Task: Recursion

www.hyperiondev.com

Introduction

Overview:

A few tasks ago we wrote code that made use of functions. So you might need to revisit this task before attempting this task.

In this task, you are going to be introduced to the concept of Recursion. In simple terms, recursion is when a function calls itself. Normally a recursive function uses conditional statements to call the function recursively or not. The main benefits of using recursion are compact code, easily understandable code and less variables used. Recursion and iteration can be used to achieve the same results.

The following guideline will help you to decide on which method to use depending on a given situation.

When to use recursion? Where compact, understandable and intuitive code is required.
When to use iteration? Where there is limited memory and faster processing is required.

The following example shows where recursion can be be used instead of iteration to calculate the Fibonacci sequence.

-The Hyperion Team

A note from the Hyperion Team...

Instructions

- Open the Recursion.sln file in the folder Recursion and read its contents. Make sure you read all of the comments and try your best to understand them.

- You may run the project to see the output. The instructions on how to do this are inside the file. Feel free to write and run your own example code before doing the tasks to become more comfortable with C#.

- Instructions on how to complete your compulsory tasks are below.

Compulsory Task 1

Follow these steps:

NOTE: Make a copy of this folder on your computer. Submit the required files when you done

Open the C# file called *recursion.sln* within your Task folder and do the following:

1. Complete the Factorial function using a recursive function where applicable in the file.

2. The C# String library determines the length of a string by counting the number of characters before the '\0' (The null termination character). This function can also be implemented using recursion. Your task is to implement this using recursion. Complete the code for the string_length function where applicable in the file.

3. Complete the GCD function below to calculate the greatest common divisor (GCD) for two numbers x and y using recursion. GCD is the largest positive integer that divides the numbers without leaving a remainder.

Hints:
1. Factorial formula is n! = n * (n-1) * (n-2)*...* 2 * 1

Still need help?

Just write your queries in your comments.txt file and your tutor will respond. Alternatively you can email us on help@hyperiondev.com.

Task Statistics

Last update to task: 19/02/2016.
Authors: Richard Niescior & Brandon Haschick
Main trainer: Umar Randeree.
Task Feedback link: Hyperion Development Feedback.