

LINQ –Day2 Assignments

Using Console Application:

1- Declare a List of numbers as shown :

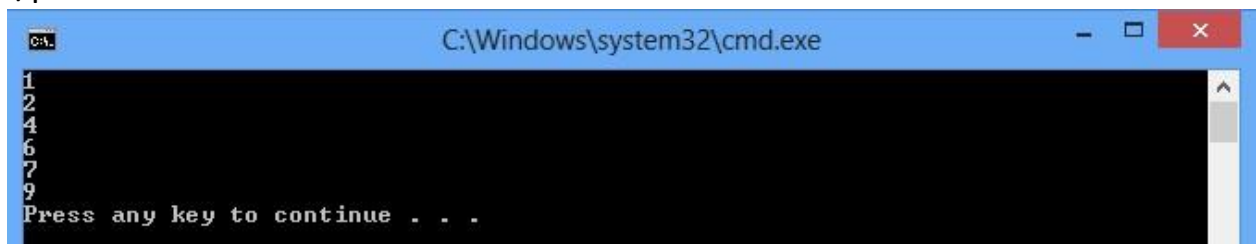
```
List<int> numbers = new List<int>() { 2,4,6,7,1,4,2,9,1};
```

Then write the following queries

[

Query1: Display numbers without any repeated Data and sorted

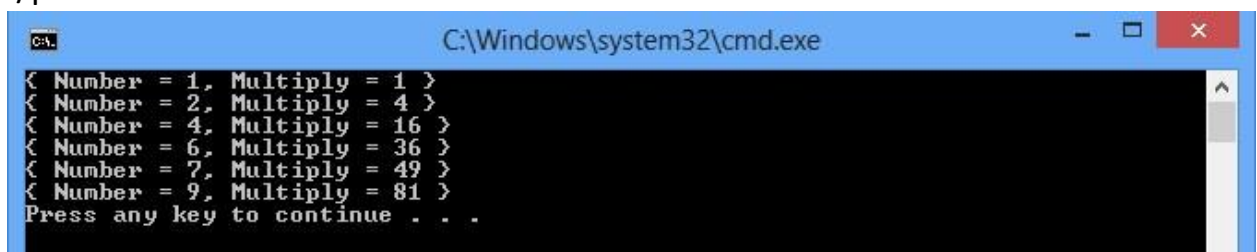
o/p should be as follow:



```
C:\Windows\system32\cmd.exe
1
2
4
6
7
9
Press any key to continue . . .
```

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

o/p should be as follow:



```
C:\Windows\system32\cmd.exe
{ Number = 1, Multiply = 1 }
{ Number = 2, Multiply = 4 }
{ Number = 4, Multiply = 16 }
{ Number = 6, Multiply = 36 }
{ Number = 7, Multiply = 49 }
{ Number = 9, Multiply = 81 }
Press any key to continue . . .
```

2- declare an array of names as shown :

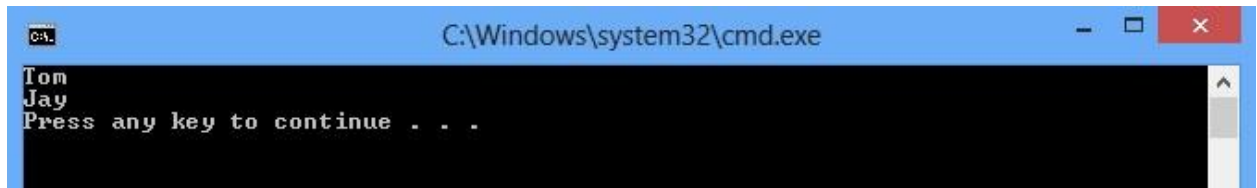
```
string[] names = { "Tom", "Dick", "Harry", "MARY", "Jay" };
```

Then write the following queries:

[USING Query Expression and method Expression]:

Query1: Select names with length equal 3.

o/p should be as follow:

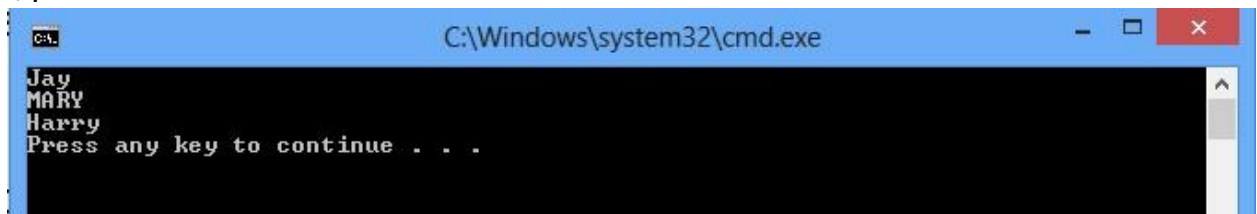


```
C:\Windows\system32\cmd.exe
Tom
Jay
Press any key to continue . . .
```

