Assignment 2

CS321 Winter 2023, Professor Christopher Diggins

Overview

This assignment involves practicing writing algorithms focusing on lists and arrays. You will do this by replacing the stub implementations of the algorithm in the C# project at https://github.com/cdiggins/cs321/tree/main/code-examples/cs321/Assignment2.

This project contains:

- a set of unit tests in the file "ListTests.cs"
- stub implementations in the file "ListOperations.cs"
- reference implementations in the file "ListOperationsReference.cs"
- some helper code in the file "Helpers.cs"

When you run the tests from the test explorer, you should see that they all fail. Your assignment is to modify the functions in file ListOperations.cs so that the given tests pass.

You should not use any system libraries except for:

- IList<T>.Count
- IList<T>.Add()
- IList<T>.[]
- IReadOnlyList<T>.Count
- IReadOnlyList<T>.[]
- object.Equals()
- object.ToString()
- IComparable<T>.CompareTo()

Submission

1. Screen shot of test results in test explorer as PNG files (using multiple screen shots if necessary)
2. All source code files (*.cs)
3. The project file (*.csproj)
4. Nothing else!

Grading Criteria

1. Tests passing – **5 points** (half points rounded up, so passing more than 80% give you full points)
2. Follow the Style Guidelines – **2 points**
3. All functions implemented – **1 point**
4. Reuse of functions – **1 point**
5. No usage of system libraries – **1 point**
6. Bonus (logic operators) – **1 point**
7. Bonus (all tests pass) **– 1 point**
8. Total: **10 Points + 2 Bonus Points**

How to get the code

- Download the file from Github as a zip
- Use Visual Studio (https://visualstudio.microsoft.com/vs/github/)
- Use Github desktop (https://desktop.github.com/)
- Use Git for windows (https://gitforwindows.org/)

Logic Operators Bonus Question:

- Implement the following functions but instead of using the built-in operators (as shown below), use only the conditional ternary operator:
  - bool And(bool a, bool b) => a && b;
  - bool Or(bool a, bool b) => a || b;
  - bool XOr(bool a, bool b) => a ^ b;
  - bool NAnd(bool a, bool b) => !(a && b);
  - bool NOr(bool a, bool b) => !(a || b);
  - bool NotA(bool a, bool b) => !a;
  - bool NotB(bool a, bool b) => !b;
- Write tests for each of the functions yourself.
- Put both the implementations and the tests you write in a new file "LogicTests.cs"