**Name = JUnaid SAleem 02-131202-057**

**Question1 :**

**Solution:**

class Chase\_Center

{

int[] item = { 50, 100, 200, 400, 500 };

int choice;

string name;

int orderNo = 4235;

int payment;

public Chase\_Center(string name)

{

this.name = name;

shop();

}

public void shop()

{

Console.WriteLine("\n-------- Welcome "+name+" ----------");

Console.WriteLine("1) Tea in Rs: "+item[0]);

Console.WriteLine("2) Burger in Rs: " + item[1]);

Console.WriteLine("3) Sandwitch in Rs: " + item[2]);

Console.WriteLine("4) Biryani in Rs: " + item[3]);

Console.WriteLine("5) Pizza in Rs: " + item[4]);

Console.Write("Enter your choice = ");

int n = int.Parse(Console.ReadLine());

if (n > 5)

{

Console.WriteLine("wrong press");

}

else

{

payment = item[n - 1];

}

Counter1();

}

public void Counter1()//generate receipt

{

Console.WriteLine("\n------------------ Counter 1 ----------------");

firstCus();

Class.customer.Enqueue(String.Format(name + "," + orderNo + "," + payment));

Console.Write("Here, is your receipt ==> " + name+orderNo);

Counter2();

}

public void Counter2()

{

Console.WriteLine("\n------------------ Counter 2 ----------------");

Console.WriteLine("your Bill is = Rs:" + payment);

}

public void firstCus()

{

string f = null;

try {

f = Class.customer.Peek();

}

catch (Exception e) {

payment = payment - ((payment / 100) \* 10);// 10% off

}

}

}

class Class

{

public static Queue<string> customer = new Queue<string>();

public void deleteCus()

{

Class.customer.Dequeue();

}

public static void searchCus()

{

bool fo = false;

foreach (string a in customer)

{

string[] name = a.Split(',');

if(name[0].StartsWith("S") == true)

{

fo = true;

Console.WriteLine("Found");

}

}

if (fo == false) {

Console.WriteLine("Not found");

}

}

}

static void Main(string[] args)

{

while (true)

{

Console.Write("Are you Admin(y/n) { q for exit} = ");

char i = Convert.ToChar(Console.ReadLine());

if (i == 'y')

{

if (Class.customer.Count == 0)

{

Console.WriteLine("No Customer");

}

else

{

foreach (string u in Class.customer)

{

Console.WriteLine(u);

}

}

}

if (i == 'q') break;

if(i == 'n')

{

Chase\_Center customer1 = new Chase\_Center("Saleem");

Chase\_Center customer2 = new Chase\_Center("Akbar");

Console.WriteLine("\nSearching Customer start with S name");

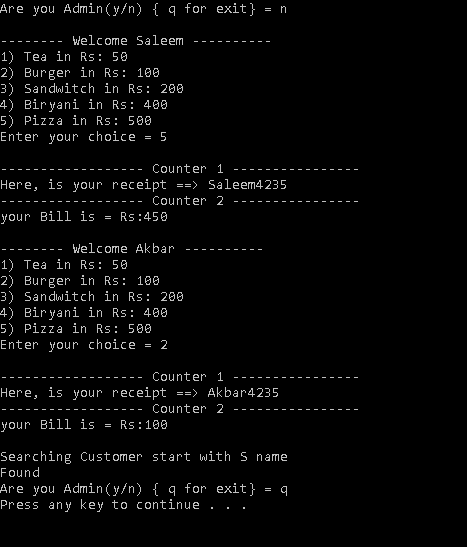
Class.searchCus();

}

}

}

**Output:-**



**Question2 :**

**Solution:**

1. **Binary Search Tree**

**2)**

class BST

{

public Node head;

public void add(int data)

{

if (head == null)

{

head = new Node(data);

}

else

{

adding(head, data);

}

}

public void DispalyTree(Node r)

{

if (r == null)

{

return;

}

Console.Write(r.data+" ");//root

DispalyTree(r.left);//left

DispalyTree(r.right);//right

}

private void adding(Node root,int d)

{

if (root==null || root.data == null)

{

return;

}

if (root.data > d)

{

adding(root.left, d);

if (root.left == null)

root.left = new Node(d);

return;

}

if (root.data < d)

{

adding(root.right, d);

if(root.right == null)

root.right = new Node(d);

return;

}

}

}

class Node

{

public Node left;

public Node right;

public int data;

public Node(int d)

{

data = d;

left = null;

right = null;

}

}

static void Main(string[] args)

{

BST myTree = new BST();

myTree.add(1);

myTree.add(2);

myTree.add(3);

myTree.DispalyTree(myTree.head);

}

**Output:-**



