**LINQ Assignment**

1. In a array of names, find the names which contains 's'

Suma, Karthika, Jayashree, Sheela

Code:

string[] words = { "Suma", "Karthika", " Jayashree", "Sheela" };

var namesWithS = names.Select(x => x).Where(y => y.StartsWith("S"));

foreach (string name in namesWithS)

Console.WriteLine(name);

1. In a array of names, find the names which has max no of character length

Suma, Karthika, Jayashree

Code:

string[] words = { "Suma", "Karthika", " Jayashree"};

var maxLength = names.OrderBy(n => n.Length).First();

Console.WriteLine(maxLength);

1. For data source string[] words1 = { "believe", "relief", "receipt", "field" } display the output like below

believe—7

relief--6

receipt--7

field—5

Code:

string[] words = { "believe", "relief", "receipt", "field" };

var lengthOf = words.GroupBy(x => x.Length).Select( y=> new { size = y.Key, items = y});

foreach(var item in lengthOf)

foreach(var it in item.items)

Console.WriteLine(it+" -"+item.size);

1. For data source string[] words = { "believe", "relief", "receipt", "field" };

Find the words which has 'ie' word within it

Code:

var wordsContainsIE = words.Where(x=>x.Contains("ie"));

foreach(var word in wordsContainsIE)

Console.WriteLine(word);

1. For the data source string[] words = { "blueberry", "blackberry", "abacus",

"apple", "cheese", "chimpanzee" };

Display them order by their length of the string and then words by descending

Code:

var wordsOrderBy = words.OrderBy(s => s.Length).ThenByDescending(s=>s);

foreach(var word in wordsOrderBy)

Console.WriteLine(word);

1. For the data source string[] words = { "blueberry", "blackberry", "abacus" };

Display result as (reverse of the string)

Yrrebeulb

Yrrebkcalb

Sucaba

Code:

var wordsReverse = words.Select(x=>x.ToCharArray().Reverse().ToArray());

foreach(var word in wordsReverse)

Console.WriteLine(word);

1. In a array of names, display alternate names

string[]={‘C#’, ‘SQL’, ‘LINQ’, ‘Java’, “SQL”}

Code:

string[] words={‘C#’, ‘SQL’, ‘LINQ’, ‘Java’, “SQL”};

var wordsSkipOne = words.Where((value,index)=> index%2==0);

foreach(var word in wordsSkipOne)

Console.WriteLine(word);

1. Int[] numbers={1,3,2,0}

string[]={‘C#’, ‘SQL’, ‘LINQ’, ‘Java’}

Display the result as

SQL

Java

LINQ

C#

Code:

int[] num={1,3,2,0};

string[] words={‘C#’, ‘SQL’, ‘LINQ’, ‘Java’};

var wordsIndex = num.Select(x => words[x]);

foreach(var word in wordsIndex)

Console.WriteLine(word);

1. Select dept, count(\*) from emp group by dept

Finance—2

Admin—1

Trainer—1

Code:

List<Employee> employees = GetListOfEmployees();

var emp = employees.GroupBy(e => e.Dept)

.Select( x=> new {dept = x.Key, count = x.Count()});

foreach(var employee in emp)

Console.WriteLine(employee.Dept+ "--"+ employee.count);