PROBLEM

User List

Tata Steel uses a software system called EMS(Event Management System) for managing seminars and meetings within the departments. Employees can use this system to reserve meeting space and specify technical services for their events.
Write a program to read the users who have reserved the meeting space and display the same using a List.

Problem Specifications

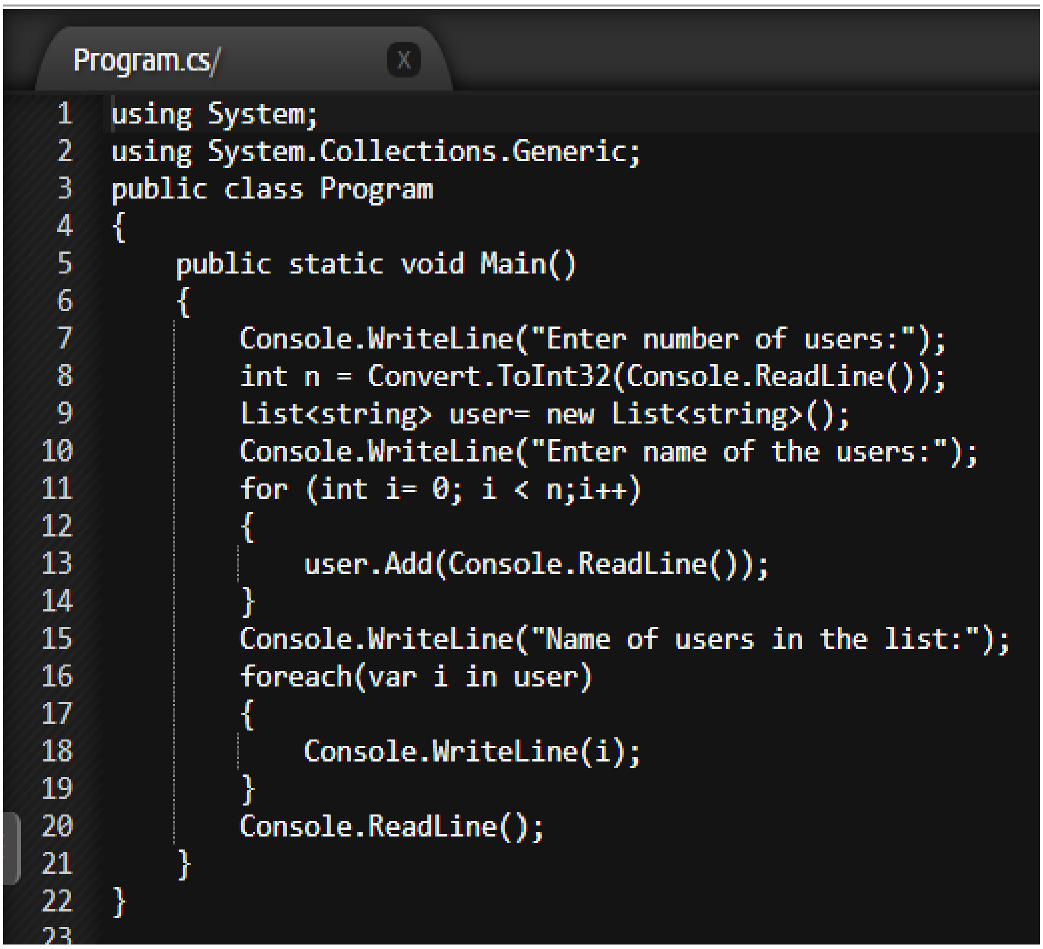
The Main Class file name should be **Program.cs** and the class name should be Program.
Use List with generic type string. Get the names of the user as input and store it in the list. Display the names of the user in the list.

Input and Output Format

Refer sample input and output for formatting specification
All text in bold corresponds to the input and rest corresponds to the output.

Sample Input and Output

Enter number of users:
5
Enter name of the users:
guru
priyanka
kavitha
gowtham
manoj
Name of users in the list:
guru
priyanka
kavitha
gowtham
manoj



```
using System;
using System.Collections.Generic;
public class Program
{
    public static void Main()
    {
        Console.WriteLine("Enter number of users:");
        int n = Convert.ToInt32(Console.ReadLine());
        List<string> user= new List<string>();
        Console.WriteLine("Enter name of the users:");
        for (int i= 0; i < n;i++)
        {
            user.Add(Console.ReadLine());
        }
        Console.WriteLine("Name of users in the list:");
        foreach(var i in user)
        {
            Console.WriteLine(i);
        }
        Console.ReadLine();
    }
}
```

PROBLEM

Dictionary

The placement coordinator prepares a statistics to know which company has given maximum number of offers to the students of his college every year. He stores the company name and number of offers in a dictionary. Here company name is the key and number of offers is the value.

