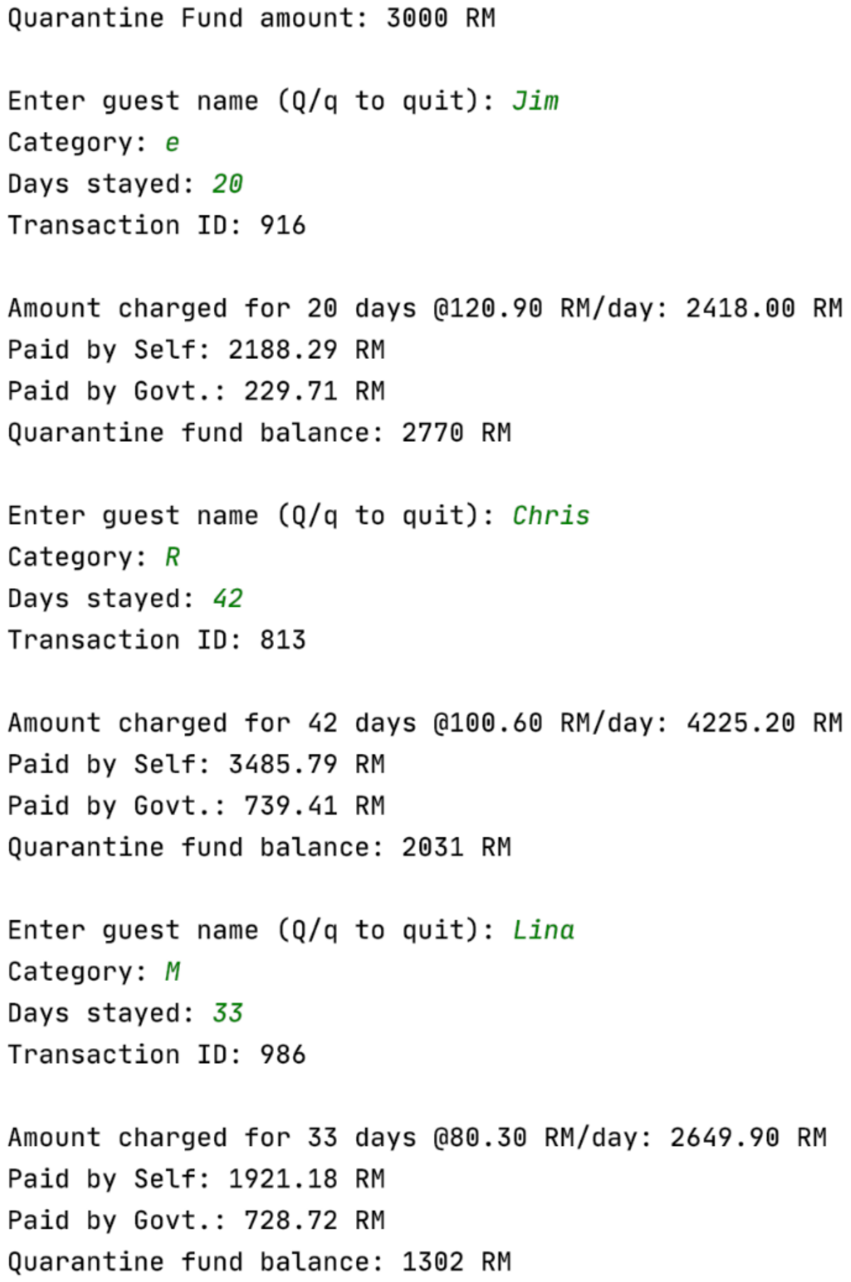
university of achievers

Name: I Nyoman Surya Pradipta

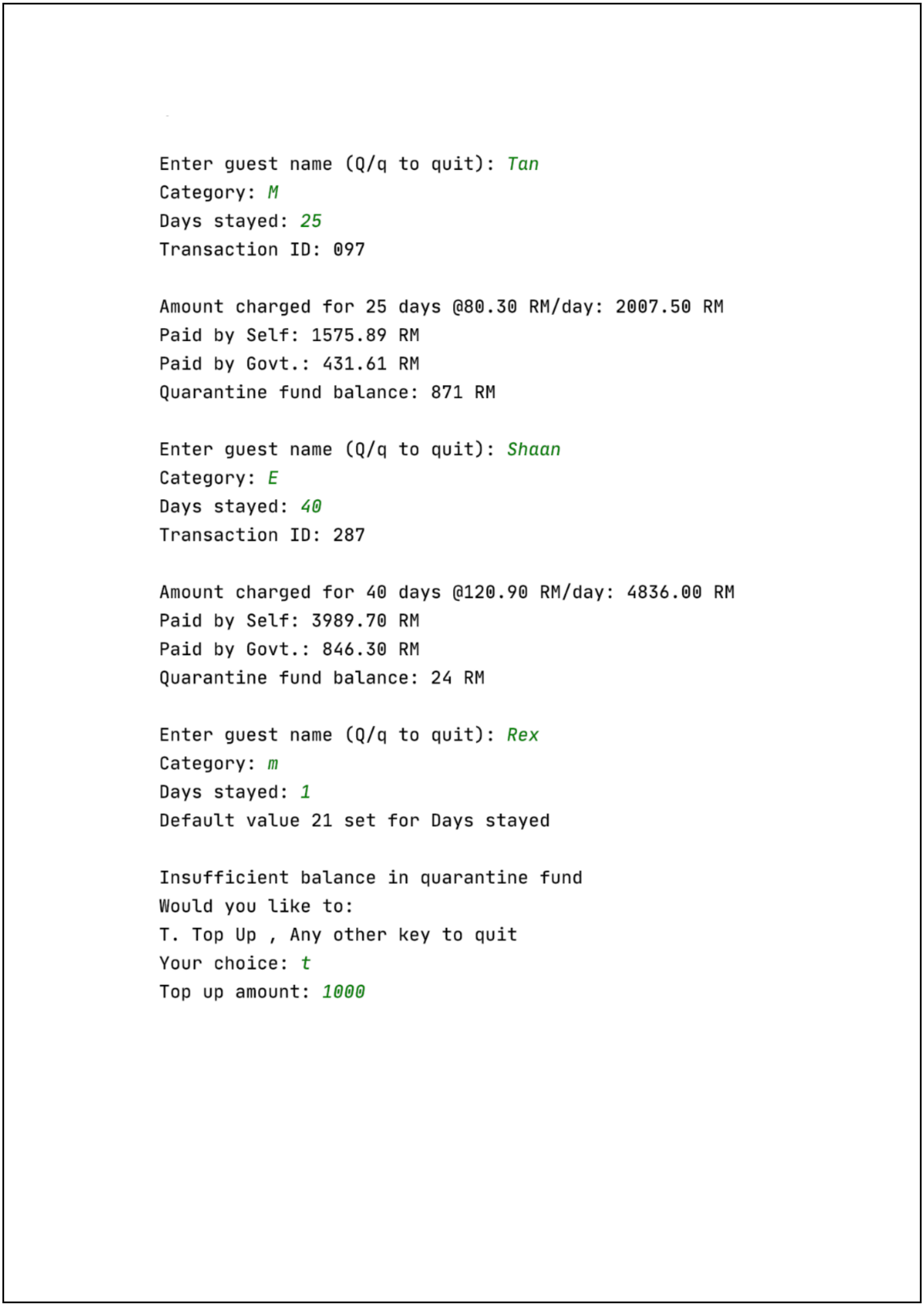
Student ID: E1900344

```
import java.util.*; // Scanner class is in this package

/**
 * Java program to HELP QUARANTINE, a simple application to calculate
 * the final bill of quarantined guests and some other information.
 * Written by I Nyoman Surya Pradipta
 * Written on 11 October 2020
 * Student ID : E1900344
 */

public class E1900344 {
    /**
     * The main method which makes use to display
     * input and output from the program.
     * @param args Not used.
     * @return void No value.
     */
    public static void main(String[] args){
        // Declare and assign to variables.
        String guestName = "", topUp, nationality;
        char category;
        int numOfTrans = 0, malayCount = 0, foreignCount = 0,
            topUpAmount, highestIndex = 0;
        byte daysStayed;
        double charge, amountCharged, paidByGovt, paidBySelf,
            totalAmount = 0, malayAvg = 0, foreignAvg = 0,
            quarantineFund = 3000;

        // Declare array list.
        var highestName = new ArrayList<String>();
        var highestCategory = new ArrayList<String>();
        var highestAmountPaid = new ArrayList<Double>();

        // Print the output of Quarantine Fund.
        System.out.printf("Quarantine Fund amount: %.0f RM\n",
quarantineFund);
        // Create a Scanner object to read data.
        Scanner input = new Scanner(System.in);
        while(!(guestName.equalsIgnoreCase("q"))){ // Testing.
            // Read each guest's name.
            System.out.print("\nEnter guest name (Q/q to quit): ");
            guestName = input.next();

            if (!(guestName.equalsIgnoreCase("q"))) { // Testing.
                // Read each category from the guest.
                System.out.print("Category: ");
                category = input.next().charAt(0);

                // Validate user input category.
                while (!validateCategory(category)) { // Testing.
                    // Read until a valid value provided.
