UNIVERSITY OF WESTMINSTER

INFORMATICS INSTITUTE OF TECHNOLOGY

Informatics Institute of Technology

Department of Computing

Software Development II Coursework Report

Module	: 4COSC010C.3: Software Development II
Module Leader	: Mr Deshan Sumanathilaka
Date of submission	: 08/08/2022
Student ID	: <20211546> / <w1911221>
Student First Name	: Lahiru
Student Surname	: De Silva

"I confirm that I understand what plagiarism / collusion / contract cheating is and have read and understood the section on Assessment Offences in the Essential Information for Students. The work that I have submitted is entirely my own. Any work from other authors is duly referenced and acknowledged."

Name : Sellahandi Lahiru Rashmitha De Silva

Student ID : w1911221

Test Cases

	Test Case	Expected Result	Actual Result	Pass/Fail
1	Fuel Queue Initialized Correctly After program starts, 100 or VFQ	Displays 'empty' for all queues.	Displays 'empty' for all Queues.	Pass
2	Add passenger "Jane" to Queue 2 102 or ACQ Enter Queue: 2 Enter Name: Jane	Display 'Jane added to the queue 2 successfully"	Display 'Jane added to the queue 2 successfully"	Pass
3	101 or VEQ View all empty queues	Display all empty queues	Display all empty queues	Pass
4	103 or RCQ: Remove a customer from a Queue. (From a specific location) Enter Queue: 2 Enter customer number : 3	Successfully remove customer from q2 Position 3	Successfully remove customer from q2 Position 3	Pass
5	104 or PCQ: Remove a served customer. Do you want to remove customer from q1 :- Enter :- yes	Successfully remove customer from q1	Successfully remove customer from q1	Pass
6	105 or VCS: View Customers Sorted in alphabetical order	All customers show alphabetical order	All customers show alphabetical order	Pass
7	106 or SPD: Store Program Data into file	Successfully stored program data into file	Successfully stored program data into file	Pass
8	107 or LPD: Load Program Data from file.	Show all stored program data from file	Show all stored program data from file	Pass
9	108 or STK: View Remaining Fuel Stock	Show Remaining Fuel Stock	Show Remaining Fuel Stock	Pass
10	109 or AFS: Add Fuel Stock.	Successfully added	Successfully added	Pass

11	999 or EXT: Exit the Program.	Stop program	Stop program	Pass
12	110 or IFQ	print the income of each Fuel queue	print the income of each Fuel queue	Pass
13	Enter no of liters required ans :- string	This value is incorrect try agein	This value is incorrect try agein	Pass
14	Add to a waiting Queue If all Fuel Queue is full	Successfully added customer in waiting queue	Successfully added customer in waiting queue	Pass

Discussion

I wrote the program in such a way that all values given by use can be recognized, If a string value is given to a place where an integer value is requested, it will be indicated as wrong and the same value will be requested again

And it has always been shown that the amount of oil is low in the 500 liter Also, when getting the value for the alphabetical code, it is correct from such a house Here, for the sake of value, it is compared and displayed again in alphabetical order

In the class version, if all the queues are full, all custom arrivals will waiting Queue Also, as soon as there is a thin empty space in the queue, did you tell him to bring the customer to the last place of the queue and show him immediately?

It is mandatory to enter the first name when entering the customer details in the Class version

Code :

