CSD3354 C Sharp Algorithms and Data Structures

Assignment: Calculating the Numerology of Names

`https://www.wikihow.com/Calculate-Your-Name-Number-in-Numerology`

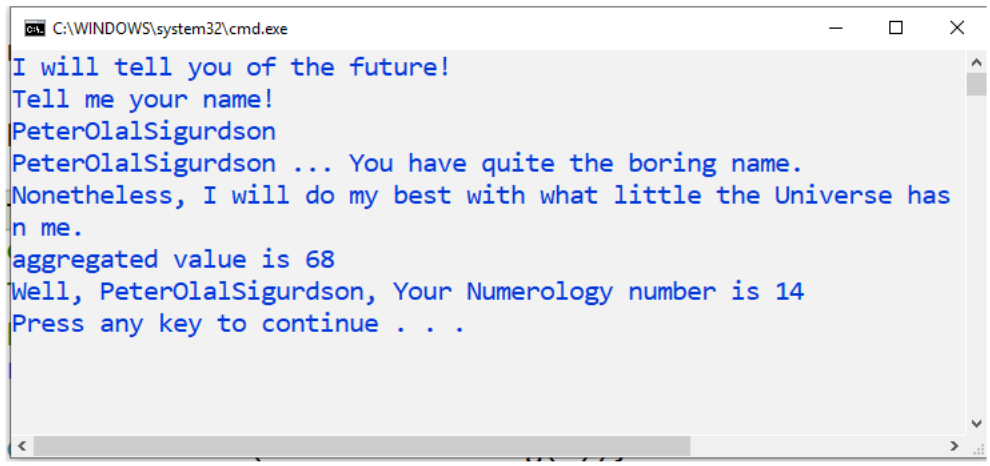
1 Learning Outcomes:

1. Understand how to create an Algorithm by converting a description of a process into METHODS
2. Learn to use the Dictionary Data Structure
3. Learn to use the Switch Statement for selecting a path of execution based on a Condition

What you are to do in this Assignment:

1. Construct a Program that obtains the user's name
2. Tell them their Numerology Number
3. Many cultures believe that your Beloved - the One you are meant to be with - will have the same numerology as you. You will need to test this for yourself!!
4. Next week for Assignment 5, you will wrap this POCO (Plain old C Object) functionality up in a Web Services API. Other users can connect to your server and be presented with a Web Page to enter their information and receive their Numerology.

Sample Output:



2 Suggestions:

2.1 Prompt the user for their name

Store this information!

2.2 Prompt the user for their name

Store this information!

2.3 Create a data structure to correlate the letters of the Alphabet to their numerical positions.

2.4 Implement the numerology algorithm for Names.

Sample code for the Dictionary Data Structure:

```
class Playground
{
    public void Run()
    {
        Dictionary<string, int> Gematria = new Dictionary<string, int>()
                                            {
                                                {"A",1},
                                                {"B", 2},
                                                {"C",3}
                                            };

        Console.WriteLine(Gematria["A"]);
    }
}
```

3 Gematria Algorithm:

1. Write out the alphabet from 'A' to 'Z.' On a piece of paper, write out all 26 letters in a horizontal line. Each letter is going to be assigned a different numerical value. You can also write the letters vertically; as long as they are in order and organized you can choose either direction. We did this with our Dictionary Data Structure!

2. Assign each letter a digit from one to nine. Start with the letter A. Write a '1' next to it and then give each following letter the next number as you go in numerical order. For example, B will have a 2, and C is 3. Once you reach 'I' which is a '9,' start back at 1 as you continue along the alphabet.

3. The system can also be summarized like this:

1

– A, J, S

2

– B, K, T

3

– C, L, U

4

– D, M, V

5

– E, N, W

6

– F, O, X

7

– G, P, Y

8

– H, Q, Z

9

– I, R

This can best be implemented with a Case, or Switch, Statement

https://www.c-sharpcorner.com/article/c-sharp-switch-statement/

```
Sample Code to implement the assignment of a Letter to a Number
with the required Rule Set:

public int ReturnCyclicNumberPosition(string InputLetter)
{
    switch (InputLetter)
    {
        case "A":
            return 1;
        case "J":
            return 1;
        case "S":
            return 1;

        default: return 0;
    }
}
```

Write out your full name. To find out your true name number, you will need to use your full name. You can find your entire name on your birth certificate or an official identification document. Don't forget to include your middle name if you have one as well.

Match each letter in your name to its corresponding number. Now that each letter has a numerical value, you can start putting numbers with the letters in your name. Below where you have written your name, write each number that goes with each individual letter.

3.1 You will have some duplicates, but that is not a problem.

3.2 For example, if your name is John Jacob Smith, then all of the Js will get a 1, the Os will get a 6, the Hs will get an 8, and so on.

4 Adding Together the Numbers

Add together all of the letter's numbers. Using a calculator or a pencil and paper, add together every single digit from your name. If your name has 20 letters, you will be adding together 20 individual numbers. You will end up with a two-digit sum after adding everything together.

4.1 For example, BATMAN is numbers 2+1+2+4+1+5, which equals 15.

5 Reduce the sum of your name's numbers into a single digit.

After you have added your numbers together, you will have a double or triple digit sum, if your name is long. To reduce the sum, add together the two digits within it. For example, if the sum of your letters is a 25, split the 25 and add 2+5 to equal 7. The 7 is your true name number.

6 Leave the Master Numbers as double digits.

If you add up the letters in your name and you get a sum that equals either 11, 22, or 33, don't reduce them. These are the three Master numbers, which can add difficulty but also depth to the personality you are discovering or the topic of numerology you are studying. These three numbers have their own personality explanations.[3] Master numbers can be reduced, but only in certain situations. They are reduced when they are found within a date or number equation. For example, if a total sum is a Master number do not reduce it. But if there is a Master number within the equation, go ahead and reduce an 11 to a 2 or a 33 to a 6 to simplify the equation.[4]

Grading Rubric:

25%	You will submit to the Work Handin Database a the URL for your Assignment. Part of your grade will be demonstrating frequent GITHUB Updates as proof of work. Your GitHub Repository should be PRIVATE and the Repository name should be Numerology. Remember to add Peter@PeterSigurdson.net as a Collaborator.
50%	Your implementation should actually work, or at least be structured well and going in the right direction.
25%	Your work should be properly packaged and handed in on time.