                    System.out.println("Invalid Category");
                    System.out.print("Please enter category: ");
                    category = input.next().charAt(0);
                }

                // Read each day stayed from the guest.
                System.out.print("Days stayed: ");
                daysStayed = input.nextByte();
                /*
                Validate the number of days stayed between 14-42,
                otherwise the default value of 21
                will be assign and printed
                */
                if (!validateDaysStayed(daysStayed)) { // Testing.
                    daysStayed = 21;
                    System.out.println("Default value " + daysStayed + "
set for Days stayed");
                }

                // The program will ask for top up or quit.
                if  (quarantineFund < 500) { // Testing.
                    // Read each choice.
                    System.out.println("\nInsufficient balance in
quarantine fund");
                    System.out.println("Would you like to:");
                    System.out.println("T. Top Up , Any other key to
quit");
                    System.out.print("Your choice: ");
                    topUp = input.next();

                    //Read "T" to top up or other key to quit.
                    if (topUp.equalsIgnoreCase("t")) { // Testing.
                        System.out.print("Top up amount: ");
                        topUpAmount = input.nextInt();
                        quarantineFund += topUpAmount;
                        System.out.printf("Quarantine fund balance: %.0f
RM\n", quarantineFund);
                    } else {
                        break; // program will be quit.
                    }
                } else {
                    // Print the output auto generate transaction ID.
                    System.out.printf("Transaction ID: %03d\n\n", (int)
autoGenerateTransID());
                    if (category == 'm' || category == 'M') {
                        /*
                         Initialize to add Malaysian
                         to array highest category.
                         */
                        nationality = "Malaysian";
                        charge = 80.3;
                        malayCount++; // Increase count from Malaysian.
                        /*
                         Add amount charged to malayAvg
                         to get average from Malaysian.
                         */
                        malayAvg +=
calculateAmountCharged(daysStayed,charge);
                    } else if (category == 'r' || category == 'R') {
                        /*
                         Initialize to add Resident
                         to array highest category.
                         */
                        nationality = "Resident";
                        charge = 100.6;
                        foreignCount++; // Increase count from Foreigner.
                        /*
                         Add amount charged to foreignAvg
                         to get average from Foreigner.
                         */
                        foreignAvg +=
calculateAmountCharged(daysStayed,charge);
                    } else {
                        /*
                         Initialize to add Expat
                         to array highest category.
                         */
                        nationality = "Expat";
                        charge = 120.9;
                        foreignCount++; // Increase count from Foreigner.
                        /*
                         Add amount charged to foreignAvg
                         to get average from Foreigner.
                         */
                        foreignAvg +=
calculateAmountCharged(daysStayed,charge);
                    }

                    // Assign calculation method to variables.
                    amountCharged = calculateAmountCharged(daysStayed,
charge);
                    paidByGovt = calculatePaidByGovt(category,
amountCharged);
                    paidBySelf = calculatePaidBySelf(amountCharged,
paidByGovt);

                    /*
                     Add data highest name, category,
                     and amount paid to array list.
                     */
                    highestName.add(guestName);
                    highestCategory.add(nationality);
                    highestAmountPaid.add(amountCharged);
                    highestIndex =
highestAmountPaidIndex(highestAmountPaid);

                    // Add amountCharged to totalAmount.
                    totalAmount += amountCharged;
                    /*
                     Add malayCount and foreignCount
                     to numOfTrans.
                     */
                    numOfTrans = malayCount + foreignCount;
                    // Subtraction paid by government to quarantine fund.
                    quarantineFund -= paidByGovt;

                    /*
                     Print the output of amount charged, paid by self,
                     government, and quarantine fund.
                     */
                    System.out.printf("Amount charged for %d days @%.2f
RM/day: %.2f RM\n",
                        daysStayed, charge, amountCharged);
                    System.out.printf("Paid by Self: %.2f RM\n",
paidBySelf);
                    System.out.printf("Paid by Govt.: %.2f RM\n",
paidByGovt);
                    System.out.printf("Quarantine fund balance: %.0f RM\n",
quarantineFund);
                }
            }
        }
        //If the program quit, some information will be printed.
        if (numOfTrans == 0) { // Testing.
            System.out.println("No transactions");
        } else {
            int rows = 43;
            for (int i = 1; i <= rows; ++i) {
                System.out.print('-');
            }
            /*
             Print output total, highest, average amount,
             and quarantine fund balance.