Array

```
import java.io.*;
import java.util.Scanner;

public class ArraysVersion {

    //For all fuel stock
    static int allFuelStock_liters = 6600;

    //For fuel details
    static int q1ServedCustomer = 0;
    static int q2ServedCustomer = 0;
    static int q3ServedCustomer = 0;

    //for identify int or string
    public static boolean NumbOrNot(String input){
        try {
            Integer.parseInt(input);
        }catch (NumberFormatException ex){
            return false;
        }
        return true;
    }



    public static void main(String[] args) {

        // All fuel center pumps arrays
        String[] FuelPumps_1 = new String[6];
        String[] FuelPumps_2 = new String[6];
        String[] FuelPumps_3 = new String[6];


        System.out.println("-------------------------------------");
        System.out.println("|   Fuel Queue Management System   |");
        System.out.println("-------------------------------------
\n");

        do {
            Scanner User = new Scanner(System.in);
            System.out.print("If you want to start this program
enter 'yes' :- ");
```

```
            String Input = User.nextLine();
            if (Input.equalsIgnoreCase("yes")) {
                System.out.println("~ ~ ~ WELCOME ~ ~ ~");
                System.out.println(" ");

                boolean menuBreak = true;
                do {
                    switch (MainMenu()) {
                        //View all Fuel Queues
                        case "100", "VFQ":
                            System.out.println(" > > > All Fuel
Queues < < < ");
                            System.out.println("\nnull =
empty\n\n");


                            System.out.println("Fuel Pumps 1");
                            PrintFuelQueues(FuelPumps_1);
                            System.out.println("\n\n");

                            System.out.println("Fuel Pumps 2");
                            PrintFuelQueues(FuelPumps_2);
                            System.out.println("\n\n");

                            System.out.println("Fuel Pumps 3");
                            PrintFuelQueues(FuelPumps_3);
                            System.out.println("\n\n");

                            int VFQ = ext_or_cont_loop();
                            if(VFQ == 2){
                                menuBreak = false;
                            }
                            break;

                        //View all Empty Queues
                        case "101", "VEQ":
                            System.out.println("> > > All Empty
Queues < < <");
                            System.out.println("\nnull =
empty\n\n");


                            System.out.println("Fuel Pumps 1");
                            AllEmptyQueues(FuelPumps_1);
                            System.out.println("\n\n");

                            System.out.println("Fuel Pumps 2");
                            AllEmptyQueues(FuelPumps_2);
                            System.out.println("\n\n");

                            System.out.println("Fuel Pumps 3");
```

```
                            AllEmptyQueues(FuelPumps_3);
                            System.out.println("\n\n");

                            int VEQ = ext_or_cont_loop();
                            if(VEQ == 2){
                                menuBreak = false;
                            }
                            break;

                        //Add customer to a Queue
                        case "102", "ACQ":
                            System.out.println("> > > Add customer
to a Queue < < <\n\n");

                            System.out.println("Fuel Pumps 1");
                            AllEmptyQueues(FuelPumps_1);
                            System.out.println("\n\n");

                            System.out.println("Fuel Pumps 2");
                            AllEmptyQueues(FuelPumps_2);
                            System.out.println("\n\n");

                            System.out.println("Fuel Pumps 3");
                            AllEmptyQueues(FuelPumps_3);
                            System.out.println("\n\n\n");

                            System.out.println(">>> The above shows
all empty queues");
                            System.out.println(">>> You can add
people to any empty queue");
                            System.out.println(">>> If you want to
stop adding customers ENTER :- 'Finish'\n");
                            System.out.println("Pump 1 queue");
                            AddCustomers(FuelPumps_1);
                            System.out.println("\n");
                            System.out.println("Pump 2 queue");
                            AddCustomers(FuelPumps_2);
                            System.out.println("\n");
                            System.out.println("Pump 3 queue");
                            AddCustomers(FuelPumps_3);
                            System.out.println("\n\n");

                            int ACQ = ext_or_cont_loop();
                            if(ACQ == 2){
                                menuBreak = false;
                            }
                            break;

                        //Remove a customer from a Queue (From a
```

```
specific location)
                        case "103", "RCQ":
                            System.out.println("> > > Remove a
customer from a specific location < < <\n\n");

                            System.out.println("Fuel Pumps 1");
                            PrintFuelQueues(FuelPumps_1);
                            System.out.println("\n\n");

                            System.out.println("Fuel Pumps 2");
                            PrintFuelQueues(FuelPumps_2);
                            System.out.println("\n\n");

                            System.out.println("Fuel Pumps 3");
                            PrintFuelQueues(FuelPumps_3);
                            System.out.println("\n\n\n");


RemoveCustomerSpecificLocation(FuelPumps_1,FuelPumps_2,FuelPumps_3)
;
                            System.out.println("\n\n");

                            int RCQ = ext_or_cont_loop();
                            if(RCQ == 2){
                                menuBreak = false;
                            }
                            break;

                        //Remove a served customer
                        case "104", "PCQ":
                            System.out.println("> > > Remove a
customer < < <\n\n");


RemoveServedCustomer(FuelPumps_1,FuelPumps_2,FuelPumps_3);

                            int PCQ = ext_or_cont_loop();
                            if(PCQ == 2){
                                menuBreak = false;
                            }
                            break;

                        //View Customers Sorted in alphabetical
order
                        case "105", "VCS":
                            System.out.println("> > > All customers
in alphabetical order < < <\n\n");

                            System.out.println("Fuel Pumps 1");
```

```
                            AtoZ(FuelPumps_1);
                            System.out.println("\n\n");

                            System.out.println("Fuel Pumps 2");
                            AtoZ(FuelPumps_2);
                            System.out.println("\n\n");

                            System.out.println("Fuel Pumps 3");
                            AtoZ(FuelPumps_3);
                            System.out.println("\n\n");

                            int VCS = ext_or_cont_loop();
                            if(VCS == 2){
                                menuBreak = false;
                            }
                            break;

                        //Store Program Data into file
                        case "106", "SPD":

StoreDataInToFile(FuelPumps_1,FuelPumps_2,FuelPumps_3);
                            break;

                        //Load Program Data from file
                        case "107", "LPD":
                            LoadDateFromFile();

                            int LPD = ext_or_cont_loop();
                            if(LPD == 2){
                                menuBreak = false;
                            }
                            break;

                        //View Remaining Fuel Stock
                        case "108", "STK":
                            System.out.println("> > > View
Remaining Fuel Stock < < <\n\n");

                            System.out.println("Remaining fuel
ftock :- "+allFuelStock_liters+" liters\n\n");

                            int STK = ext_or_cont_loop();
                            if(STK == 2){
                                menuBreak = false;
                            }
                            break;

                        //Add Fuel Stock
                        case "109", "AFS":
```

```
                                System.out.println("> > > Add Fuel
Stock < < <\n\n");
                                Scanner AFSuser = new
Scanner(System.in);
                                boolean AFSloopbreak = true;
                                while (AFSloopbreak){
                                    System.out.print("Give the amount
of fuel you are refilling in litres :- ");
                                    String AFSinput =
AFSuser.nextLine();
                                    if(NumbOrNot(AFSinput)){
                                        if (Integer.valueOf(AFSinput)
<= 0){
                                            System.out.println("This
Value is Incorrect\n");
                                        }else {
                                            allFuelStock_liters +=
Integer.valueOf(AFSinput);
                                            System.out.println("\n > >
> Successfully added "+AFSinput+" litres");
                                            System.out.println(" > > >
Now Remaining Fuel Stock is "+allFuelStock_liters+" litres\n\n\n");
                                            AFSloopbreak = false;
                                        }
                                    }else {
                                        System.out.println("Enter
integer value .....!\n");
                                    }
                                }

                                int AFS = ext_or_cont_loop();
                                if(AFS == 2){
                                    menuBreak = false;
                                }
                                break;

                            //Exit the Program
                            case "999", "EXT":
                                System.out.println("~ ~ ~ It's pleasure
to serve you ~ ~ ~\n\n");
                                for (int i = 0;i <= 75; i++){
                                    System.out.print("- ");
                                }
                                menuBreak = false;
                                break;
                            default:
                                System.out.println("~ ~ ~ Incorrect
answer try again .............!\n\n");
                                for (int i = 0;i <= 75; i++){
```

```
                                System.out.print("- ");
                            }
                            System.out.println("\n\n");
                        }
                    }while (menuBreak);
                    break;

                } else {
                    System.out.println("~ ~ ~ Try again.....!");
                }
            }while (true);

        }

        //main menu
        public static String MainMenu(){

            System.out.println("< < < < *** > > > >");
            System.out.println("|    Main  Menu   |");
            System.out.println("< < < < *** > > > >\n");

            System.out.println("100 or VFQ: View all Fuel Queues");
            System.out.println("101 or VEQ: View all Empty Queues");
            System.out.println("102 or ACQ: Add customer to a Queue");
            System.out.println("103 or RCQ: Remove a customer from a
Queue (From a specific location)");
            System.out.println("104 or PCQ: Remove a served customer");
            System.out.println("105 or VCS: View Customers Sorted in
alphabetical order");
            System.out.println("106 or SPD: Store Program Data into
file");
            System.out.println("107 or LPD: Load Program Data from
file");
            System.out.println("108 or STK: View Remaining Fuel
Stock");
            System.out.println("109 or AFS: Add Fuel Stock");
            System.out.println("999 or EXT: Exit the Program\n");

            if (allFuelStock_liters <= 500){
                System.out.println("\n > > > > > Fuel Stock Level is
low and it is "+allFuelStock_liters+" liters\n\n");
            }

            System.out.print("Enter your selection :- ");
            Scanner UserSelection = new Scanner(System.in);
            String US = UserSelection.nextLine();
            System.out.println(" ");
            for (int i = 0;i <= 75; i++){
                System.out.print("- ");
```

```
        }
        System.out.println("\n\n");
        return US;
    }



    //Ext or Cont Loop
    public static int ext_or_cont_loop(){
        Scanner UserSelection = new Scanner(System.in);
        boolean vfqBreak = true;
        int output = 0;
        while (vfqBreak) {
            System.out.println(" 1) Back to Main Menu");
            System.out.println(" 2) Exit the Program");
            System.out.print("Enter answer :- ");
            String  ex_or_back = UserSelection.nextLine();
            System.out.println("\n");

            if (ex_or_back.equalsIgnoreCase("1")) {
                for (int i = 0; i <= 75; i++) {
                    System.out.print("- ");
                }
                System.out.println("\n\n");
                output *= 0;
                output += 1;
                vfqBreak = false;

            } else if (ex_or_back.equalsIgnoreCase("2")) {
                System.out.println("~ ~ ~ It's a pleasure to serve
you ~ ~ ~");
                output *= 0;
                output += 2;
                vfqBreak = false;

            } else {
                System.out.println("Unexpected answer retry
.............!");
            }
        }
        return output;
    }



    //100 or VFQ: View all Fuel Queues
    public static void PrintFuelQueues(String[] FuelCenter){
        System.out.println("          1             2
3             4             5             6              >>>Queue
```

```
numbers");
        for (int i = 0; i < FuelCenter.length; i++){
            System.out.print("          "+FuelCenter[i]);
        }
    }



    //101 or VEQ: View all Empty Queues
    public static void  AllEmptyQueues(String[] VEQ){
        String laststr = VEQ[5];
        if (laststr == null){
            PrintFuelQueues(VEQ);
        }else {
            System.out.println("This queues is full");
        }
    }



    //102 or ACQ: Add customer to a Queue
    public static void AddCustomers(String[] ACQ){
        Scanner user = new Scanner(System.in);

        //Check it out last null or not
        if (ACQ[5] == null) {
            for (int i = 0; i < 6; i++) {

                boolean loopbreack = true;
                while (ACQ[i] == null && loopbreack){
                    System.out.print("Add new customer to queues
position " + (i + 1) + " :- ");
                    String input = user.nextLine();
                    if (input.equalsIgnoreCase("finish")) {
                        System.out.println("\n");
                        int ii = 5-i;
                        i=i+ii;
                        loopbreack = false;
                    } else {
                        ACQ[i] = input;
                        System.out.println("Queues position " + (i
+ 1) + "  >>> Successfully added customer '" + ACQ[i] + "'\n");
                    }
                }
            }
        }
        //if is null Its meaning that queue is full
        else {
            System.out.println("Queue is full !");
```

```
        }
    }



