Q1:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

class Item

{

long \_id;

public long Id

{

get { return \_id; }

set { \_id = value; }

}

string \_name, \_itemCode;

public string ItemCode

{

get { return \_itemCode; }

set { \_itemCode = value; }

}

public string Name

{

get { return \_name; }

set { \_name = value; }

}

double \_cost;

public double Cost

{

get { return \_cost; }

set { \_cost = value; }

}

public Item() { }

public Item(long \_id, string \_name, string \_itemCode, double \_cost)

{

Id = \_id;

Name = \_name;

ItemCode = \_itemCode;

Cost = \_cost;

}

public override string ToString()

{

return string.Format("{0} {1,10} {2,15} {3,15:0.0}", Id,Name,ItemCode,Cost);

}

public override bool Equals(object obj)

{

if (obj == null)

return false;

if (obj.GetType() != this.GetType())

return false;

Item i = (Item)obj;

return i.Name.Equals(this.Name,StringComparison.InvariantCultureIgnoreCase) &&

i.ItemCode.Equals(this.ItemCode,StringComparison.InvariantCultureIgnoreCase);

}

public override int GetHashCode()

{

return base.GetHashCode();

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

class Program

{

static void Main(string[] args)

{

Item[] items=new Item[2];

for(int i=0;i<2;i++)

{

Console.WriteLine("Enter item {0} detail:",i+1);

string d=Console.ReadLine();

string[] d1=d.Split(',');

items[i]=new Item(long.Parse(d1[0]),d1[1],d1[2],(double.Parse(d1[3])));

}

for(int i=0;i<2;i++)

{

Console.WriteLine("\nItem {0}:",i+1);

Console.WriteLine("{0} {1,10} {2,15} {3,15}", "ID","Name","Item Code","Cost");

Console.WriteLine(items[i].ToString());

}

if(items[0].Equals(items[1]))

{

Console.WriteLine("\nItem 1 is same as Item 2");

}

else

{

Console.WriteLine("\nItem 1 and Item 2 are different");

}

}

}

Q2:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Globalization;

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Enter the coupon code:");

string code = Console.ReadLine();

Console.WriteLine("1.Validate coupon code\n2.Check validity of coupon code\nEnter the choice:");

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

if (ch == 1)

{

bool ag=ValidateCouponCode(code);

if(ag==true)

{

Console.WriteLine("Coupon code validated");

}

else if (ag == false)

{

if (!code.StartsWith("Q"))

{

Console.WriteLine("Input string was not in the correct format");

Console.WriteLine("Coupon code is invalid");

}

else

{

Console.WriteLine("Coupon code is invalid");

}

}

}

else if (ch == 2)

{

Console.WriteLine("Enter the bought date:");

DateTime date = DateTime.ParseExact(Console.ReadLine(), "dd-MM-yyyy", null);

bool aga = CheckValidityOfCouponCode(code, date);

if (aga == true)

{

Console.WriteLine("Coupon code is valid");

}

else

{

Console.WriteLine("The validity of coupon code is over");

}

}

}

public static bool ValidateCouponCode(string couponCode)

{

int c = 0;

for (int i = 0; i < couponCode.Length; i++)

{

if (couponCode.Length == 10 &&

char.IsDigit(couponCode[2]) && char.IsDigit(couponCode[3]) &&

char.IsLetter(couponCode[0]) && char.IsLetter(couponCode[9]) &&

(char.IsLetter(couponCode[i]) || char.IsDigit(couponCode[i])))

{ c = 1; }

}

if (c == 1) return true;

else return false;

}

public static bool CheckValidityOfCouponCode(string couponCode, DateTime boughtDate)

{

string a = couponCode.Substring(2, 2);

int n = int.Parse(a);

DateTime d = DateTime.ParseExact("01-01-2018","dd-MM-yyyy",null,System.Globalization.DateTimeStyles.None);

var d1 = boughtDate.AddDays(n);

//var diff = (d - boughtDate).TotalDays;

if (d1>d)

{

return true;

}

else

{

return false;

}

}

}

Q3:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

class Item

{

long \_id;

public long Id

{

get { return \_id; }

set { \_id = value; }

}

string \_name, \_itemCode;

public string ItemCode

{

get { return \_itemCode; }

set { \_itemCode = value; }

}

public string Name

{

get { return \_name; }

set { \_name = value; }

}

double \_cost;

public double Cost

{

get { return \_cost; }

set { \_cost = value; }

}

public Item() { }

public Item(long \_id, string \_name, string \_itemCode, double \_cost)

{

Id = \_id;

Name = \_name;

ItemCode = \_itemCode;

Cost = \_cost;

}

public override string ToString()

{

return string.Format("{0} {1,15} {2,15:0.0}", Name,ItemCode,Cost);

}

public static Item CreateItem(string itemDetail)

{

string[] s = itemDetail.Split(',');

Item t=new Item();

t=new Item(long.Parse(s[0]),s[1],s[2],double.Parse(s[3]));

return t;

}

public static Item SearchItemByName(string itemName, Item itemList)

{

Item t=null;

if (itemName.Equals(itemList.Name))

{

t=itemList;

}

return t;

}

public static Item FindAllItemByPriceRange(Item itemList, double minRate, double maxRate)

{

Item t = null;

if (itemList.Cost <= maxRate && itemList.Cost >= minRate)

{

t = itemList;

}

return t;

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

class Program

{

static void Main(string[] args)

{

int ch;

