LINQ –Day1 Assignments

Using Console Application:

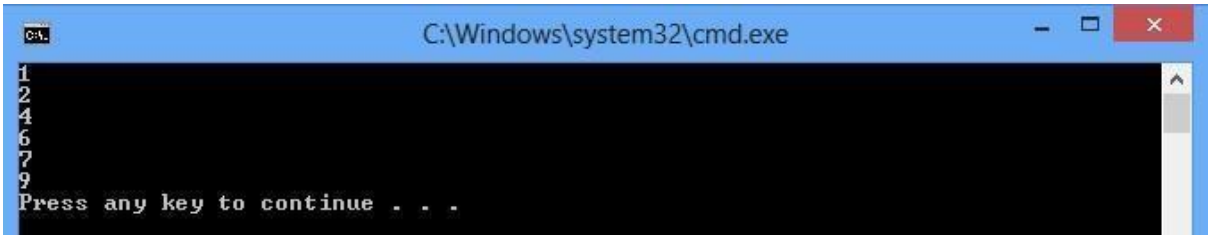
1- Declare a List of numbers as shown :

```
List<int> numbers = new List<int>() {
2,4,6,7,1,4,2,9,1};
```

Then write the following queries

Query1: Display numbers without any repeated Data and sorted

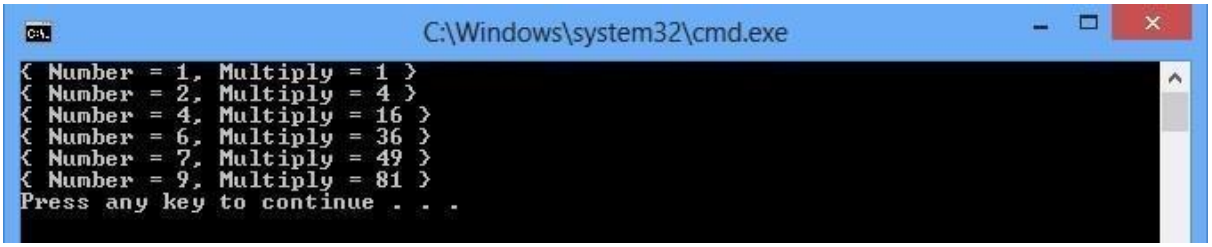
o/p should be as follow:




```
            List<int> numbers = new List<int>() { 2, 4,
6, 7, 1, 4, 2, 9, 1 };
            var q1 = numbers.Select(n =>
n).Distinct().OrderBy(n => n);
            foreach (var s in q1)
            {
                Console.WriteLine(s);
            }
```

Query2: using **Query1** result and show each number and it's multiplication

o/p should be as follow:




```
        List<int> numbers = new List<int>() { 2, 4, 6,
7, 1, 4, 2, 9, 1 };
        var q1 = numbers.Select(n =>
n).Distinct().OrderBy(n => n);
        var results = q1.Select(n => new { Number = n,
Multiply = n * n });
        foreach (var result in results)
        {
            Console.WriteLine($"< Number =
{result.Number}, Multiply = {result.Multiply} >");
        }
```

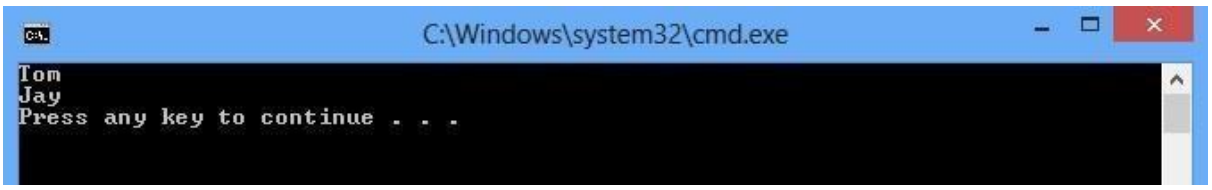
2- declare an array of names as shown :

```
string[] names = { "Tom", "Dick", "Harry", "MARY",
"Jay" };
```

Then write the following queries:

Query1: Select names with length equal 3.

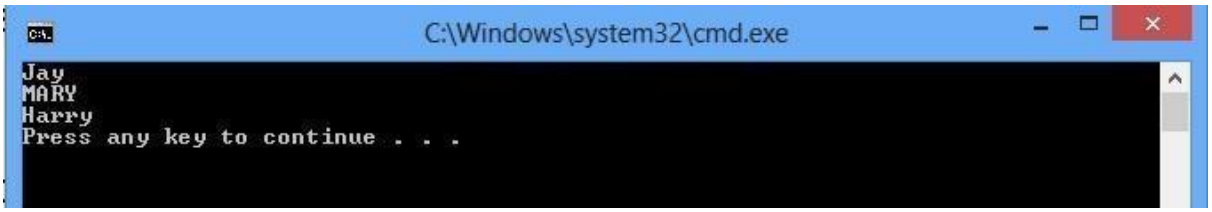
o/p should be as follow:




```
        string[] names = { "Tom", "Dick", "Harry",
"MARY", "Jay" };
        var stringQuery = names.Where(s => s.Length ==
3);
        foreach (var name in stringQuery)
        {
            Console.WriteLine(name);
        }
```