    //103 or RCQ: Remove a customer from a Queue (From a specific
location)
    public static void RemoveCustomerSpecificLocation(String[] q1,
String[] q2, String[] q3){
        Scanner user = new Scanner(System.in);
        boolean loopbreak = true;
        while (loopbreak){
            System.out.print("Do you want to remove customer from
which queues :- ");
            String input = user.nextLine();
            if (NumbOrNot(input)) {
                int uinp = Integer.valueOf(input);
                if (0 < uinp && uinp < 4) {
                    switch (uinp) {
                        case 1:
                            boolean loopbreak2 = true;
                            while (loopbreak2) {
                                System.out.print("And customer
queues number :- ");
                                String input2 = user.nextLine();
                                if (NumbOrNot(input2)) {
                                    int uinp2 =
Integer.valueOf(input2);
                                    int forsout = uinp2;
                                    if (0 < uinp2 && uinp2 <7) {
                                        String[] Q_Blank = new
String[1];
                                        int removeCustomer = uinp2
-1;
                                        for (int i=uinp2;i<6;i++){
                                            q1[removeCustomer] =
q1[uinp2];
                                            removeCustomer++;
                                            uinp2++;
                                        }
                                        q1[5] = Q_Blank[0];

System.out.println("\nSuccessfully remove customer form fuel queue
1 position "+forsout);
                                        loopbreak2 = false;
                                    }else {

System.out.println("Unexpected answer retry .............!\n");
                                    }
```

```
                                } else {
                                    System.out.println("Unexpected
answer retry .....!\n");
                                }
                            }
                            loopbreak = false;
                            break;
                        case 2:
                            boolean loopbreak3 = true;
                            while (loopbreak3) {
                                System.out.print("And customer
queues number :- ");
                                String input3 = user.nextLine();
                                if (NumbOrNot(input3)) {
                                    int uinp3 =
Integer.valueOf(input3);
                                    int forsout2 = uinp3;
                                    if (0 < uinp3 && uinp3 <7) {
                                        String[] Q_Blank2 = new
String[1];
                                        int removeCustomer2 = uinp3
-1;
                                        for (int i=uinp3;i<6;i++){
                                            q2[removeCustomer2] =
q2[uinp3];
                                            removeCustomer2++;
                                            uinp3++;
                                        }
                                        q2[5] = Q_Blank2[0];

System.out.println("\nSuccessfully remove customer form fuel queue
1 position "+forsout2);
                                        loopbreak3 = false;
                                    }else {

System.out.println("Unexpected answer retry .....!\n");
                                    }
                                } else {
                                    System.out.println("Unexpected
answer retry .....!\n");
                                }
                            }
                            loopbreak = false;
                            break;
                        case 3:
                            boolean loopbreak4 = true;
                            while (loopbreak4) {
                                System.out.print("And customer
queues number :- ");
```

```
                                String input4 = user.nextLine();
                                if (NumbOrNot(input4)) {
                                    int uinp4 =
Integer.valueOf(input4);
                                    int forsout3 = uinp4;
                                    if (0 < uinp4 && uinp4 <7) {
                                        String[] Q_Blank3 = new
String[1];
                                        int removeCustomer3 = uinp4
-1;
                                        for (int i=uinp4;i<6;i++){
                                            q3[removeCustomer3] =
q1[uinp4];
                                            removeCustomer3++;
                                            uinp4++;
                                        }
                                        q1[5] = Q_Blank3[0];

System.out.println("\nSuccessfully remove customer form fuel queue
1 position "+forsout3);
                                        loopbreak4 = false;
                                    }else {

System.out.println("Unexpected answer retry .....!\n");
                                    }
                                } else {
                                    System.out.println("Unexpected
answer retry .....!\n");
                                }
                            }
                            loopbreak = false;
                            break;
                        }
                    } else {
                        System.out.println("Unexpected answer retry
.....!\n");
                    }
                }else {
                    System.out.println("Unexpected answer retry
.....!\n");
                }
            }
        }



    //104 or PCQ: Remove a served customer
    public static void RemoveServedCustomer(String[] q1,String[]
q2,String[] q3){
```

```
        Scanner PCQuser = new Scanner(System.in);
        System.out.println("Fuel Pumps 1");
        if(q1[0] == null){
            System.out.println("This queue is empty .....!");
        }else {
            boolean PCQloopbreak = true;
            while (PCQloopbreak){
                System.out.print("Do you want to remove served
customer from this queue ? (yes/no) :- ");
                String PCQinput = PCQuser.nextLine();
                if (PCQinput.equalsIgnoreCase("yes")){
                    allFuelStock_liters -= 10;
                    q1ServedCustomer += 1;
                    String[] Q_blank1 = new String[1];
                    for (int i=0;i<5;i++){
                        q1[i] = q1[(i+1)];
                    }
                    q1[5] = Q_blank1[0];
                    System.out.println("Successfully remove served
customer");
                    PCQloopbreak = false;
                }
                else if (PCQinput.equalsIgnoreCase("no")) {
                    System.out.println(">>> Skip..........!");
                    PCQloopbreak = false;
                } else {
                    System.out.println("Unexpected answer retry
............!\n");
                }
            }
        }
        System.out.println("\n\n");
        System.out.println("Fuel Pumps 2");
        if(q2[0] == null){
            System.out.println("This queue is empty .....!");
        }else {
            boolean PCQloopbreak = true;
            while (PCQloopbreak){
                System.out.print("Do you want to remove served
customer from this queue ? (yes/no) :- ");
                String PCQinput = PCQuser.nextLine();
                if (PCQinput.equalsIgnoreCase("yes")){
                    allFuelStock_liters -= 10;
                    q2ServedCustomer += 1;
                    String[] Q_blank2 = new String[1];
                    for (int i=0;i<5;i++){
                        q2[i] = q2[(i+1)];
                    }
```

```
                    q2[5] = Q_blank2[0];
                    System.out.println("Successfully remove served
customer");
                    PCQloopbreak = false;
                }
                else if (PCQinput.equalsIgnoreCase("no")) {
                    System.out.println(">>> Skip.........!");
                    PCQloopbreak = false;
                } else {
                    System.out.println("Unexpected answer retry
............!\n");
                }
            }
        }
        System.out.println("\n\n");
        System.out.println("Fuel Pumps 3");
        if(q3[0] == null){
            System.out.println("This queue is empty .....!");
        }else {
            boolean PCQloopbreak = true;
            while (PCQloopbreak){
                System.out.print("Do you want to remove served
customer from this queue ? (yes/no) :- ");
                String PCQinput = PCQuser.nextLine();
                if (PCQinput.equalsIgnoreCase("yes")){
                    q3ServedCustomer += 1;
                    allFuelStock_liters -= 10;
                    String[] Q_blank3 = new String[1];
                    for (int i=0;i<5;i++){
                        q3[i] = q3[(i+1)];
                    }
                    q3[5] = Q_blank3[0];
                    System.out.println("Successfully remove served
customer");
                    PCQloopbreak = false;
                }
                else if (PCQinput.equalsIgnoreCase("no")) {
                    System.out.println(">>> Skip.........!");
                    PCQloopbreak = false;
                } else {
                    System.out.println("Unexpected answer retry
............!\n");
                }
            }
        }
        System.out.println("\n\n");
    }
```

```
    //105 or VCS: View Customers Sorted in alphabetical order
    public static void AtoZ(String[] VCS){

        if(VCS[0] != null) {

            //Counting null count
            int null_count = 0;
            for (int i = 0; i < VCS.length; i++) {
                if (VCS[i] == null) {
                    null_count = null_count + 1;
                }
            }

            //For remove null elements
            String[] VCS2 = new String[VCS.length - null_count];
            for (int i = 0; i < VCS.length; i++) {
                if (VCS[i] != null) {
                    VCS2[i] = VCS[i];
                }
            }

            char[][] strToChar2D = new char[VCS2.length][];
            char[] all_1st_Elements = new char[VCS2.length];
            char[] withOutSameElements = new char[VCS2.length];
            char[][] AtoZ = new char[VCS2.length][];
            int x = 0;
            int y = 0;
            int z = 0;
            char temp;

            //Converting all string array elements to char array
            for (int i = 0; i < VCS2.length; i++) {
                strToChar2D[i] = VCS2[i].toCharArray();
            }

            //Selecting the first element from all the elements in
the char array
            for (int i = 0; i < VCS2.length; i++) {
                all_1st_Elements[i] = strToChar2D[i][0];
            }

            //Converting all elements to A to Z order
            for (int i = 0; i < VCS2.length; i++) {
                for (int ii = i + 1; ii < VCS2.length; ii++) {
                    if (all_1st_Elements[i] > all_1st_Elements[ii])
{
                        temp = all_1st_Elements[i];
                        all_1st_Elements[i] = all_1st_Elements[ii];
```

```
                        all_1st_Elements[ii] = temp;
                    }
                }
            }

            //Counting similar elements
            for (int i = 0; i < all_1st_Elements.length - 1; i++) {
                if (all_1st_Elements[i] != all_1st_Elements[i + 1])
{
                    withOutSameElements[x++] = all_1st_Elements[i];
                } else {
                    y = y + 1;
                }
            }
            withOutSameElements[x++] =
all_1st_Elements[all_1st_Elements.length - 1];

            char[] withOutSameElementsAndSize = new
char[VCS2.length - y];

            //Append without all element and size in new array
            for (int i = 0; i < withOutSameElementsAndSize.length;
i++) {
                withOutSameElementsAndSize[i] =
withOutSameElements[i];
            }

            //Append A to Z all elements in new array
            for (int i = 0; i < withOutSameElementsAndSize.length;
i++) {
                for (int ii = 0; ii < VCS2.length; ii++) {
                    if (withOutSameElementsAndSize[i] ==
strToChar2D[ii][0]) {
                        AtoZ[z++] = strToChar2D[ii];
                    }
                }
            }

            //Printing all elements in A to Z oder
            for (int i = 0; i < AtoZ.length; i++) {
                System.out.print(" "+(i+1)+") ");
                System.out.println(AtoZ[i]);
            }
        }else {
            System.out.println(">>> This queues Empty .........!");
        }
    }
```

```
    //106 or SPD: Store Program Data into file
    public static void StoreDataInToFile(String[] q1, String[] q2,
String[] q3){

        File fuelQ = new File("Fuel_Queues_Deta.txt");

        String FQ1 = "\n                    < < < < < < <***> > > >
> > >\n";
        String FQ2 = "                    |     Fuel Station Data
|\n";
        String FQ3 = "                   < < < < < < <***> > > > >
> >\n";
        String total = "\n\nTotal amount of oil currently remaining
:- ";
        String FQCount = "\n\n\n\n     Fuel Queue ";
        String AOCD = "\nAmount of oil currently delivered :- ";
        String NPCRO = "\nNumber of people currently receiving oil
:- ";
        String PCWQ = "\nPeople currently waiting in queue";

        try{
            fuelQ.createNewFile();
            BufferedWriter write = new BufferedWriter(new
FileWriter("Fuel_Queues_Deta.txt"));

write.write(FQ1+FQ2+FQ3+total+(allFuelStock_liters)+FQCount+"1"+AOC
D+(q1ServedCustomer*10)+NPCRO+(q1ServedCustomer)+PCWQ);
            for (int i = 0; i < q1.length; i++){
                if(q1[0]==null){
                    write.write(" :- This queue is empty .....!");
                    break;
                }else if (q1[i]==null){
                    break;
                }else {
                    write.write("\n");
                    write.write((i+1)+") "+q1[i]);
                }
            }

write.write(FQCount+"2"+AOCD+(q2ServedCustomer*10)+NPCRO+(q2ServedC
ustomer)+PCWQ);
            for (int i = 0; i < q2.length; i++){
                if(q2[0]==null){
                    write.write(" :- This queue is empty .....!");
                    break;
                }else if (q2[i]==null){
                    break;
                }else {
                    write.write("\n");
```

```
                    write.write((i+1)+") "+q2[i]);
                }
            }

write.write(FQCount+"3"+AOCD+(q3ServedCustomer*10)+NPCRO+(q3ServedC
ustomer)+PCWQ);
            for (int i = 0; i < q3.length; i++){
                if(q3[0]==null){
                    write.write(" :- This queue is empty .....!");
                    break;
                }else if (q3[i]==null){
                    break;
                }else {
                    write.write("\n");
                    write.write((i+1)+") "+q3[i]);
                }
            }
            System.out.println("Successfully Store Program Data
Into File .....!");
            System.out.println("\n");
            for (int i = 0;i <= 75; i++){
                System.out.print("- ");
            }
            System.out.println("\n\n");
            write.close();
        }catch (Exception e){
            System.out.println(e);
        }
    }


    //107 or LPD: Load Program Data from file
    public static void LoadDateFromFile(){
        File fuelQ = new File("Fuel_Queues_Deta.txt");

        try {
            BufferedReader reader = new BufferedReader(new
FileReader(fuelQ));

            String str;
            while ((str = reader.readLine()) != null){
                System.out.println(str);
            }
            System.out.println("\n");

        }catch (Exception e){
            System.out.println(e);
        }
    }
```

```
}
```