List<Item> itemList = new List<Item>();

do

{

Console.WriteLine("1. Add items\n2. Search item by name\n3. Get item between price range\n4. Exit");

Console.WriteLine("Enter your choice:");

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

if (ch == 1)

{

Console.WriteLine("Enter the number of items:");

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

for (int i = 0; i < n; i++)

{

string s = Console.ReadLine();

itemList.Add(Item.CreateItem(s));

}

}

if (ch == 2)

{

Item t = null;

int count = 0;

Console.WriteLine("Enter the name:");

string s = Console.ReadLine();

foreach (var item1 in itemList)

{

t = Item.SearchItemByName(s, item1);

if (t != null)

{

Console.WriteLine("Item Detail");

Console.WriteLine("Item name: " + item1.Name);

Console.WriteLine("Item code: " + item1.ItemCode);

Console.WriteLine("Item Cost: {0:0.0}", item1.Cost); count++;

}

}

if (count == 0) Console.WriteLine("Item "+s+" not found");

}

if (ch == 3)

{

Console.WriteLine("Enter the max and min cost:");

double min = double.Parse(Console.ReadLine());

double max = double.Parse(Console.ReadLine());

Item t = null;

Console.WriteLine("{0} {1,15} {2,15}", "Name", "Code", "Cost");

foreach (var item1 in itemList)

{

t = Item.FindAllItemByPriceRange(item1, min, max);

if (t != null)

{

Console.WriteLine(t.ToString());

}

}

//Item.FindAllItemsByPriceRange(itemList, min, max);

}

if (ch == 4)

{

break;

}

} while (ch > 0 && ch <= 4);

}

}

Q4:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

class Item

{

long \_id;

public long Id

{

get { return \_id; }

set { \_id = value; }

}

string \_name;

public string Name

{

get { return \_name; }

set { \_name = value; }

}

double \_cost;

public double Cost

{

get { return \_cost; }

set { \_cost = value; }

}

int \_quantity;

public int Quantity

{

get { return \_quantity; }

set { \_quantity = value; }

}

public Item() { }

public Item(long \_id, string \_name, double \_cost, int \_quantity)

{

Id = \_id;

Name = \_name;

Cost = \_cost;

Quantity = \_quantity;

}

public static double CalculateTotalBill(List<Item> itemList)

{

double total = 0;

foreach (var i in itemList)

{

total = total + (i.\_cost \* i.\_quantity);

}

return total;

}

public static double CalculateTotalBill(List<Item> itemList, int deliveryType)

{

double total = 0;

double amt = 0;

foreach (var i in itemList)

{

total = total + (i.\_cost \* i.\_quantity);

}

if (deliveryType == 2)

{

amt = total + (0.08 \* total);

}

else if (deliveryType == 1)

{

amt = total + (0.15 \* total);

}

return amt;

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

class Program

{

static void Main(string[] args)

{

List<Item> itemList = new List<Item>();

Console.WriteLine("Enter the number of items:");

int noOfItems = Convert.ToInt32(Console.ReadLine());

for (int i = 0; i < noOfItems; i++)

{

String itemDetails = Console.ReadLine();

String[] splited = itemDetails.Split(',');

itemList.Add(new Item(Convert.ToInt64(splited[0]), splited[1], Convert.ToDouble(splited[2]), Convert.ToInt32(splited[3])));

}

Console.WriteLine("1.Store\n2.Online\nEnter the choice:");

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

if (ch == 1)

{

Console.WriteLine("Total amount:{0:0.00}", Item.CalculateTotalBill(itemList));

}

else if (ch == 2)

{

Console.WriteLine("1.One day delivery\n2.Normal delivery\nEnter delivery type:");

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

Console.WriteLine("Total amount:{0:0.00}", Item.CalculateTotalBill(itemList, choice));

}

}

}

Q6:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace quess6

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Enter the number of purchase:");

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

SortedDictionary<String, int> d = new SortedDictionary<string, int>();

for (int i = 0; i < n; i++)

{

string inp = Console.ReadLine();

Purchase.ObtainPurchaseWithItem(d, inp);

}

Console.WriteLine("{0} {1,15}", "Item name", "Quantity");

foreach (var item in d)

{

Console.WriteLine("{0} {1,15}", item.Key, item.Value);

}

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace quess6

{

class Purchase

{

private long \_id;

private DateTime \_purchaseDate;

private double \_totalAmount;

private string \_user;

public long Id

{

get { return \_id; }

set { \_id = value; }

}

public DateTime PurchaseDate

{

get { return \_purchaseDate; }

set { \_purchaseDate = value; }

}

public double TotalAmount

{

get { return \_totalAmount; }

set { \_totalAmount = value; }

}

public string User

{

get { return \_user; }

set { \_user = value; }

}

public Purchase()

{ }

public Purchase(long \_id, DateTime \_purchaseDate, double \_totalAmount, string \_user)

{

this.\_id = \_id;

this.\_purchaseDate = \_purchaseDate;

this.\_totalAmount = \_totalAmount;

this.\_user = \_user;

}

public static void ObtainPurchaseWithItem(SortedDictionary<String, int> dictionary, string purchaseDetail)

{

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

for (int j = 3; j < a.Length; j = j + 3)

{

if (dictionary.ContainsKey(a[j]))

{

dictionary[a[j]] = dictionary[a[j]] + Convert.ToInt32(a[j + 2]);

}

else

{

dictionary.Add(a[j], Convert.ToInt32(a[j + 2]));

}

}

}

}

}