Exercise: Concurrency Patterns – The MapReduce desgn pattern

In this exercise you will become acquainted with the MapReduce pattern as implemented in the text for the subject, i.e. using PLINQ queries to implement MapReduce. Through this, you will become familiar with the four steps in MapReduce, Distribute, Map, Group and Reduce.

You are provided with an example of using the MapReduce algorithm, consisting of a source code implementation and a set of books (in .txt format). This example will count how many words from a set of books have a given length. By investigating and manipulating this example you will solve the exercise.

Exercise 1:
Investigate the provided example. Make sure you understand the parts of the algorithm, namely Source() Map(), and Reduce(). Run the example – do you understand what the individual steps of the MapReduce() algorithm does?

Exercise 2:
Change the example so that it counts the number of occurrences of each word. This requires you to change the Map() and Reduce() functions, but not MapReduce()

Exercise 3:
You are provided with a set of files each representing a huge deck of playing cards. Use the MapReduce pattern to answer the following question:

- What is the sum of card values of the individual suits?

Exercise 4:
Implement a sequential program which answers the same question as in Exercise 3. Disregarding the time it takes to actually read the files from disk (which is a sequential operation), which is the faster – the sequential implementation or MapReduce()?