class

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

public class ClassVersion {

    //For all fuel stock
    static int allFuelStock_liters = 6600;

    //for count Served Customer
    static int q1ServedCustomer = 0;
    static int q2ServedCustomer = 0;
    static int q3ServedCustomer = 0;
    static int q4ServedCustomer = 0;
    static int q5ServedCustomer = 0;

    //for count Served Liter
    static int q1ServedLiter = 0;
    static int q2ServedLiter = 0;
    static int q3ServedLiter = 0;
    static int q4ServedLiter = 0;
    static int q5ServedLiter = 0;

    //For waiting Queue
    static ArrayList<String> WQFirstName = new ArrayList<>();
    static ArrayList<String> WQSecondName = new ArrayList<>();
    static ArrayList<String> WQVehicleNo = new ArrayList<>();
    static ArrayList<Integer> WQLitersRequired = new ArrayList<>();

    //for identify int or string
    public static boolean NumbOrNot(String input){
        try {
            Integer.parseInt(input);
        }catch (NumberFormatException ex){
            return false;
        }
        return true;
    }



    public static void main(String[] args) {
```

```
        //For fuel details
        FuelQueue Queue1 = new FuelQueue();
        FuelQueue Queue2 = new FuelQueue();
        FuelQueue Queue3 = new FuelQueue();
        FuelQueue Queue4 = new FuelQueue();
        FuelQueue Queue5 = new FuelQueue();

        FuelQueue[] Queue =
{(Queue1),(Queue2),(Queue3),(Queue4),(Queue5)};

        //Fuel Queue Management System
        System.out.println("-------------------------------------");
        System.out.println("|   Fuel Queue Management System   |");
        System.out.println("-------------------------------------
\n");

        do {
            Scanner User = new Scanner(System.in);
            System.out.print("If you want to start this program
enter 'yes' :- ");
            String Input = User.nextLine();
            if (Input.equalsIgnoreCase("yes")) {
                System.out.println("~ ~ ~ WELCOME ~ ~ ~");
                System.out.println(" ");

                boolean menuBreak = true;
                do {
                    switch (MainMenu()) {
                        //View all Fuel Queues
                        case "100", "VFQ":
                            System.out.println(" > > > All Fuel
Queues < < < ");

                            ViewAllFuelQueues(Queue);

                            int VFQ = ext_or_cont_loop();
                            if(VFQ == 2){
                                menuBreak = false;
                            }
                            break;

                        //View all Empty Queues
                        case "101", "VEQ":
                            System.out.println("> > > All Empty
Queues < < <");

                            ViewAllEmptyQueues(Queue);
                            System.out.println("\n\n");
```

```
                            int VEQ = ext_or_cont_loop();
                            if(VEQ == 2){
                                menuBreak = false;
                            }
                            break;

                        //Add customer to a Queue
                        case "102", "ACQ":
                            System.out.println("> > > Add customer
to a Queue < < <\n\n");

                            System.out.println(">>> Your customer
will be added to the location with the shortest queue\n");

                            System.out.println("Enter customer
details");
                            AddCustomerToAQueue(Queue);
                            System.out.println("\n\n");

                            int ACQ = ext_or_cont_loop();
                            if(ACQ == 2){
                                menuBreak = false;
                            }
                            break;

                        //Remove a customer from a Queue (From a
specific location)
                        case "103", "RCQ":
                            System.out.println("> > > Remove a
customer from a specific location < < <\n\n");

                            AllEmptyQueues(Queue);

RemoveCustomerFromSpecificLocation(Queue);

                            int RCQ = ext_or_cont_loop();
                            if(RCQ == 2){
                                menuBreak = false;
                            }
                            break;

                        //Remove a served customer
                        case "104", "PCQ":
                            System.out.println("> > > Remove a
customer < < <\n\n");

                            RemoveServedCustomer(Queue);
```

```
                            int PCQ = ext_or_cont_loop();
                            if(PCQ == 2){
                                menuBreak = false;
                            }
                            break;

                        //View Customers Sorted in alphabetical
order
                        case "105", "VCS":
                            System.out.println("> > > All customers
in alphabetical order < < <\n\n");

                            System.out.println("Fuel Pumps 1");
                            AtoZBefore(Queue,0);
                            System.out.println("\n\n");

                            System.out.println("Fuel Pumps 2");
                            AtoZBefore(Queue,1);
                            System.out.println("\n\n");

                            System.out.println("Fuel Pumps 3");
                            AtoZBefore(Queue,2);
                            System.out.println("\n\n");

                            System.out.println("Fuel Pumps 4");
                            AtoZBefore(Queue,3);
                            System.out.println("\n\n");

                            System.out.println("Fuel Pumps 5");
                            AtoZBefore(Queue,4);
                            System.out.println("\n\n");

                            int VCS = ext_or_cont_loop();
                            if(VCS == 2){
                                menuBreak = false;
                            }
                            break;

                        //Store Program Data into file
                        case "106", "SPD":
                            StoreDataInToFile(Queue);
                            break;

                        //Load Program Data from file
                        case "107", "LPD":
                            LoadDateFromFile();

                            int LPD = ext_or_cont_loop();
                            if(LPD == 2){
```

```
                                menuBreak = false;
                            }
                            break;

                        //View Remaining Fuel Stock
                        case "108", "STK":
                            System.out.println("> > > View
Remaining Fuel Stock < < <\n\n");

                            System.out.println("Remaining fuel
ftock :- "+allFuelStock_liters+" liters\n\n");

                            int STK = ext_or_cont_loop();
                            if(STK == 2){
                                menuBreak = false;
                            }
                            break;

                        //Add Fuel Stock
                        case "109", "AFS":
                            System.out.println("> > > Add Fuel
Stock < < <\n\n");
                            Scanner AFSuser = new
Scanner(System.in);
                            boolean AFSloopbreak = true;
                            while (AFSloopbreak){
                                System.out.print("Give the amount
of fuel you are refilling in litres :- ");
                                String AFSinput =
AFSuser.nextLine();
                                if(NumbOrNot(AFSinput)){
                                    if (Integer.valueOf(AFSinput)
<= 0){
                                        System.out.println("This
Value is Incorrect\n");
                                    }else {
                                        allFuelStock_liters +=
Integer.valueOf(AFSinput);
                                        System.out.println("\n > >
> Successfully added "+AFSinput+" litres");
                                        System.out.println(" > > >
Now Remaining Fuel Stock is "+allFuelStock_liters+" litres\n\n\n");
                                        AFSloopbreak = false;
                                    }
                                }else {
                                    System.out.println("Enter
integer value .....!\n");
                                }
                            }
```

```
                            int AFS = ext_or_cont_loop();
                            if(AFS == 2){
                                menuBreak = false;
                            }
                            break;

                        //income of each Fuel queue
                        case "110", "IFQ":
                            System.out.println("> > > Income of
each Fuel Queue < < <\n\n");

                            IncomeFuelQueues();

                            int IFQ = ext_or_cont_loop();
                            if(IFQ == 2){
                                menuBreak = false;
                            }
                            break;

                        //Exit the Program
                        case "999", "EXT":
                            System.out.println("~ ~ ~ It's pleasure
to serve you ~ ~ ~\n\n");
                            for (int i = 0;i <= 75; i++){
                                System.out.print("- ");
                            }
                            menuBreak = false;
                            break;
                        default:
                            System.out.println("~ ~ ~ Incorrect
answer try again .....!\n\n");
                            for (int i = 0;i <= 75; i++){
                                System.out.print("- ");
                            }
                            System.out.println("\n\n");
                    }
                }while (menuBreak);
                break;

            } else {
                System.out.println("~ ~ ~ Try again.....!");
            }
        }while (true);

    }

    //Main Menu
    public static String MainMenu(){
```

```
        System.out.println("< < < < *** > > > >");
        System.out.println("|    Main  Menu    |");
        System.out.println("< < < < *** > > > >\n");