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

(Use toLower method and Contains method)

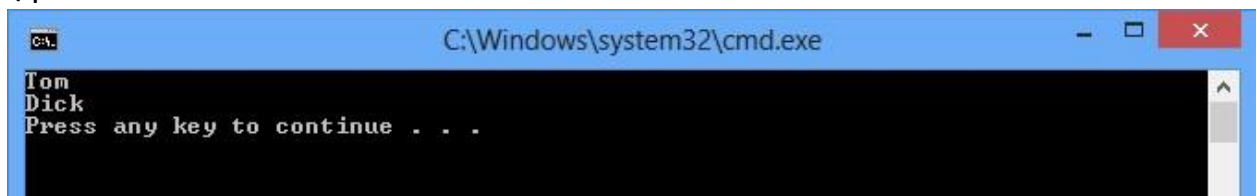
o/p should be as follow:



```
C:\Windows\system32\cmd.exe
Jay
MARY
Harry
Press any key to continue . . .
```

Query3: Display the first 2 names

o/p should be as follow:



```
C:\Windows\system32\cmd.exe
Tom
Dick
Press any key to continue . . .
```

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"}}},

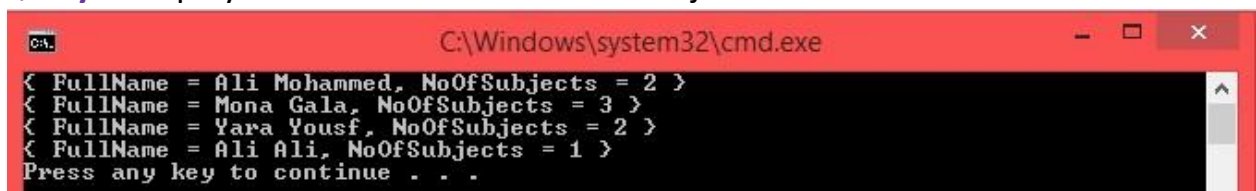
};

```

Then write the following queries

[[USING Method Chaining](#)] for each query

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

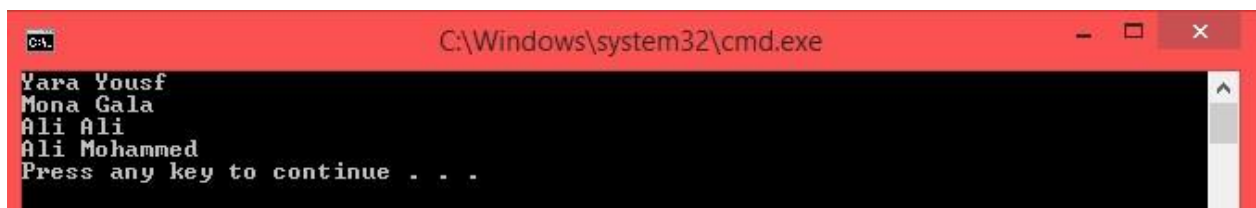


```

C:\Windows\system32\cmd.exe
< FullName = Ali Mohammed, NoOfSubjects = 2 >
< FullName = Mona Gala, NoOfSubjects = 3 >
< FullName = Yara Yousf, NoOfSubjects = 2 >
< FullName = Ali Ali, NoOfSubjects = 1 >
Press any key to continue . . .

```

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



```

C:\Windows\system32\cmd.exe
Yara Yousf
Mona Gala
Ali Ali
Ali Mohammed
Press any key to continue . . .

```

Query3:

Display each student and student's subject as follow (use selectMany)

```
C:\Windows\system32\cmd.exe

< StudentName = Ali Mohammed, SubjectName = EF >
< StudentName = Ali Mohammed, SubjectName = UML >
< StudentName = Mona Gala, SubjectName = EF >
< StudentName = Mona Gala, SubjectName = XML >
< StudentName = Mona Gala, SubjectName = JS >
< StudentName = Yara Yousf, SubjectName = EF >
< StudentName = Yara Yousf, SubjectName = JS >
< StudentName = Ali Ali, SubjectName = UML >
Press any key to continue . . .
```

Query4: Then as follow (use GroupBy)

```
C:\Windows\system32\cmd.exe

Ali Mohammed
  EF
  UML
Mona Gala
  EF
  XML
  JS
Yara Yousf
  EF
  JS
Ali Ali
  UML
Press any key to continue . . .
```