Contiguous Numbers (Page 1 of 2)

Dear Applicant,

This coding exam is accompanied with an **incomplete** Visual Studio 2015 solution. Use the instructions in this document to complete the solution. Then return it to me at [ffrancis@security101.com](mailto:ffrancis@security101.com). The purpose of this coding exam is to primarily test your LINQ skills using C#. The solution is a console application that allows a user to enter a set of numbers, delimited by commas, at the command prompt. The user must enter at least three numbers. The application will then be able to determine the maximum sum that can be obtained from any three consecutive integers. Once the method has determined those three integers, the method will display them on the screen.

For example, if a user enters the numbers 1, 2, 6, 2, the application would return the values 2, 6, 2 because those three numbers yield that maximum sum of 10. As another example, if a user enters the numbers 1, 2, 6, 2, -20, 5, 9, 3, -10, 3, 5, 1, the application would return the value 5, 9, 3. The sum of 5, 9, 3 is 17 and is the maximum sum that can be obtained from adding any three consecutive numbers. If more than one set of numbers produce the same sum, then provide the first set as the answer. The application should ignore any element in the CSV list that it cannot convert into an integer.

The output of the application is shown below.

|  |
| --- |
| Enter numbers: 1 2 6 2 -20 5 9 3 -10 3 5 1  The three integers that yield the maximum sum are: 5, 9, and 3. |

The implementation of the application is pseudo-coded on the following page. You may copy that code and use it as a starting point for your application. If you are unable to use LINQ in your implementation, then use what you do know. If you do not have Visual Studio 2015, create your own solution using whatever Visual Studio version you have. I have included the pseudo-code on the following page.

Contiguous Numbers (Page 2 of 2)

using System;

using System.Linq;

namespace ConsoleApplication1

{

class Program

{

static void Main(string[] args)

{

Console.Write("Enter numbers as CSV: ");

// Read the entry using the Console.ReadLine() method

// Convert the entry into an array of strings.

// Use the name "entries" for the variable that stores the string

// array object.

// Use a LINQ-to-Object statement to convert the string array to

// an integer array.

// Use the name "numbers" for the variable that stores the integer

// array object.

// The following line exists so that this program will compile.

int[] numbers = { 1, 2, 6, 2, -20, 5, 9, 3, -10, 3, 5, 1 };

// Use another LINQ statement to get sets of three consecutive integer.

// Hint 1: You will have to generate a collection of indices that

// exists in the integer array.

// The Enumerable class has a Range method that can do that task for you.

// Hint 2: This latter LINQ statement makes use of the following

// operators:

// Select, Skip, Take, Sum, OrderByDescending, ThenBy, First.

// Store the result of the latter LINQ query in a variable called

// "result".

// This variable is an anonymous type. However, it will require that it

// has a integer property called "Index". That property tells you what

// index to use

// in the original integer array.

// The following line exists so that this program will compile.

var result = new { Index = 5, Sum = 17 };

//Your final statements are written below.

Console.WriteLine

(

"The numbers {0}, {1}, and {2} will yield the maximum sum of {3}.",

numbers[result.Index],

numbers[result.Index + 1],

numbers[result.Index + 2],

result.Sum

);

Console.ReadKey();

}

}

}