        System.out.println("100 or VFQ: View all Fuel Queues");
        System.out.println("101 or VEQ: View all Empty Queues");
        System.out.println("102 or ACQ: Add customer to a Queue");
        System.out.println("103 or RCQ: Remove a customer from a
Queue (From a specific location)");
        System.out.println("104 or PCQ: Remove a served customer");
        System.out.println("105 or VCS: View Customers Sorted in
alphabetical order");
        System.out.println("106 or SPD: Store Program Data into
file");
        System.out.println("107 or LPD: Load Program Data from
file");
        System.out.println("108 or STK: View Remaining Fuel
Stock");
        System.out.println("109 or AFS: Add Fuel Stock");
        System.out.println("110 or IFQ: Income of each Fuel
Queue");
        System.out.println("999 or EXT: Exit the Program\n");

        if (allFuelStock_liters <= 500){
            System.out.println("\n > > > > > Fuel Stock Level is
low and it is "+allFuelStock_liters+" liters\n\n");
        }

        System.out.print("Enter your selection :- ");
        Scanner UserSelection = new Scanner(System.in);
        String US = UserSelection.nextLine();
        System.out.println(" ");
        for (int i = 0;i <= 75; i++){
            System.out.print("- ");
        }
        System.out.println("\n\n");
        return US;
    }



    //Ext or Cont Loop
    public static int ext_or_cont_loop(){
        Scanner UserSelection = new Scanner(System.in);
        boolean vfqBreak = true;
        int output = 0;
        while (vfqBreak) {
            System.out.println(" 1) Back to Main Menu");
```

```
            System.out.println(" 2) Exit the Program");
            System.out.print("Enter answer :- ");
            String  ex_or_back = UserSelection.nextLine();
            System.out.println("\n");

            if (ex_or_back.equalsIgnoreCase("1")) {
                for (int i = 0; i <= 75; i++) {
                    System.out.print("- ");
                }
                System.out.println("\n\n");
                output *= 0;
                output += 1;
                vfqBreak = false;

            } else if (ex_or_back.equalsIgnoreCase("2")) {
                System.out.println("~ ~ ~ It's a pleasure to serve
you ~ ~ ~");
                output *= 0;
                output += 2;
                vfqBreak = false;

            } else {
                System.out.println("Unexpected answer retry
.............!");
            }
        }
        return output;
    }



    //All Empty Queues
    public static void AllEmptyQueues(FuelQueue[] Queue){

        for (int i=0;i<5;i++){
            System.out.println("Fuel Pumps "+(i+1)+"\n");
            if (Queue[i].Passengers[0].getFirstName() == null){
                System.out.println("     This queue is empty
.....!\n\n");
            }else {
                for (int j=0;j<6;j++){

System.out.print(Queue[0].Passengers[j].getFirstName()+"     ");
                }
                System.out.println("\n\n");
            }
        }
    }
```

```
    //100 or VFQ: View all Fuel Queues
    public static void ViewAllFuelQueues(FuelQueue[] Queue){
        System.out.println("\n\nnull = Empty");
        System.out.println("Here is the order in which the customer
details is located  ......!");
        System.out.println("First name    Second name    VehicleNo
LitersRequired\n\n");
        for (int i=0;i<5;i++){
            System.out.println("> > > Fuel Pumps "+(i+1));
            for (int j=0;j<6;j++){
                System.out.print("\n     "+(j+1)+")
"+Queue[i].Passengers[j].getFirstName());
                System.out.print("
"+Queue[i].Passengers[j].getSecondName());
                System.out.print("
"+Queue[i].Passengers[j].getVehicleNo());
                System.out.print("
"+Queue[i].Passengers[j].getLitersRequired());
            }
            System.out.println("\n\n\n");
        }
    }



    //101 or VEQ: View all Empty Queues
    public static void ViewAllEmptyQueues(FuelQueue[] Queue) {
        System.out.println("\n\nnull = empty");
        System.out.println("This order has customer details
......!");
        System.out.println("First name    Second name    VehicleNo
LitersRequired");
        for (int j=0;j<5;j++){
            System.out.println("\n\n\nFuel Pumps "+(j+1));
            if (Queue[j].Passengers[5].getFirstName() == null) {
                for (int i = 0; i < 6; i++) {
                    System.out.print("\n     " + (i + 1) + ") " +
Queue[j].Passengers[i].getFirstName());
                    System.out.print("     " +
Queue[j].Passengers[i].getSecondName());
                    System.out.print("     " +
Queue[j].Passengers[i].getVehicleNo());
                    System.out.print("     " +
Queue[j].Passengers[i].getLitersRequired());
                }
            }else {
                System.out.print(" > > > This Fule Queue is Empty
```

```
.....!");
            }
        }
    }



    //102 or ACQ: Add customer to a Queue
    public static void AddCustomerToAQueue(FuelQueue[] Queue) {

        int[] hig = new int[5];
        int temp = 0;
        hig[0] = 0;
        hig[1] = 0;
        hig[2] = 0;
        hig[3] = 0;
        hig[4] = 0;
        int q1 = 0;
        int q2 = 0;
        int q3 = 0;
        int q4 = 0;
        int q5 = 0;

        for (int j = 0; j < 6; j++) {
            if (Queue[0].Passengers[j].getFirstName() == null) {
                hig[0] += 1;
                q1 += 1;
            }
            if (Queue[1].Passengers[j].getFirstName() == null) {
                hig[1] += 1;
                q2 += 1;
            }
            if (Queue[2].Passengers[j].getFirstName() == null) {
                hig[2] += 1;
                q3 += 1;
            }
            if (Queue[3].Passengers[j].getFirstName() == null) {
                hig[3] += 1;
                q4 += 1;
            }
            if (Queue[4].Passengers[j].getFirstName() == null) {
                hig[4] += 1;
                q5 += 1;
            }
        }

        if (hig[0] > hig[1]) {
            temp = hig[0];
            hig[0] = hig[1];
```

```
            hig[1] = temp;
        }
        if (hig[1] > hig[2]) {
            temp = hig[2];
            hig[2] = hig[1];
            hig[1] = temp;
        }
        if (hig[2] > hig[3]) {
            temp = hig[2];
            hig[2] = hig[3];
            hig[3] = temp;
        }
        if (hig[3] > hig[4]) {
            temp = hig[3];
            hig[3] = hig[4];
            hig[4] = temp;
        }

        Scanner user = new Scanner(System.in);
        boolean loopbreak1 = true;
        boolean loopbreak2 = true;
        boolean loopbreak4 = true;
        String FN = "";
        while (loopbreak1) {
            System.out.print("   1).   First Name :- ");
            FN = user.nextLine();
            if (FN.length() != 0) {
                loopbreak1 = false;
            } else {
                System.out.println("   Enter name correctly
.....!\n");
            }
        }
        String SN = "";
        while (loopbreak2) {
            System.out.print("   2).   Second Name :- ");
            SN = user.nextLine();
            if (SN.length() != 0) {
                loopbreak2 = false;
            } else {
                System.out.println("   Enter name correctly
.....!\n");
            }
        }
        System.out.print("   3).   Vehicle No :- ");
        String VN = user.nextLine();
        int LR = 0;
        do {
            System.out.print("   4).   No. of liters required :-
```

```
");
            if (user.hasNextInt()){
                LR = user.nextInt();
                if (LR <= 0){
                    System.out.println("This Value is
Incorrect\n");
                } else if (10 < LR) {
                    System.out.println("The maximum amount of oil
provided is only 10 liters\n");
                }else {
                    loopbreak4 = false;
                }
            }else {
                System.out.println("\n   Enter integer value
.....!\n");
                user.next();
            }
        }while (loopbreak4);

        if (Queue[4].Passengers[5].getFirstName() == null){
            if (q1 == hig[4]){
                int count1 = 6 - q1;
                Queue[0].Passengers[count1].setFirstName(FN);
                Queue[0].Passengers[count1].setSecondName(SN);
                Queue[0].Passengers[count1].setVehicleNo(VN);
                Queue[0].Passengers[count1].setLitersRequired(LR);
                System.out.println("Successfully added customer
into Fuel Pumps 1 Passenger "+(count1+1));

            } else if (q2 == hig[4]) {
                int count2 = 6 - q2;
                Queue[1].Passengers[count2].setFirstName(FN);
                Queue[1].Passengers[count2].setSecondName(SN);
                Queue[1].Passengers[count2].setVehicleNo(VN);
                Queue[1].Passengers[count2].setLitersRequired(LR);
                System.out.println("Successfully added customer
into Fuel Pumps 2 Passenger "+(count2+1));

            } else if (q3 == hig[4]) {
                int count3 = 6 - q3;
                Queue[2].Passengers[count3].setFirstName(FN);
                Queue[2].Passengers[count3].setSecondName(SN);
                Queue[2].Passengers[count3].setVehicleNo(VN);
                Queue[2].Passengers[count3].setLitersRequired(LR);
                System.out.println("Successfully added customer
into Fuel Pumps 3 Passenger "+(count3+1));

            } else if (q4 == hig[4]) {
                int count4 = 6 - q4;
```

```
                Queue[3].Passengers[count4].setFirstName(FN);
                Queue[3].Passengers[count4].setSecondName(SN);
                Queue[3].Passengers[count4].setVehicleNo(VN);
                Queue[3].Passengers[count4].setLitersRequired(LR);
                System.out.println("Successfully added customer
into Fuel Pumps 4 Passenger "+(count4+1));