             */
            System.out.printf("\n\nTotal Amount from %d guests is %.2f 
RM\n", numOfTrans, totalAmount );
            System.out.printf("Highest amount paid %.2f RM by %s, is an 
%s\n", highestAmountPaid.get(highestIndex),
                highestName.get(highestIndex), 
highestCategory.get(highestIndex));
            System.out.printf("Average amount from Malaysians: %.2f RM\n", 
calculateAvg(malayAvg, malayCount));
            System.out.printf("Average amount from Foreigners: %.2f RM\n", 
calculateAvg(foreignAvg, foreignCount));
            System.out.printf("Quarantine fund balance: %.0f RM\n", 
quarantineFund);
        }
    }

    /**
     * This method is used to validate nationality among Malaysian, 
Resident, and Expat
     * until a valid value is provided.
     * @param category The category value to validate.
     * @return boolean Expressions.
     */
    public static boolean validateCategory(char category) {
        boolean returnValue;
        returnValue =
            category == 'M' || category == 'm' ||
            category == 'R' || category == 'r' ||
            category == 'E' || category == 'e';
        return returnValue;
    }

    /**
     * This method is used to validate day stayed
     * @param daysStayed The daysStayed value to validate.
     * @return boolean Expressions.
     */
    public static boolean validateDaysStayed(byte daysStayed){
        boolean returnValue;
        returnValue  = daysStayed >= 14 && daysStayed <= 42;
        return returnValue;
    }

    /**
     * This method is used to calculation auto generate transaction ID.
     * @return double Calculation auto generate number between 001-999.
     */
    public static double autoGenerateTransID() {
        return (Math.random() * 999) + 1;
    }
    /**
     * This method is used to calculation amount charged.
     * @param daysStayed Value to be multiplied charge.
     * @param charge Value to be multiplied daysStayed.
     * @return double Multiplication daysStayed and charge.
     */
    public static double calculateAmountCharged(byte daysStayed, double
charge) {
        return daysStayed * charge;
    }

    /**
     * This method is used to calculation charges paid by Government
     * according to nationality (malaysian or foreigner) and amount
charged.
     * @param category Value to check statements.
     * @param amountCharged Value to be multiplied discount.
     * @return double Multiplication by the amount charged and discount
     * to get the charge.
     */
    public static double calculatePaidByGovt(char category, double
amountCharged) {
        double charge;
        if (category == 'm' || category == 'M') {
            if (amountCharged >= 2500)
                charge = amountCharged * 275/1000;
            else if (amountCharged >= 2000)
                charge = amountCharged * 215/1000;
            else if (amountCharged >= 1500)
                charge = amountCharged * 15/100;
            else
                charge = 100;
        } else {
            if (amountCharged >= 3000)
                charge = amountCharged * 175/1000;
            else if (amountCharged >= 2000)
                charge = amountCharged * 95/1000;
            else
                charge = 80;
        }
        return charge;
    }

    /**
     * This method is used to calculation charges paid by Self.
     * @param amountCharged Value to be multiplied by paidByGovt.
     * @param paidByGovt Value to be multiplied by amountCharged.
     * @return double This returns subtraction by amountCharged and charge.
     */
    public static double calculatePaidBySelf(double amountCharged, double
paidByGovt) {
        return amountCharged - paidByGovt;
    }

    /**
     * This method is used to calculation average from nationality
(malaysian or foreigner).
     * @param nationality Value to be division by nationCount.
     * @param nationCount  Value to be division by nationality.
     * @return double Division by nationality and count.
     */
    public static double calculateAvg (double nationality, int nationCount)
{
        return nationality / nationCount;
    }

    /**
     * This method is used to calculation to get index from array.
     * @param dataAmountPaid Value data in array.
     * @return int Highest index.
     */
    public static int highestAmountPaidIndex (ArrayList<Double>
dataAmountPaid) {
        // Declare and initialize.
        Double[] arr = dataAmountPaid.toArray(Double[]::new);
        Double highest = arr[0];
        int highestIndex = 0;
        for (int i = 0; i < arr.length; i++) {
            if (highest < arr[i]) {
                highest = arr[i]; // Get the highest amount.
                highestIndex = i; // Get the highest index.
