**LAB 4**

Write a console application that inputs a sentence from the user

(assume no punctuation), then determines and displays the nonduplicate words in alphabetical order.

Treat uppercase and lowercase letters the same. [Hint: You can use string method Split with

no arguments, as in sentence.Split(), to break a sentence into an array of strings containing the

individual words. By default, Split uses spaces as delimiters. Use string method ToLower in the

select and orderby clauses of your LINQ query to obtain the lowercase version of each word.

Solution :

// Prompt the user for a sentence and display the distinct words.

using System;

using System.Linq;

public class LINQDistinctWords

{

public static void Main( string[] args )

{

// prompt the user for input

Console.WriteLine( "Please enter a sentence:" );

string sentence = Console.ReadLine(); // read input sentence

string[] words = sentence.Split(); // split sentence into words

// use LINQ to sort the words and convert to lowercase

var sortedWords =

from word in words

let lowerCaseWord = word.ToLower()

orderby lowerCaseWord

select lowerCaseWord;

Console.WriteLine( "\nYou entered:" ); // display header

Console.WriteLine( sentence ); // display the input

Console.Write( "\nDistinct words:" ); // display header

// display only the distinct words

foreach ( var word in sortedWords.Distinct() )

Console.Write( " {0}", word );

Console.WriteLine(); // output end of line

} // end Main

} // end class LINQDistinctWords

Write a console application that inputs a sentence from the user

(assume no punctuation), then determines and displays the nonduplicate words in alphabetical order.

Treat uppercase and lowercase letters the same. [Hint: You can use string method Split with

no arguments, as in sentence.Split(), to break a sentence into an array of strings containing the

individual words. By default, Split uses spaces as delimiters. Use string method ToLower in the

select and orderby clauses of your LINQ query to obtain the lowercase version of each word.

Solution :

// Prompt the user for a sentence and display the distinct words.

using System;

using System.Linq;

public class LINQDistinctWords

{

public static void Main( string[] args )

{

// prompt the user for input

Console.WriteLine( "Please enter a sentence:" );

string sentence = Console.ReadLine(); // read input sentence

string[] words = sentence.Split(); // split sentence into words

// use LINQ to sort the words and convert to lowercase

var sortedWords =

from word in words

let lowerCaseWord = word.ToLower()

orderby lowerCaseWord

select lowerCaseWord;

Console.WriteLine( "\nYou entered:" ); // display header

Console.WriteLine( sentence ); // display the input

Console.Write( "\nDistinct words:" ); // display header

// display only the distinct words

foreach ( var word in sortedWords.Distinct() )

Console.Write( " {0}", word );

Console.WriteLine(); // output end of line

} // end Main

} // end class LINQDistinctWords

Write a console application that inputs a sentence from the user

(assume no punctuation), then determines and displays the nonduplicate words in alphabetical order.

Treat uppercase and lowercase letters the same. [Hint: You can use string method Split with

no arguments, as in sentence.Split(), to break a sentence into an array of strings containing the

individual words. By default, Split uses spaces as delimiters. Use string method ToLower in the

select and orderby clauses of your LINQ query to obtain the lowercase version of each word.

Solution :

// Prompt the user for a sentence and display the distinct words.

using System;

using System.Linq;

public class LINQDistinctWords

{

public static void Main( string[] args )

{

// prompt the user for input

Console.WriteLine( "Please enter a sentence:" );

string sentence = Console.ReadLine(); // read input sentence

string[] words = sentence.Split(); // split sentence into words

// use LINQ to sort the words and convert to lowercase

var sortedWords =

from word in words

let lowerCaseWord = word.ToLower()

orderby lowerCaseWord

select lowerCaseWord;

Console.WriteLine( "\nYou entered:" ); // display header

Console.WriteLine( sentence ); // display the input

Console.Write( "\nDistinct words:" ); // display header

// display only the distinct words

foreach ( var word in sortedWords.Distinct() )

Console.Write( " {0}", word );

Console.WriteLine(); // output end of line

} // end Main

} // end class LINQDistinctWords

Write a console application that inputs a sentence from the user

(assume no punctuation), then determines and displays the nonduplicate words in alphabetical order.

Treat uppercase and lowercase letters the same. [Hint: You can use string method Split with

no arguments, as in sentence.Split(), to break a sentence into an array of strings containing the

individual words. By default, Split uses spaces as delimiters. Use string method ToLower in the

select and orderby clauses of your LINQ query to obtain the lowercase version of each word.

Solution :

// Prompt the user for a sentence and display the distinct words.

using System;

using System.Linq;

public class LINQDistinctWords

{

public static void Main( string[] args )

{

// prompt the user for input

Console.WriteLine( "Please enter a sentence:" );

string sentence = Console.ReadLine(); // read input sentence

string[] words = sentence.Split(); // split sentence into words

// use LINQ to sort the words and convert to lowercase

var sortedWords =

from word in words

let lowerCaseWord = word.ToLower()

orderby lowerCaseWord

select lowerCaseWord;

Console.WriteLine( "\nYou entered:" ); // display header

Console.WriteLine( sentence ); // display the input

Console.Write( "\nDistinct words:" ); // display header

// display only the distinct words

foreach ( var word in sortedWords.Distinct() )

Console.Write( " {0}", word );

Console.WriteLine(); // output end of line

} // end Main

} // end class LINQDistinctWords