Write a program to display the name of the company which has given the maximum number of offers this year. The program should contain the following attributes,

_companyname - string
_offers - int

Problem Specifications:

The Main Class file name should be **Program.cs** and the class name should be **Program**.

Use Dictionary with generic type <string, int>. Get the company names and number of offers as input and store it in the Dictionary. Display the name of the company which has given the maximum number of offers this year.

Input and Output Format:

Refer sample input and output for formatting specifications.

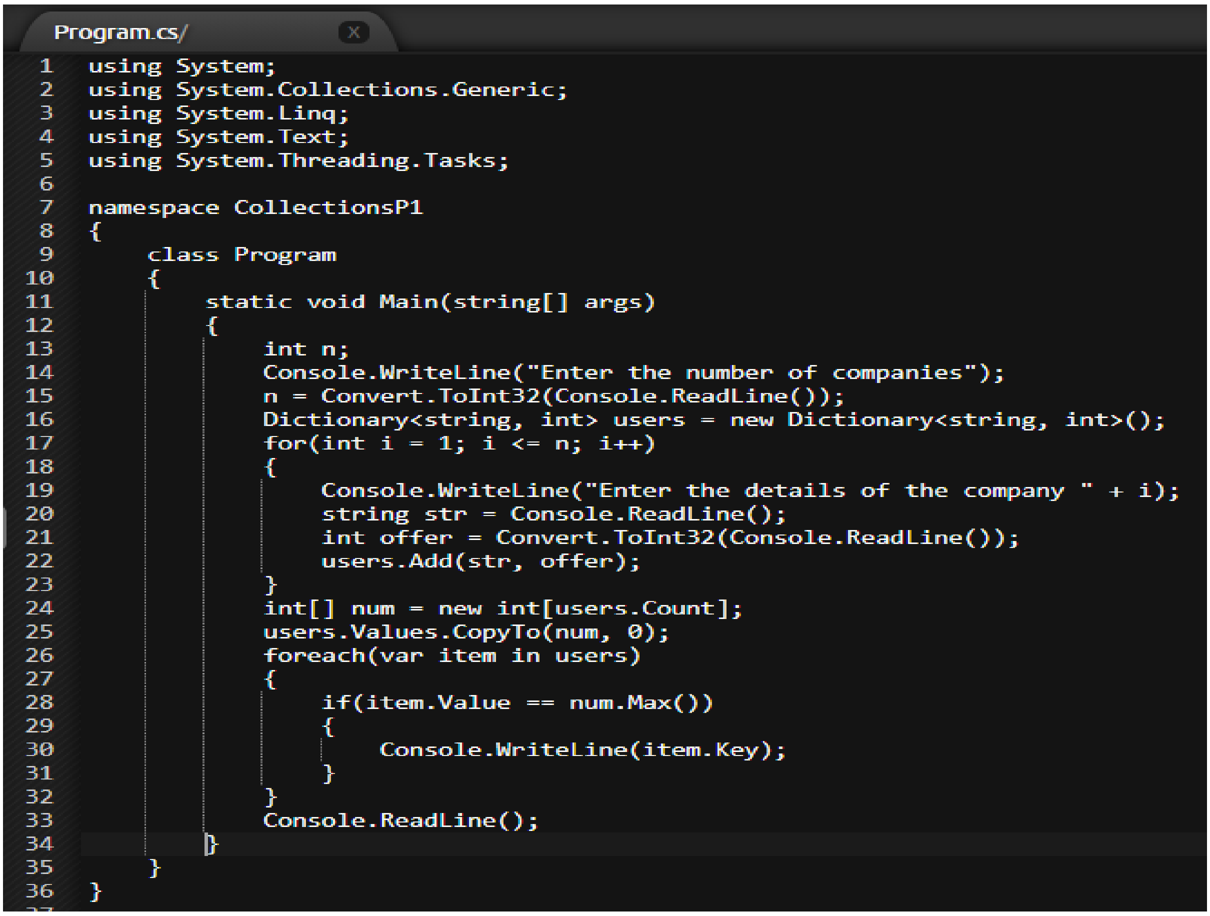
Input consists of number of companies who visited the campus this year, company name and number of recruitment offers given by the company.

