HW04 Problem Set

CS-3020

Assignment
Deitel & Deitel Exercise 9.4, 9.5, 10.10

HW04-1: (Deitel & Deitel Exercise 9.4)

9.4 *(Duplicate Word Removal)* Write a console app 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.]

HW04-2: (Deitel & Deitel Exercise 9.5)

9.5 *(Sorting Letters and Removing Duplicates)* Write a console app that inserts 30 random letters into a `List< char >`. Perform the following queries on the `List` and display your results: [*Hint:* Strings can be indexed like arrays to access a character at a specific index.]

a) Use LINQ to sort the `List` in ascending order.
b) Use LINQ to sort the `List` in descending order.
c) Display the `List` in ascending order with duplicates removed.

HW04-3: (Deitel & Deitel Exercise 10.10)

10.10 *(`HugeInteger` Class)* Create a class `HugeInteger` which uses a 40-element array of digits to store integers as large as 40 digits each. Provide methods `Input`, `ToString`, `Add` and `Subtract`. For comparing `HugeInteger` objects, provide the following methods: `IsEqualTo`, `IsNotEqualTo`, `IsGreaterThan`, `IsLessThan`, `IsGreaterThanOrEqualTo` and `IsLessThanOrEqualTo`. Each of these is a method that returns `true` if the relationship holds between the two `HugeInteger` objects and returns `false` if the relationship does not hold. Provide method `IsZero`. If you feel ambitious, also provide methods `Multiply`, `Divide` and `Remainder`. In the `Input` method, use the `string` method `ToCharArray` to convert the input string into an array of characters, then iterate through these characters to create your `HugeInteger`. [*Note:* The .NET Framework Class Library includes type `BigInteger` for arbitrary sized integer values.]

Grading Rubric
Each problem is worth 10 pts (score will be recorded as a percentage of that amount)

10% Properly submitted
10% Properly named
20% Adequate comments
10% Runs
20% Produces correct output
30% Effort evidenced by the submitted work