            } else if (q5 == hig[4]) {
                int count5 = 6 - q5;
                Queue[4].Passengers[count5].setFirstName(FN);
                Queue[4].Passengers[count5].setSecondName(SN);
                Queue[4].Passengers[count5].setVehicleNo(VN);
                Queue[4].Passengers[count5].setLitersRequired(LR);
                System.out.println("Successfully added customer
into Fuel Pumps 5 Passenger "+(count5+1));
            }
        }else {
            WQFirstName.add(FN);
            WQSecondName.add(SN);
            WQVehicleNo.add(VN);
            WQLitersRequired.add(LR);
            System.out.println("Successfully added customer into
waiting Queue Position "+(WQFirstName.size()));
        }
    }



    //103 or RCQ: Remove a customer from a Queue (From a specific
location)
    public static void
RemoveCustomerFromSpecificLocation(FuelQueue[] Queue){
        Scanner user = new Scanner(System.in);
        boolean loopbreak = true;
        while (loopbreak){
            System.out.print("Do you want to remove customer from
which queues :- ");
            String input = user.nextLine();
            if (NumbOrNot(input)){
                int uinp = Integer.valueOf(input);
                if (0 < uinp && uinp < 6) {
                    System.out.print("And customer queues number :-
");
                    String input2 = user.nextLine();
                    if (NumbOrNot(input2)){
                        int uipn2 = Integer.valueOf(input2);
                        if (0<uipn2 && uipn2 <7){

                            int count1 = 0,count2 = 0,count3 =
```

```
0,count4 = 0,count5 = 0;

                            if
(Queue[0].Passengers[5].getFirstName() != null){
                                count1 += 1;
                            }
                            if
(Queue[1].Passengers[5].getFirstName() != null) {
                                count2 += 1;
                            }
                            if
(Queue[2].Passengers[5].getFirstName() != null) {
                                count3 += 1;
                            }
                            if
(Queue[3].Passengers[5].getFirstName() != null) {
                                count4 += 1;
                            }
                            if
(Queue[4].Passengers[5].getFirstName() != null) {
                                count5 += 1;
                            }

                            Queue[(uinp-1)].Passengers[(uipn2-
1)].setFirstName(null);
                            Queue[(uinp-1)].Passengers[(uipn2-
1)].setSecondName(null);
                            Queue[(uinp-1)].Passengers[(uipn2-
1)].setVehicleNo(null);
                            Queue[(uinp-1)].Passengers[(uipn2-
1)].setLitersRequired(0);
                            System.out.println("\nSuccessfully
remove customer form fuel queue "+uinp+" Passenger
"+uipn2+"\n\n\n");

                            if (uipn2 == 6){
                            }else {
                                for (int i=(uipn2-1);i<5;i++){
                                    Queue[(uinp-
1)].Passengers[i].setFirstName(Queue[(uinp-
1)].Passengers[(i+1)].getFirstName());
                                    Queue[(uinp-
1)].Passengers[i].setSecondName(Queue[(uinp-
1)].Passengers[(i+1)].getSecondName());
                                    Queue[(uinp-
1)].Passengers[i].setVehicleNo(Queue[(uinp-
1)].Passengers[(i+1)].getVehicleNo());
                                    Queue[(uinp-
1)].Passengers[i].setLitersRequired(Queue[(uinp-
```

```
1)].Passengers[(i+1)].getLitersRequired());
                                }
                                Queue[(uinp-
1)].Passengers[5].setFirstName(null);//WQFirstName.get(0)
                                Queue[(uinp-
1)].Passengers[5].setSecondName(null);//WQSecondName.get(0)
                                Queue[(uinp-
1)].Passengers[5].setVehicleNo(null);//WQVehicleNo.get(0)
                                Queue[(uinp-
1)].Passengers[5].setLitersRequired(0);//WQLitersRequired.get(0)

                                if (count1 == 1 &&
WQFirstName.size() != 0){
                                    if (count2 == 1 &&
WQFirstName.size() != 0){
                                        if (count3 == 1 &&
WQFirstName.size() != 0){
                                            if (count4 == 1 &&
WQFirstName.size() != 0){
                                                if (count5 == 1 &&
WQFirstName.size() != 0){
                                                    Queue[(uinp-
1)].Passengers[5].setFirstName(WQFirstName.get(0));
                                                    Queue[(uinp-
1)].Passengers[5].setSecondName(WQSecondName.get(0));
                                                    Queue[(uinp-
1)].Passengers[5].setVehicleNo(WQVehicleNo.get(0));
                                                    Queue[(uinp-
1)].Passengers[5].setLitersRequired(WQLitersRequired.get(0));

WQFirstName.remove(0);

WQSecondName.remove(0);

WQVehicleNo.remove(0);

WQLitersRequired.remove(0);
                                                    count1 -= 1;
                                                    count2 -= 1;
                                                    count3 -= 1;
                                                    count4 -= 1;
                                                    count5 -= 1;
                                                }
                                            }
                                        }
                                    }
                                }
                            }
                            loopbreak = false;
```

```
                        }else {
                            System.out.println("Unexpected answer
retry .....!\n");
                        }
                    }else {
                        System.out.println("Unexpected answer retry
.....!\n");
                    }
                }else {
                    System.out.println("Unexpected answer retry
.....!\n");
                }
            }else {
                System.out.println("Unexpected answer retry
.....!\n");
            }
        }
    }



    //104 or PCQ: Remove a served customer
    public static void RemoveServedCustomer(FuelQueue[] Queue) {

        int count1 = 0,count2 = 0,count3 = 0,count4 = 0,count5 = 0;

        if (allFuelStock_liters == 0){
            System.out.println(" > > > > > Fuel Stock Is Finish
.....!\n\n\n");
        }else {
            Scanner user = new Scanner(System.in);
            System.out.println("Select what Fuel Queue Customer you
want to Remove\n");
            boolean loopbreak = true;
            while (loopbreak) {
                System.out.print("Remove Fule Queue 1 (yes/no) :-
");
                String input = user.nextLine();
                if (input.equalsIgnoreCase("yes")){

                    count1 += 1;
                    q1ServedCustomer += 1;
                    q1ServedLiter +=
Queue[0].Passengers[0].getLitersRequired();
                    allFuelStock_liters -=
Queue[0].Passengers[0].getLitersRequired();
                    for (int i=0;i<5;i++){

Queue[0].Passengers[i].setFirstName(Queue[0].Passengers[(i+1)].getF
```

```
irstName());

Queue[0].Passengers[i].setSecondName(Queue[0].Passengers[(i+1)].get
SecondName());

Queue[0].Passengers[i].setVehicleNo(Queue[0].Passengers[(i+1)].getV
ehicleNo());

Queue[0].Passengers[i].setLitersRequired(Queue[0].Passengers[(i+1)]
.getLitersRequired());
                    }
                    Queue[0].Passengers[5].setFirstName(null);
                    Queue[0].Passengers[5].setSecondName(null);
                    Queue[0].Passengers[5].setVehicleNo(null);
                    Queue[0].Passengers[5].setLitersRequired(0);
                    System.out.println("\n");
                    loopbreak = false;
                } else if (input.equalsIgnoreCase("no")) {
                    System.out.println("\n");
                    loopbreak = false;
                }else {
                    System.out.println("Unexpected answer retry
.............!\n");
                }
            }
            loopbreak = true;
            while (loopbreak){
                System.out.print("Remove Fule Queue 2 (yes/no) :-
");
                String input2 = user.nextLine();
                if (input2.equalsIgnoreCase("yes")){

                    count2 += 1;
                    q2ServedCustomer += 1;
                    q2ServedLiter +=
Queue[1].Passengers[0].getLitersRequired();
                    allFuelStock_liters -=
Queue[1].Passengers[0].getLitersRequired();
                    for (int i=0;i<5;i++){

Queue[1].Passengers[i].setFirstName(Queue[1].Passengers[(i+1)].getF
irstName());

Queue[1].Passengers[i].setSecondName(Queue[1].Passengers[(i+1)].get
SecondName());

Queue[1].Passengers[i].setVehicleNo(Queue[1].Passengers[(i+1)].getV
ehicleNo());
```

```
Queue[1].Passengers[i].setLitersRequired(Queue[1].Passengers[(i+1)]
.getLitersRequired());
                    }
                    Queue[1].Passengers[5].setFirstName(null);
                    Queue[1].Passengers[5].setSecondName(null);
                    Queue[1].Passengers[5].setVehicleNo(null);
                    Queue[1].Passengers[5].setLitersRequired(0);
                    System.out.println("\n");
                    loopbreak = false;
                } else if (input2.equalsIgnoreCase("no")) {
                    System.out.println("\n");
                    loopbreak = false;
                }else {
                    System.out.println("Unexpected answer retry
.............!\n");
                }
            }
            loopbreak = true;
            while (loopbreak){
                System.out.print("Remove Fule Queue 3 (yes/no) :-
");
                String input3 = user.nextLine();
                if (input3.equalsIgnoreCase("yes")){

                    count3 += 1;
                    q3ServedCustomer += 1;
                    q3ServedLiter +=
Queue[2].Passengers[0].getLitersRequired();
                    allFuelStock_liters -=
Queue[2].Passengers[0].getLitersRequired();
                    for (int i=0;i<5;i++){

Queue[2].Passengers[i].setFirstName(Queue[2].Passengers[(i+1)].getF
irstName());

Queue[2].Passengers[i].setSecondName(Queue[2].Passengers[(i+1)].get
SecondName());