            }
        }
        return highestIndex;
    }
}
```

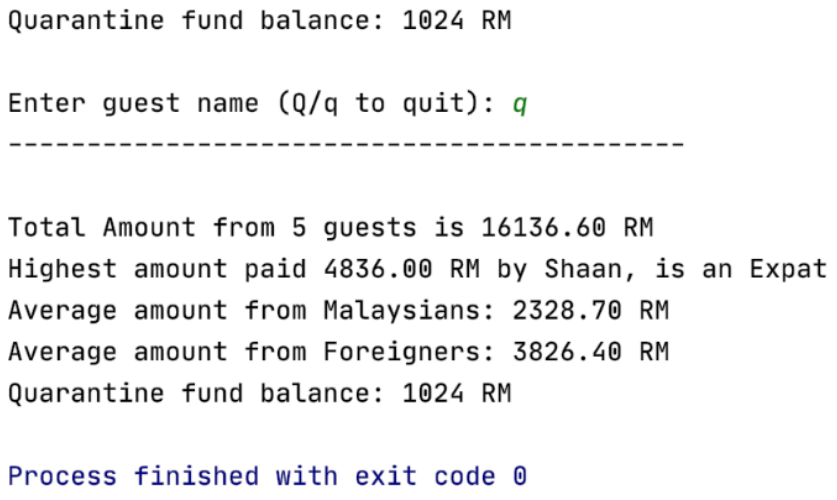
Output



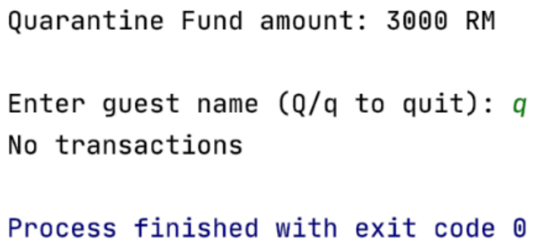
```
Quarantine Fund amount: 3000 RM

Enter guest name (Q/q to quit): Jim
Category: e
Days stayed: 20
Transaction ID: 916

Amount charged for 20 days @120.90 RM/day: 2418.00 RM
Paid by Self: 2188.29 RM
Paid by Govt.: 229.71 RM
Quarantine fund balance: 2770 RM

Enter guest name (Q/q to quit): Chris
Category: R
Days stayed: 42
Transaction ID: 813

Amount charged for 42 days @100.60 RM/day: 4225.20 RM
Paid by Self: 3485.79 RM
Paid by Govt.: 739.41 RM
Quarantine fund balance: 2031 RM

Enter guest name (Q/q to quit): Lina
Category: M
Days stayed: 33
Transaction ID: 986

Amount charged for 33 days @80.30 RM/day: 2649.90 RM
Paid by Self: 1921.18 RM
Paid by Govt.: 728.72 RM
Quarantine fund balance: 1302 RM
```



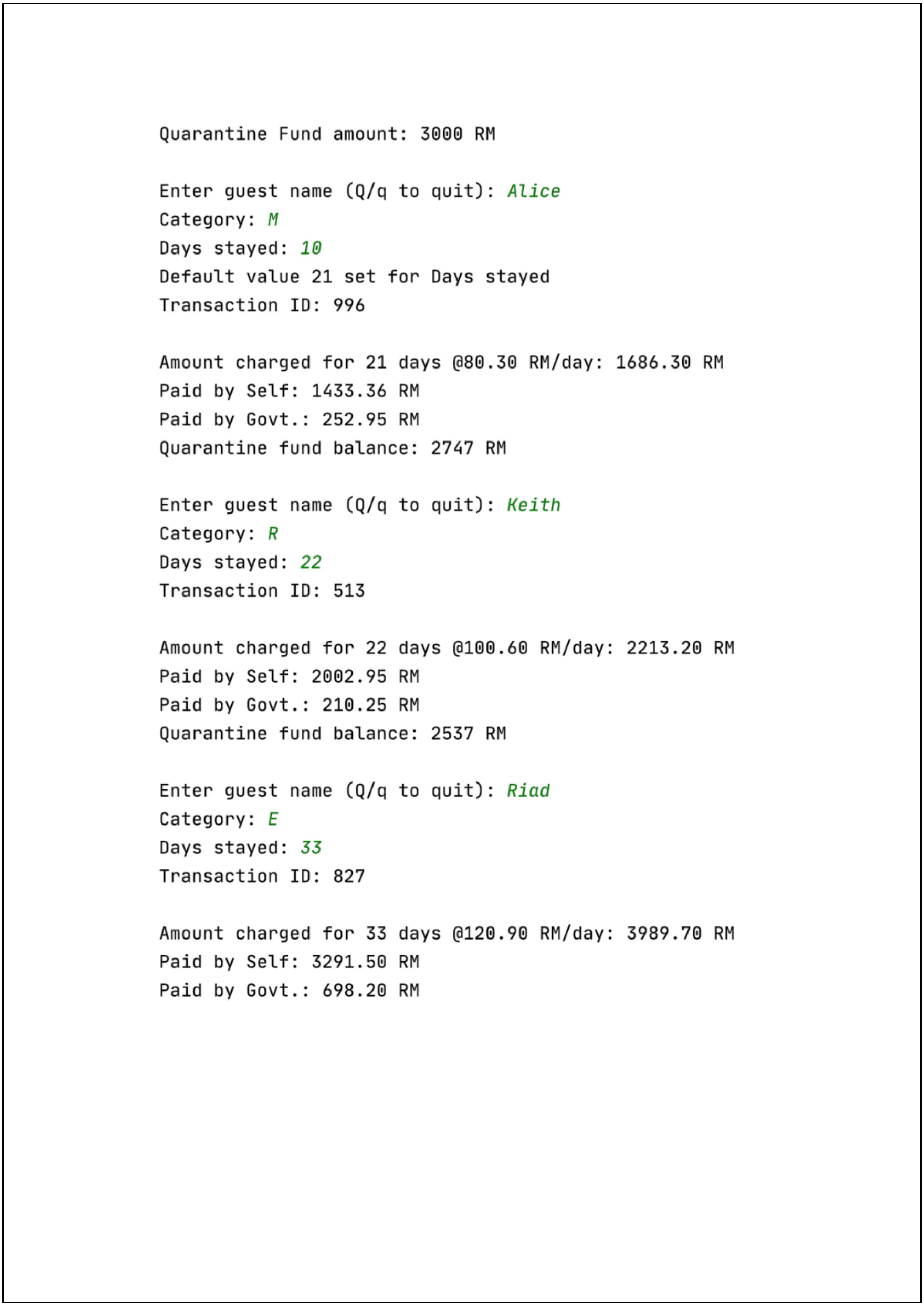
```
Enter guest name (Q/q to quit): Tan
Category: M
Days stayed: 25
Transaction ID: 097

Amount charged for 25 days @80.30 RM/day: 2007.50 RM
Paid by Self: 1575.89 RM
Paid by Govt.: 431.61 RM
Quarantine fund balance: 871 RM

Enter guest name (Q/q to quit): Shaan
Category: E
Days stayed: 40