Output consists of company name with maximum of recruitments.

All text in bold corresponds to input and the rest corresponds to output.

Sample Input and Output:

Enter the number of companies
3
Enter the details of the company 1
Amphisoft Technologies
57
Enter the details of the company 2
E-box Academy
49
Enter the details of the company 3
InfoView
46
Amphisoft Technologies



```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace CollectionsP1
{
    class Program
    {
        static void Main(string[] args)
        {
            int n;
            Console.WriteLine("Enter the number of companies");
            n = Convert.ToInt32(Console.ReadLine());
            Dictionary<string, int> users = new Dictionary<string, int>();
            for(int i = 1; i <= n; i++)
            {
                Console.WriteLine("Enter the details of the company " + i);
                string str = Console.ReadLine();
                int offer = Convert.ToInt32(Console.ReadLine());
                users.Add(str, offer);
            }
            int[] num = new int[users.Count];
            users.Values.CopyTo(num, 0);
            foreach(var item in users)
            {
                if(item.Value == num.Max())
                {
                    Console.WriteLine(item.Key);
                }
            }
            Console.ReadLine();
        }
    }
}
```

PROBLEM

Sort Hall Details - IComparable Interface

[Note:
Strictly adhere to the object-oriented specifications given as a part of the problem statement.
Follow the naming conventions as mentioned. Create separate classes in separate files.]

Create a class **Hall** with the following private member variables/ attribute which extends **IComparable** interface.

Data Type	Variable
string	_name
long	_mobileNo
string	_ownerName
double	_costPerDay

Include appropriate properties.
Include a four-argument constructor with parameters in the following order,
public Hall(string _name, long _mobileNo, string _ownerName, double _costPerDay)
Override **CompareTo** Method in Hall class which compares the halls based on **costPerDay** of the halls.
Override **ToString()** method in the **Hall** class to display the Hall details specified in the specified format.
string.Format("{0,-20} {1,-15} {2,-15} {3,-10}",_name,_mobileNo,_ownerName,_costPerDay)

Note: Consider the costPerDay of halls are unique.

Create **Program** class with **Main** method, get the number of halls and hall details.
Get Hall details in a comma separated format in the following order **(name,mobileNo,ownerName,costPerDay).**
Display the sorted hall details.

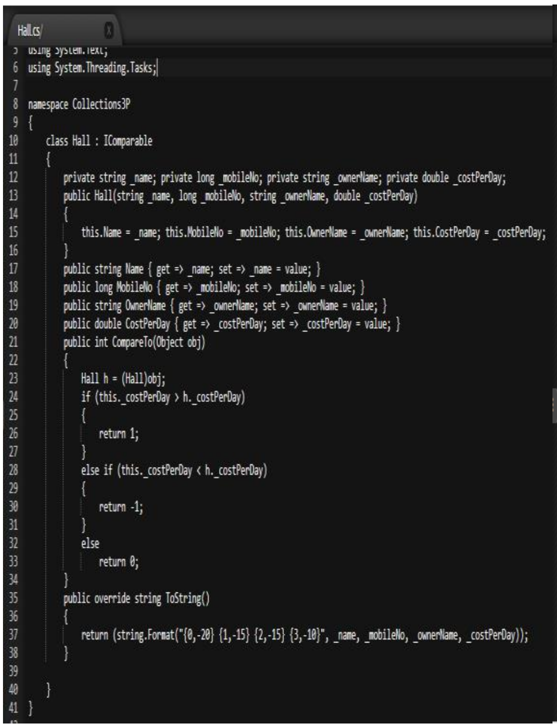
Input and Output Format

Refer sample input and output for formatting specifications.
All text in bold corresponds to the input and the rest corresponds to output.