Queue[2].Passengers[i].setVehicleNo(Queue[2].Passengers[(i+1)].getV
ehicleNo());

Queue[2].Passengers[i].setLitersRequired(Queue[2].Passengers[(i+1)]
.getLitersRequired());
                    }
                    Queue[2].Passengers[5].setFirstName(null);
                    Queue[2].Passengers[5].setSecondName(null);
                    Queue[2].Passengers[5].setVehicleNo(null);
                    Queue[2].Passengers[5].setLitersRequired(0);
                    System.out.println("\n");
```

```
                    loopbreak = false;
                } else if (input3.equalsIgnoreCase("no")) {
                    System.out.println("\n");
                    loopbreak = false;
                }else {
                    System.out.println("Unexpected answer retry
.............!\n");
                }
            }
            loopbreak = true;
            while (loopbreak){
                System.out.print("Remove Fule Queue 4 (yes/no) :-
");
                String input4 = user.nextLine();
                if (input4.equalsIgnoreCase("yes")){

                    count4 += 1;
                    q4ServedCustomer += 1;
                    q4ServedLiter +=
Queue[3].Passengers[0].getLitersRequired();
                    allFuelStock_liters -=
Queue[3].Passengers[0].getLitersRequired();
                    for (int i=0;i<5;i++){

Queue[3].Passengers[i].setFirstName(Queue[3].Passengers[(i+1)].getF
irstName());

Queue[3].Passengers[i].setSecondName(Queue[3].Passengers[(i+1)].get
SecondName());

Queue[3].Passengers[i].setVehicleNo(Queue[3].Passengers[(i+1)].getV
ehicleNo());

Queue[3].Passengers[i].setLitersRequired(Queue[3].Passengers[(i+1)]
.getLitersRequired());
                    }
                    Queue[3].Passengers[5].setFirstName(null);
                    Queue[3].Passengers[5].setSecondName(null);
                    Queue[3].Passengers[5].setVehicleNo(null);
                    Queue[3].Passengers[5].setLitersRequired(0);
                    System.out.println("\n");
                    loopbreak = false;
                } else if (input4.equalsIgnoreCase("no")) {
                    System.out.println("\n");
                    loopbreak = false;
                }else {
                    System.out.println("Unexpected answer retry
.............!\n");
                }
```

```
            }
            loopbreak = true;
            while (loopbreak){
                System.out.print("Remove Fule Queue 5 (yes/no) :-
");
                String input5 = user.nextLine();
                if (input5.equalsIgnoreCase("yes")){

                    count5 += 1;
                    q5ServedCustomer += 1;
                    q5ServedLiter +=
Queue[4].Passengers[0].getLitersRequired();
                    allFuelStock_liters -=
Queue[4].Passengers[0].getLitersRequired();
                    for (int i=0;i<5;i++){

Queue[4].Passengers[i].setFirstName(Queue[4].Passengers[(i+1)].getF
irstName());

Queue[4].Passengers[i].setSecondName(Queue[4].Passengers[(i+1)].get
SecondName());

Queue[4].Passengers[i].setVehicleNo(Queue[4].Passengers[(i+1)].getV
ehicleNo());

Queue[4].Passengers[i].setLitersRequired(Queue[4].Passengers[(i+1)]
.getLitersRequired());
                    }
                    Queue[4].Passengers[5].setFirstName(null);
                    Queue[4].Passengers[5].setSecondName(null);
                    Queue[4].Passengers[5].setVehicleNo(null);
                    Queue[4].Passengers[5].setLitersRequired(0);
                    System.out.println("\n");
                    loopbreak = false;
                } else if (input5.equalsIgnoreCase("no")) {
                    System.out.println("\n");
                    loopbreak = false;
                }else {
                    System.out.println("Unexpected answer retry
.............!\n");
                }
            }
            //For Waiting Queue
            if (count1 == 1 && WQFirstName.size() != 0){

Queue[0].Passengers[5].setFirstName(WQFirstName.get(0));

Queue[0].Passengers[5].setSecondName(WQSecondName.get(0));
```

```
Queue[0].Passengers[5].setVehicleNo(WQVehicleNo.get(0));

Queue[0].Passengers[5].setLitersRequired(WQLitersRequired.get(0));
                WQFirstName.remove(0);
                WQSecondName.remove(0);
                WQVehicleNo.remove(0);
                WQLitersRequired.remove(0);
                count1 -= 1;
            }
            if (count2 == 1 && WQFirstName.size() != 0) {

Queue[1].Passengers[5].setFirstName(WQFirstName.get(0));

Queue[1].Passengers[5].setSecondName(WQSecondName.get(0));

Queue[1].Passengers[5].setVehicleNo(WQVehicleNo.get(0));

Queue[1].Passengers[5].setLitersRequired(WQLitersRequired.get(0));
                WQFirstName.remove(0);
                WQSecondName.remove(0);
                WQVehicleNo.remove(0);
                WQLitersRequired.remove(0);
                count2 -= 1;
            }
            if (count3 == 1 && WQFirstName.size() != 0) {

Queue[2].Passengers[5].setFirstName(WQFirstName.get(0));

Queue[2].Passengers[5].setSecondName(WQSecondName.get(0));

Queue[2].Passengers[5].setVehicleNo(WQVehicleNo.get(0));

Queue[2].Passengers[5].setLitersRequired(WQLitersRequired.get(0));
                WQFirstName.remove(0);
                WQSecondName.remove(0);
                WQVehicleNo.remove(0);
                WQLitersRequired.remove(0);
                count3 -= 1;
            }
            if (count4 == 1 && WQFirstName.size() != 0) {

Queue[3].Passengers[5].setFirstName(WQFirstName.get(0));

Queue[3].Passengers[5].setSecondName(WQSecondName.get(0));

Queue[3].Passengers[5].setVehicleNo(WQVehicleNo.get(0));

Queue[3].Passengers[5].setLitersRequired(WQLitersRequired.get(0));
                WQFirstName.remove(0);
```

```
                WQSecondName.remove(0);
                WQVehicleNo.remove(0);
                WQLitersRequired.remove(0);
                count4 -= 1;
            }
            if (count5 == 1 && WQFirstName.size() != 0) {

Queue[4].Passengers[5].setFirstName(WQFirstName.get(0));

Queue[4].Passengers[5].setSecondName(WQSecondName.get(0));

Queue[4].Passengers[5].setVehicleNo(WQVehicleNo.get(0));

Queue[4].Passengers[5].setLitersRequired(WQLitersRequired.get(0));
                WQFirstName.remove(0);
                WQSecondName.remove(0);
                WQVehicleNo.remove(0);
                WQLitersRequired.remove(0);
                count5 -= 1;
            }
        }
    }



    //105 or VCS: View Customers Sorted in alphabetical order
    public static void AtoZBefore(FuelQueue[] Queue,int no){

        String[] q = new String[6];
        for (int i=0;i<6;i++){
            q[i] = Queue[no].Passengers[i].getFirstName();
        }
        AtoZAfter(q);
    }
    public static void AtoZAfter(String[] VCS){
        if(VCS[0] != null) {

            //Counting null count
            int null_count = 0;
            for (int i = 0; i < VCS.length; i++) {
                if (VCS[i] == null) {
                    null_count = null_count + 1;
                }
            }

            //For remove null elements
            String[] VCS2 = new String[VCS.length - null_count];
            for (int i = 0; i < VCS.length; i++) {
                if (VCS[i] != null) {
```

```
                VCS2[i] = VCS[i];
            }
        }

        char[][] strToChar2D = new char[VCS2.length][];
        char[] all_1st_Elements = new char[VCS2.length];
        char[] withOutSameElements = new char[VCS2.length];
        char[][] AtoZ = new char[VCS2.length][];
        int x = 0;
        int y = 0;
        int z = 0;
        char temp;

        //Converting all string array elements to char array
        for (int i = 0; i < VCS2.length; i++) {
            strToChar2D[i] = VCS2[i].toCharArray();
        }

        //Selecting the first element from all the elements in
the char array
        for (int i = 0; i < VCS2.length; i++) {
            all_1st_Elements[i] = strToChar2D[i][0];
        }

        //Converting all elements to A to Z order
        for (int i = 0; i < VCS2.length; i++) {
            for (int ii = i + 1; ii < VCS2.length; ii++) {
                if (all_1st_Elements[i] > all_1st_Elements[ii])
{
                    temp = all_1st_Elements[i];
                    all_1st_Elements[i] = all_1st_Elements[ii];
                    all_1st_Elements[ii] = temp;
                }
            }
        }

        //Counting similar elements
        for (int i = 0; i < all_1st_Elements.length - 1; i++) {
            if (all_1st_Elements[i] != all_1st_Elements[i + 1])
{
                withOutSameElements[x++] = all_1st_Elements[i];
            } else {
                y = y + 1;
            }
        }
        withOutSameElements[x++] =
all_1st_Elements[all_1st_Elements.length - 1];

        char[] withOutSameElementsAndSize = new
```

```
char[VCS2.length - y];

            //Append without all element and size in new array
            for (int i = 0; i < withOutSameElementsAndSize.length;
i++) {
                withOutSameElementsAndSize[i] =
withOutSameElements[i];
            }