Query2: Select names that contains “a” letter (Capital or Small)then sort them by length

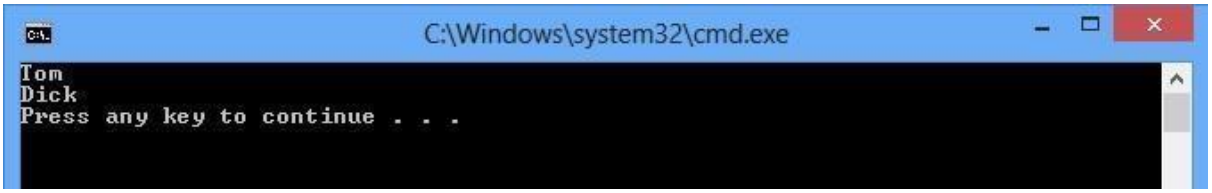
o/p should be as follow:




```
     string[] names = { "Tom", "Dick", "Harry", "MARY",
"Jay" };
     var stringQuery = names.Where(s =>
s.ToLower().Contains('a')).OrderBy(s => s.Length);
     foreach (var name in stringQuery)
     {
         Console.WriteLine(name);
     }
```

Query3: Display the first 2 names

o/p should be as follow:



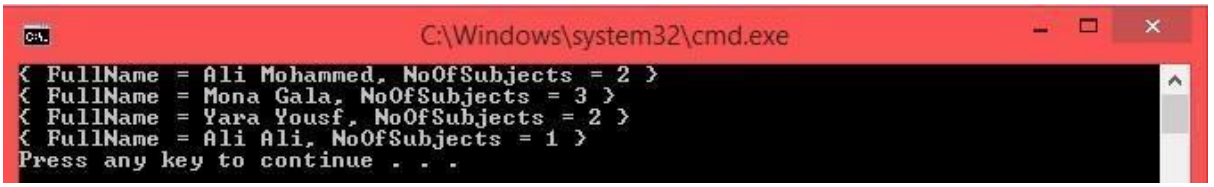

```
      string[] names = { "Tom", "Dick", "Harry", "MARY",
"Jay" };
    var stringQuery = names.Take(2);
    foreach (var name in stringQuery)
    {
        Console.WriteLine(name);
    }
```

3- Declare a class Subject that contains the following properties (Code, Name) and declare Student class that contains the following properties (ID, FirstName, LastName , Subject []), then define List of students As follow

```
List<Student> students=new List<Student> (){
new Student(){ ID=1, FirstName="Ali",
LastName="Mohammed",
subjects=new Subjcet[]{ new Subjcet(){ Code=22,Name="EF"},
new Subjcet(){ Code=33,Name="UML"}}},              new
Student(){ ID=2, FirstName="Mona", LastName="Gala",
subjects=new Subject []{ new Subject(){ Code=22,Name="EF"}, new
Subject (){ Code=34,Name="XML"},new Subject (){ Code=25,
Name="JS"}}},              new
Student(){ ID=3, FirstName="Yara", LastName="Yousf", subjects=new
Subject
[]{ new Subject (){ Code=22,Name="EF"}, new Subject
(){ Code=25,Name="JS"}}},                new
Student(){ ID=1, FirstName="Ali", LastName="Ali",
subjects=new Subject []{  new Subject (){ Code=33,Name="UML"}}}, };
```

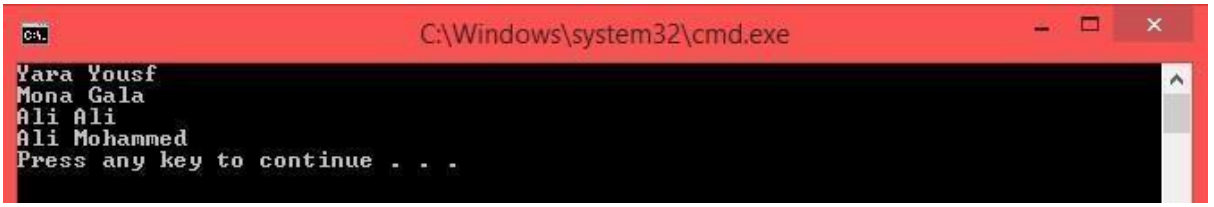
Then write the following queries

Query1: Display Full name and number of subjects for each student as follow




```
            var query1 = students.Select(s => new
            {
                FullName = $"{s.FirstName} {s.LastName}",
                NumberOfSubjects = s.subjects.Length
            });
            foreach (var result in query1)
            {
                Console.WriteLine($"< FullName =
{result.FullName}, NoOfSubjects =
{result.NumberOfSubjects} >");
            }
```

Query2: Write a query which orders the elements in the list by FirstName **Descending** then by LastName **Ascending** and result of query displays only first names and last names for the elements in list as follow




```
            var query2 = students.OrderByDescending(s
=> s.FirstName).ThenBy(s => s.LastName);
            foreach(var result in query2)
            {
                Console.WriteLine($"{result.FirstName}
{result.LastName}");
            }
```