Transaction ID: 287

Amount charged for 40 days @120.90 RM/day: 4836.00 RM
Paid by Self: 3989.70 RM
Paid by Govt.: 846.30 RM
Quarantine fund balance: 24 RM

Enter guest name (Q/q to quit): Rex
Category: m
Days stayed: 1
Default value 21 set for Days stayed

Insufficient balance in quarantine fund
Would you like to:
T. Top Up , Any other key to quit
Your choice: t
Top up amount: 1000
```



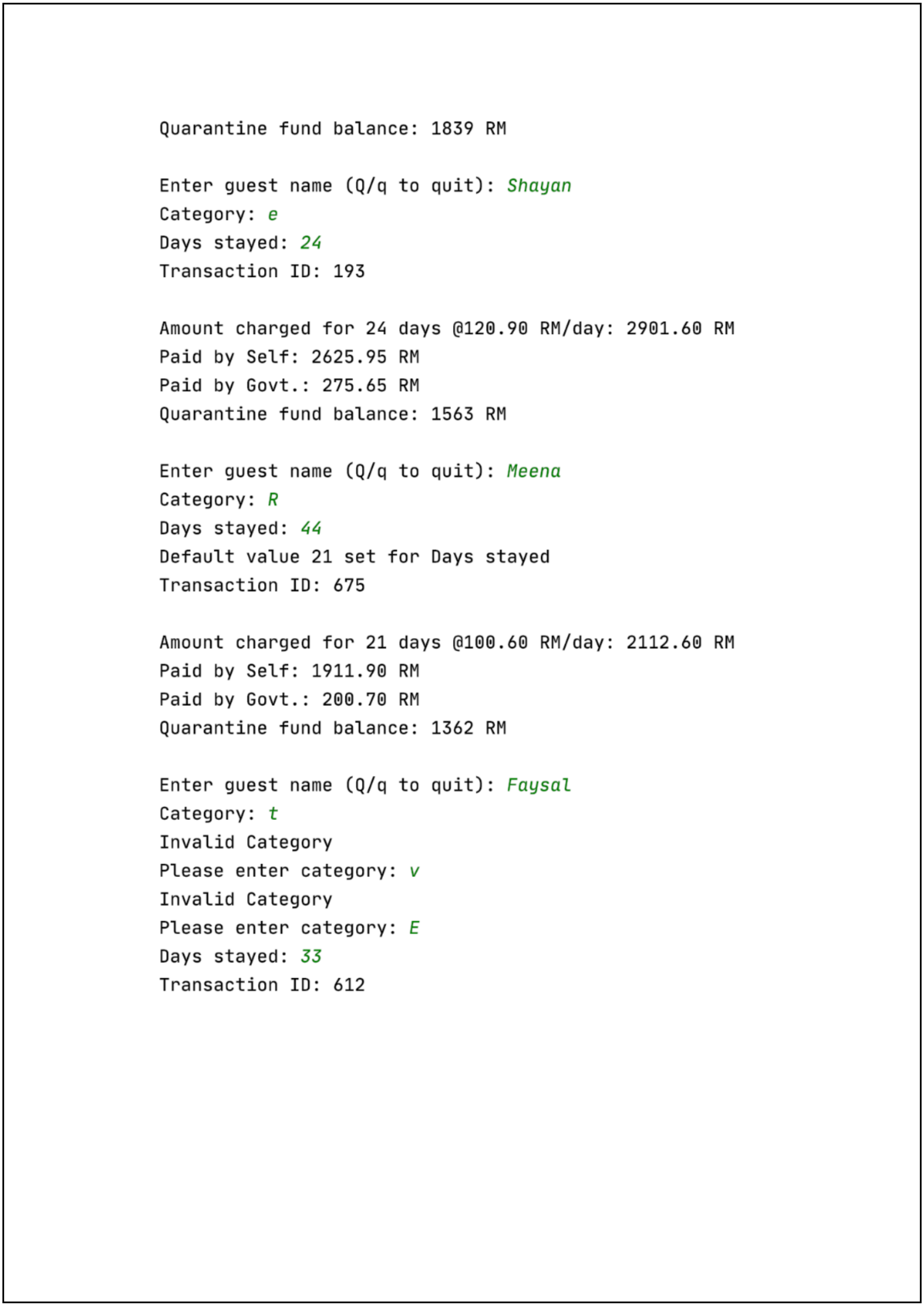
```
Quarantine fund balance: 1024 RM

Enter guest name (Q/q to quit): q
-------------------------------------------

Total Amount from 5 guests is 16136.60 RM
Highest amount paid 4836.00 RM by Shaan, is an Expat
Average amount from Malaysians: 2328.70 RM
Average amount from Foreigners: 3826.40 RM
Quarantine fund balance: 1024 RM

Process finished with exit code 0
```



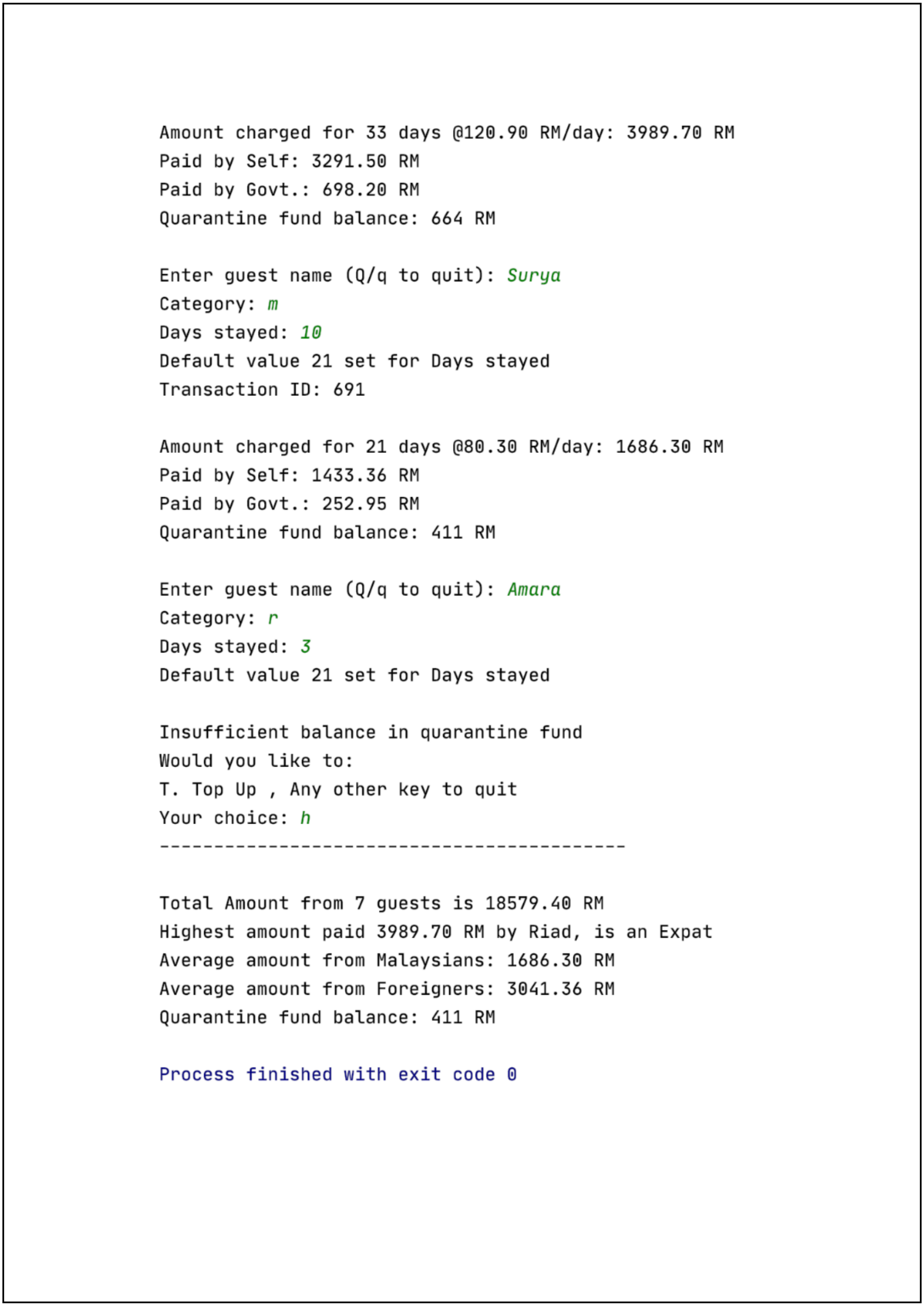
```
Quarantine Fund amount: 3000 RM

Enter guest name (Q/q to quit): q
No transactions

Process finished with exit code 0
```



```
Quarantine Fund amount: 3000 RM

Enter guest name (Q/q to quit): Alice
Category: M
Days stayed: 10
Default value 21 set for Days stayed
Transaction ID: 996

Amount charged for 21 days @80.30 RM/day: 1686.30 RM
Paid by Self: 1433.36 RM