Sample Input And Output 1:

Enter total number of halls
2
Enter Hall Details
royal orchid,9092304616,John,25000
Sodit Banquet,9381415278,peter,14000

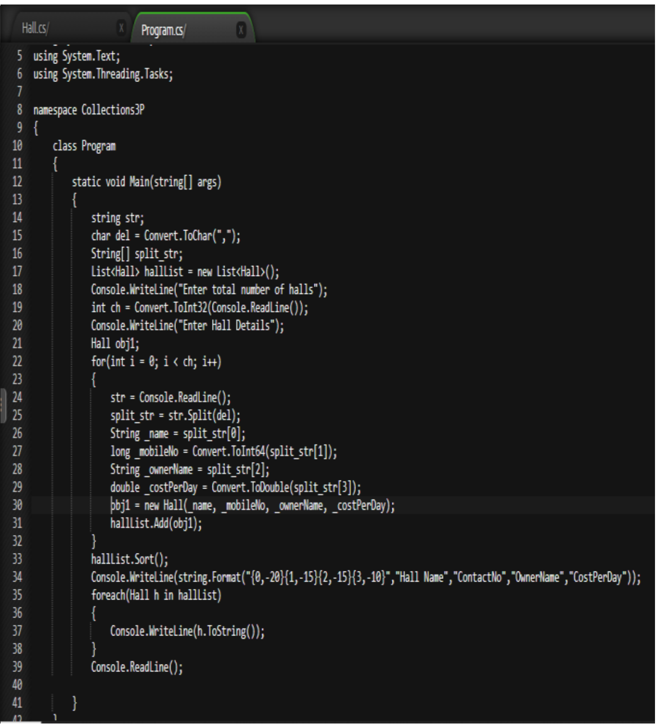
```
Hall Name        ContactNo     OwnerName     CostPerDay
Sodit Banquet      9381415278     peter        14000
royal orchid     9092304616     John         25000
```



```
using System.Text;
using System.Threading.Tasks;

namespace Collections3P
{
    class Hall : IComparable
    {
        private string _name; private long _mobileNo; private string _ownerName; private double _costPerDay;
        public Hall(string _name, long _mobileNo, string _ownerName, double _costPerDay)
        {
            this.Name = _name; this.MobileNo = _mobileNo; this.OwnerName = _ownerName; this.CostPerDay = _costPerDay;
        }
        public string Name { get => _name; set => _name = value; }
        public long MobileNo { get => _mobileNo; set => _mobileNo = value; }
        public string OwnerName { get => _ownerName; set => _ownerName = value; }
        public double CostPerDay { get => _costPerDay; set => _costPerDay = value; }
        public int CompareTo(Object obj)
        {
            Hall h = (Hall)obj;
            if (this._costPerDay > h._costPerDay)
            {
                return 1;
            }
            else if (this._costPerDay < h._costPerDay)
            {
                return -1;
            }
            else
                return 0;
        }
        public override string ToString()
        {
            return (string.Format("{0,-20} {1,-15} {2,-15} {3,-10}", _name, _mobileNo, _ownerName, _costPerDay));
        }

    }
}
```



```
using System.Text;
using System.Threading.Tasks;

namespace Collections3P
{
    class Program
    {
        static void Main(string[] args)
        {
            string str;
            char del = Convert.ToChar(",");
            String[] split_str;
            List<Hall> hallList = new List<Hall>();
            Console.WriteLine("Enter total number of halls");
            int ch = Convert.ToInt32(Console.ReadLine());
            Console.WriteLine("Enter Hall Details");
            Hall obj1;
            for(int i = 0; i < ch; i++)
            {
                str = Console.ReadLine();
                split_str = str.Split(del);
                String _name = split_str[0];
                long _mobileNo = Convert.ToInt64(split_str[1]);
                String _ownerName = split_str[2];
                double _costPerDay = Convert.ToDouble(split_str[3]);
                obj1 = new Hall(_name, _mobileNo, _ownerName, _costPerDay);
                hallList.Add(obj1);
            }
            hallList.Sort();
            Console.WriteLine(string.Format("{0,-20}{1,-15}{2,-15}{3,-10}","Hall Name","ContactNo","OwnerName","CostPerDay"));
            foreach(Hall h in hallList)
            {
                Console.WriteLine(h.ToString());
            }
            Console.ReadLine();

        }
```

PROBLEM

Raquel is the director of a Event management company. She decided to manage all the events details using a application. Franziska helped them in creating a application. Now they have started using the application. While uploading the hall details, they found that there are dual entries for some of the hall.

Write a C# program to get the hall names from the user and display the unique hall names use **IEnumerator** to iterate the set.

Create **Program** class with **Main** method, get the list of hall names and display the unique hall entries.