            //Append A to Z all elements in new array
            for (int i = 0; i < withOutSameElementsAndSize.length;
i++) {
                for (int ii = 0; ii < VCS2.length; ii++) {
                    if (withOutSameElementsAndSize[i] ==
strToChar2D[ii][0]) {
                        AtoZ[z++] = strToChar2D[ii];
                    }
                }
            }

            //Printing all elements in A to Z oder
            System.out.println(" ");
            for (int i = 0; i < AtoZ.length; i++) {
                System.out.print(" "+(i+1)+") ");
                System.out.println(AtoZ[i]);
            }
        }else {
            System.out.println(">>> This queues Empty .........!");
        }
    }



    //106 or SPD: Store Program Data into file
    public static void StoreDataInToFile(FuelQueue[] Queue){

        File fuelQ = new File("Fuel_Queues_Deta.txt");

        String FQ1 = "\n                        < < < < < < <***> > > >
> > >\n";
        String FQ2 = "                        |     Fuel Station Data
|\n";
        String FQ3 = "                       < < < < < < <***> > > > >
> >\n";
        String total = "\n\nTotal amount of oil currently remaining
:- ";
        String FQCount = "\n\n\n\n     Fuel Queue ";
        String AOCD = "\nAmount of oil currently delivered :- ";
        String NPCRO = "\nNumber of people currently receiving oil
```

```
:- ";
        String PCWQ = "\nPeople currently waiting in queue";

        try{
            fuelQ.createNewFile();
            BufferedWriter write = new BufferedWriter(new
FileWriter("Fuel_Queues_Deta.txt"));

write.write(FQ1+FQ2+FQ3+total+(allFuelStock_liters)+FQCount+"1"+AOC
D+(q1ServedLiter)+NPCRO+(q1ServedCustomer)+PCWQ);
            for (int i = 0; i < Queue[0].Passengers.length; i++){
                if(Queue[0].Passengers[0].getFirstName()==null){
                    write.write(" :- This queue is empty .....!");
                    break;
                }else if
(Queue[0].Passengers[i].getFirstName()==null){
                    break;
                }else {
                    write.write("\n");
                    write.write((i+1)+")
"+Queue[0].Passengers[i].getFirstName()+"
"+Queue[0].Passengers[i].getSecondName()+"
"+Queue[0].Passengers[i].getVehicleNo()+"
"+Queue[0].Passengers[i].getLitersRequired());
                }
            }

write.write(FQCount+"2"+AOCD+(q2ServedLiter)+NPCRO+(q2ServedCustome
r)+PCWQ);
            for (int i = 0; i < Queue[1].Passengers.length; i++){
                if(Queue[1].Passengers[0].getFirstName()==null){
                    write.write(" :- This queue is empty .....!");
                    break;
                }else if
(Queue[1].Passengers[i].getFirstName()==null){
                    break;
                }else {
                    write.write("\n");
                    write.write((i+1)+")
"+Queue[1].Passengers[i].getFirstName()+"
"+Queue[1].Passengers[i].getSecondName()+"
"+Queue[1].Passengers[i].getVehicleNo()+"
"+Queue[1].Passengers[i].getLitersRequired());
                }
            }

write.write(FQCount+"3"+AOCD+(q3ServedLiter)+NPCRO+(q3ServedCustome
r)+PCWQ);
            for (int i = 0; i < Queue[2].Passengers.length; i++){
```

```
                if(Queue[2].Passengers[0].getFirstName()==null){
                    write.write(" :- This queue is empty .....!");
                    break;
                }else if
(Queue[2].Passengers[i].getFirstName()==null){
                    break;
                }else {
                    write.write("\n");
                    write.write((i+1)+")
"+Queue[2].Passengers[i].getFirstName()+"
"+Queue[2].Passengers[i].getSecondName()+"
"+Queue[2].Passengers[i].getVehicleNo()+"
"+Queue[2].Passengers[i].getLitersRequired());
                }
            }

write.write(FQCount+"4"+AOCD+(q4ServedLiter)+NPCRO+(q4ServedCustome
r)+PCWQ);
            for (int i = 0; i < Queue[3].Passengers.length; i++){
                if(Queue[3].Passengers[0].getFirstName()==null){
                    write.write(" :- This queue is empty .....!");
                    break;
                }else if
(Queue[3].Passengers[i].getFirstName()==null){
                    break;
                }else {
                    write.write("\n");
                    write.write((i+1)+")
"+Queue[3].Passengers[i].getFirstName()+"
"+Queue[3].Passengers[i].getSecondName()+"
"+Queue[3].Passengers[i].getVehicleNo()+"
"+Queue[3].Passengers[i].getLitersRequired());
                }
            }

write.write(FQCount+"5"+AOCD+(q5ServedLiter)+NPCRO+(q5ServedCustome
r)+PCWQ);
            for (int i = 0; i < Queue[4].Passengers.length; i++){
                if(Queue[4].Passengers[0].getFirstName()==null){
                    write.write(" :- This queue is empty .....!");
                    break;
                }else if
(Queue[4].Passengers[i].getFirstName()==null){
                    break;
                }else {
                    write.write("\n");
                    write.write((i+1)+")
"+Queue[4].Passengers[i].getFirstName()+"
"+Queue[4].Passengers[i].getSecondName()+"
```

```
"+Queue[4].Passengers[i].getVehicleNo()+"
"+Queue[4].Passengers[i].getLitersRequired());
                }
            }
            System.out.println("Successfully Store Program Data
Into File .....!");
            System.out.println("\n");
            for (int i = 0;i <= 75; i++){
                System.out.print("- ");
            }
            System.out.println("\n\n");
            write.close();
        }catch (Exception e){
            System.out.println(e);
        }
    }



    //107 or LPD: Load Program Data from file
    public static void LoadDateFromFile(){
        File fuelQ = new File("Fuel_Queues_Deta.txt");

        try {
            BufferedReader reader = new BufferedReader(new
FileReader(fuelQ));
            String str;
            if ((str = reader.readLine()) == null){
                System.out.println("This File is Empty\n\n");
            }else {
                while ((str = reader.readLine()) != null){
                    System.out.println(str);
                }
                System.out.println("\n");
            }

        }catch (FileNotFoundException e){
            System.out.println("The system cannot find the file
specified");
        }catch (Exception e){
            System.out.println(e);
        }
    }



    //110 or IFQ: Income of each Fuel Queue
    public static void IncomeFuelQueues(){
```

```
        int literPrice = 430;
        System.out.println("     Fuel Queue 1");
        System.out.println("Amount of oil currently delivered :-
"+q1ServedLiter+" liters");
        System.out.println("Number of people currently receiving
oil :- "+q1ServedCustomer);
        System.out.println("Income of this Fuel Queue :-
"+(q1ServedLiter*literPrice)+" LKR\n\n");

        System.out.println("     Fuel Queue 2");
        System.out.println("Amount of oil currently delivered :-
"+q2ServedLiter+" liters");
        System.out.println("Number of people currently receiving
oil :- "+q2ServedCustomer);
        System.out.println("Income of this Fuel Queue :-
"+(q2ServedLiter*literPrice)+" LKR\n\n");

        System.out.println("     Fuel Queue 3");
        System.out.println("Amount of oil currently delivered :-
"+q3ServedLiter+" liters");
        System.out.println("Number of people currently receiving
oil :- "+q3ServedCustomer);
        System.out.println("Income of this Fuel Queue :-
"+(q3ServedLiter*literPrice)+" LKR\n\n");

        System.out.println("     Fuel Queue 4");
        System.out.println("Amount of oil currently delivered :-
"+q4ServedLiter+" liters");
        System.out.println("Number of people currently receiving
oil :- "+q4ServedCustomer);
        System.out.println("Income of this Fuel Queue :-
"+(q4ServedLiter*literPrice)+" LKR\n\n");

        System.out.println("     Fuel Queue 5");
        System.out.println("Amount of oil currently delivered :-
"+q5ServedLiter+" liters");
        System.out.println("Number of people currently receiving
oil :- "+q5ServedCustomer);
        System.out.println("Income of this Fuel Queue :-
"+(q5ServedLiter*literPrice)+" LKR\n\n\n");
    }
}
```

FuelQueue

```
public class FuelQueue {

    Passenger Passenger1 = new Passenger();
    Passenger Passenger2 = new Passenger();
    Passenger Passenger3 = new Passenger();
    Passenger Passenger4 = new Passenger();
    Passenger Passenger5 = new Passenger();
    Passenger Passenger6 = new Passenger();

    Passenger[] Passengers =
{(Passenger1),(Passenger2),(Passenger3),(Passenger4),(Passenger5),(
Passenger6)};
}
```

passenger

```
public class Passenger {
    String FirstName;
    String SecondName;
    String VehicleNo;
    int LitersRequired;

    public String getFirstName() {
        return FirstName;
    }

    public void setFirstName(String firstName) {
        FirstName = firstName;
    }

    public String getSecondName() {
        return SecondName;
    }

    public void setSecondName(String secondName) {
        SecondName = secondName;
    }

    public String getVehicleNo() {
        return VehicleNo;
    }

    public void setVehicleNo(String vehicleNo) {
        VehicleNo = vehicleNo;
    }

    public int getLitersRequired() {
```

```
        return LitersRequired;
    }

    public void setLitersRequired(int litersRequired) {
        LitersRequired = litersRequired;
    }
}
```

<<END>>