Paid by Govt.: 252.95 RM
Quarantine fund balance: 2747 RM

Enter guest name (Q/q to quit): Keith
Category: R
Days stayed: 22
Transaction ID: 513

Amount charged for 22 days @100.60 RM/day: 2213.20 RM
Paid by Self: 2002.95 RM
Paid by Govt.: 210.25 RM
Quarantine fund balance: 2537 RM

Enter guest name (Q/q to quit): Riad
Category: E
Days stayed: 33
Transaction ID: 827

Amount charged for 33 days @120.90 RM/day: 3989.70 RM
Paid by Self: 3291.50 RM
Paid by Govt.: 698.20 RM
```



```
Quarantine fund balance: 1839 RM

Enter guest name (Q/q to quit): Shayan
Category: e
Days stayed: 24
Transaction ID: 193

Amount charged for 24 days @120.90 RM/day: 2901.60 RM
Paid by Self: 2625.95 RM
Paid by Govt.: 275.65 RM
Quarantine fund balance: 1563 RM

Enter guest name (Q/q to quit): Meena
Category: R
Days stayed: 44
Default value 21 set for Days stayed
Transaction ID: 675

Amount charged for 21 days @100.60 RM/day: 2112.60 RM
Paid by Self: 1911.90 RM
Paid by Govt.: 200.70 RM
Quarantine fund balance: 1362 RM

Enter guest name (Q/q to quit): Faysal
Category: t
Invalid Category
Please enter category: v
Invalid Category
Please enter category: E
Days stayed: 33
Transaction ID: 612
```



```
Amount charged for 33 days @120.90 RM/day: 3989.70 RM