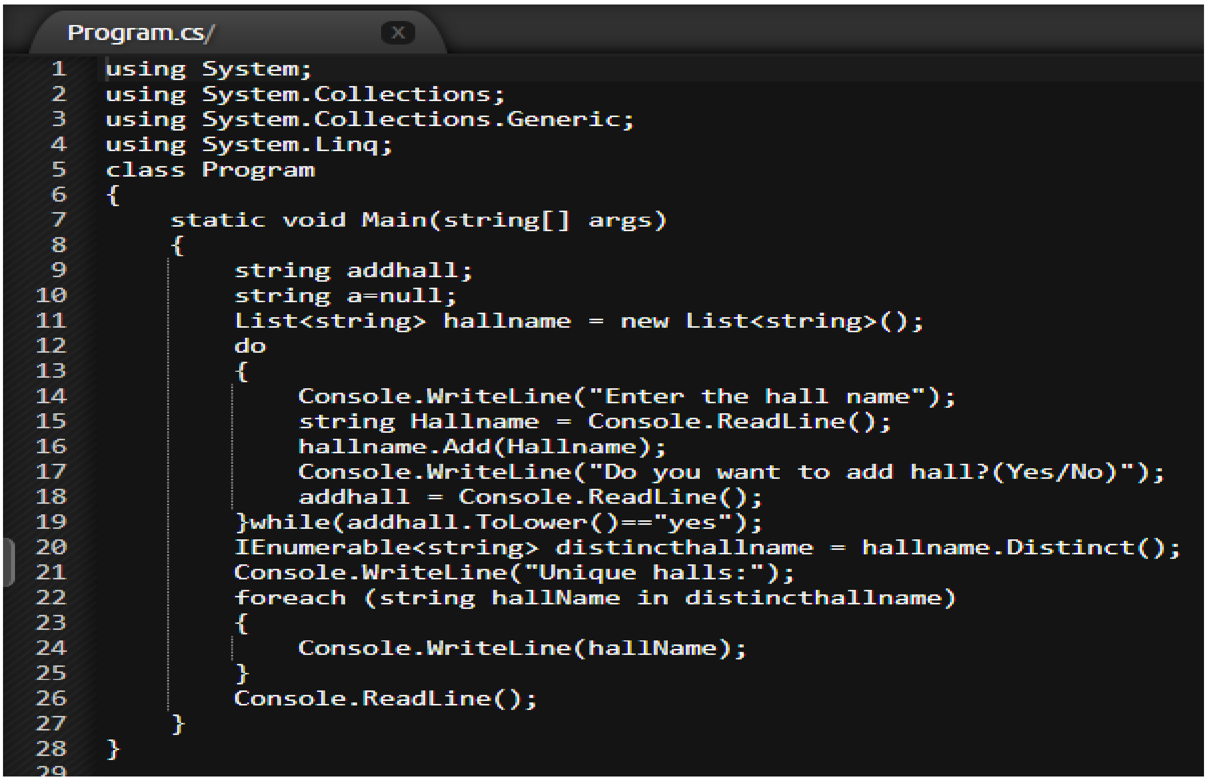
Input and Output Format

Refer sample input and output for formatting specifications.
All text in bold corresponds to the input and the rest corresponds to output.

Sample Input And Ouput :

Enter the hall name
Hall Paradise
Do you want to add hall?(Yes/No)
Yes
Enter the hall name
Rudolfinum
Do you want to add hall?(Yes/No)
Yes
Enter the hall name
The Walt Disney Concert Hall
Do you want to add hall?(Yes/No)
Yes
Enter the hall name
Rudolfinum
Do you want to add hall?(Yes/No)
No
Unique halls:
Hall Paradise

Rudolfinum
The Walt Disney Concert Hall



```
using System;
using System.Collections;
using System.Collections.Generic;
using System.Linq;
class Program
{
    static void Main(string[] args)
    {
        string addhall;
        string a=null;
        List<string> hallname = new List<string>();
        do
        {
            Console.WriteLine("Enter the hall name");
            string Hallname = Console.ReadLine();
            hallname.Add(Hallname);
            Console.WriteLine("Do you want to add hall?(Yes/No)");
            addhall = Console.ReadLine();
        }while(addhall.ToLower()=="yes");
        IEnumerable<string> distincthallname = hallname.Distinct();
        Console.WriteLine("Unique halls:");
        foreach (string hallName in distincthallname)
        {
            Console.WriteLine(hallName);
        }
        Console.ReadLine();
    }
}
```