Paid by Self: 3291.50 RM
Paid by Govt.: 698.20 RM
Quarantine fund balance: 664 RM

Enter guest name (Q/q to quit): Surya
Category: m
Days stayed: 10
Default value 21 set for Days stayed
Transaction ID: 691

Amount charged for 21 days @80.30 RM/day: 1686.30 RM
Paid by Self: 1433.36 RM
Paid by Govt.: 252.95 RM
Quarantine fund balance: 411 RM

Enter guest name (Q/q to quit): Amara
Category: r
Days stayed: 3
Default value 21 set for Days stayed

Insufficient balance in quarantine fund
Would you like to:
T. Top Up , Any other key to quit
Your choice: h
-------------------------------------------

Total Amount from 7 guests is 18579.40 RM
Highest amount paid 3989.70 RM by Riad, is an Expat
Average amount from Malaysians: 1686.30 RM
Average amount from Foreigners: 3041.36 RM
Quarantine fund balance: 411 RM

Process finished with exit code 0
```

Assignment1_E1900344

ORIGINALITY REPORT

8%	3%	0%	6%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to HELP UNIVERSITY Student Paper	4%
2	Submitted to Study Group Australia Student Paper	2%
3	tuanbean.wordpress.com Internet Source	1%
4	www.coursehero.com Internet Source	1%

Exclude quotes	On	Exclude matches	Off
Exclude bibliography	On		