CS1083 Assignment #2 - Winter 2021

Due: Friday, January 29th before 4:00pm (Atlantic), submitted in the Assignment 2 dropbox in Desire2Learn. (Read the submission instructions at the end of this document carefully).

The purpose of this assignment is to provide you with more practice with loops and arrays.

This assignment is to be done individually. If you have questions, direct them to a tutor/assistant during a help session in the "Faculty of Computer Science Student Success Centre" team or to your course instructor.

1. Operations on Sorted Arrays:

The purpose of this exercise is to provide more practice with working with loops and arrays. Here you will be writing three methods that operate on sorted arrays: finding the number of duplicate elements when comparing two arrays, merging two sorted arrays into one sorted array, and adding an element to an existing array such that the ordering is preserved. The arrays you will be creating and working with will be filled arrays (not partially filled arrays).

You are provided with three java source files in D2L. One, **Book.java**, is a class that represents a book in someone's personal library and provides methods to access attributes, and change the number of times the book has been read. The ISBN, title and author of the book cannot be changed. The second, **Library.java**, represents a personal library of books, and provides constructors to create a library given an array of books or create one by reading input using a Scanner object. In the latter case the input format consists of a line with the number of books, followed by a line for each book containing values separated by commas (see example below). The library is **sorted in ascending value of the ISBN (International Standard Book Number) of the books** (these are unique for each book). Finally, **TestLibrary.java** is a simple test driver that will test each of the new methods.

Your task is to implement three methods in Library.java, and test them appropriately.

1. **public int findUnique(Library other)**
   Returns the number of books that occur in one library but not the other; this library and the other one (passed as a parameter). Use the fact that the libraries are sorted to do this efficiently.

2. **public Library merge(Library other)**

   Merges this library with another one, producing a new sorted library. If the same book appears in both libraries, then it appears twice in the merged library (i.e., duplicates are not removed and the order of duplicates does not matter).

3. **public void addBook(Book bookIn)**

   Adds a book to this library, in the appropriate location so that the library remains sorted in ascending order by the ISBN of the books. If you add a book that is already in the course, then they will appear twice (the order of duplicates does not matter).

To receive full marks you must create an efficient solution that takes the fact that each list is sorted into account.

Example

Consider the following input of two libraries (myLibrary and yourLibrary) and an additional book at the end:

```
5
9780006716693,The Last Battle,Lewis,2
9780007591855,The Silmarillion,Tolkien,3
9780048232298,The Lord of the Rings,Tolkien,1
9780385491037,The Robber Bride,Atwood,3
9781400079179,The Da Vinci Code,Brown,2
3
9780048232298,The Lord of the Rings,Tolkien,1
9780062230621,The Confidence Code,Kay,0
9780385491037,The Robber Bride,Atwood,3
9780071079170,The Skin We're In,Cole,5
```

After the test driver is completed and run with this input, the following should be produced:

```
9780006716693   The Last Battle         Lewis   2
9780007591855   The Silmarillion        Tolkien 3
9780048232298   The Lord of the Rings   Tolkien 1
9780385491037   The Robber Bride        Atwood  3
9781400079179   The Da Vinci Code       Brown   2

9780048232298   The Lord of the Rings   Tolkien 1
9780062230621   The Confidence Code     Kay     0
9780385491037   The Robber Bride        Atwood  3

Number of unique books: 4
My Library after adding a book:
9780006716693   The Last Battle         Lewis   2
9780007591855   The Silmarillion        Tolkien 3
9780048232298   The Lord of the Rings   Tolkien 1
9780071079170   The Skin We're In       Cole    5
9780385491037   The Robber Bride        Atwood  3
9781400079179   The Da Vinci Code       Brown   2

Merged Libraries:
9780006716693   The Last Battle         Lewis   2
9780007591855   The Silmarillion        Tolkien 3
9780048232298   The Lord of the Rings   Tolkien 1
9780048232298   The Lord of the Rings   Tolkien 1
9780062230621   The Confidence Code     Kay     0
9780071079170   The Skin We're In       Cole    5
9780385491037   The Robber Bride        Atwood  3
9780385491037   The Robber Bride        Atwood  3
9781400079179   The Da Vinci Code       Brown   2
```

NOTE: Include an @author tag with your name and student ID in the Javadoc comments for your completed Library class.

2. Testing

Using the test driver provided, test your completed **Library.java** program with 5 different test cases (i.e. run the **TestLibrary** class 5 times, with different input each time.) Choose test cases that give you good coverage (i.e. test different scenarios). The example that is provided above may be used as one of the 5 test cases.

Recall: you can use the redirection operator **<** to send input data to your program from a text file. So, you may find it convenient to create text files to hold the input data for each of your test cases (rather than having to type it all in over and over again each time you run your program).

Similarly, you can use the other redirection operator **>** to send output to a text file from your running program.

When you copy and paste your test input and corresponding results (output) into your report, be sure to explain why each test case that you chose to include is important.

Your electronic submission (submitted via Desire2Learn) will consist of two files:

i. a written report. This should begin with a title page that includes: the course (CS 1083), your section (FR01B, FR02B, FR03B), the assignment number, your full name, and your UNB student number. That should be followed by two sections, with each part clearly identified with a section heading. Include:

   a. the completed source code for the Library class (`Library.java`),

   b. input and associated output for all 5 test cases,

   c. an explanation of each test case.

   This written report should be prepared using a word processor; we recommend using Microsoft Word (i.e. create a .docx file for your report). Copy & paste your java source code & required output into the report document. Add appropriate headings for each part. Fix up the formatting where necessary, adjusting line breaks & page breaks to ensure that your document is easy to read. Use a monospaced font for your code to maintain proper indentation.) Once the report is complete and you've checked it all over, save the .docx file for your own records, and then **save a second copy in pdf format for submission**. (Note: Be sure to open that file in a pdf viewer to verify that the pdf was generated correctly.) **The SINGLE pdf file containing your report will be submitted to the appropriate assignment drop box on Desire2Learn**. (It is important that you submit a pdf file and NOT the original Word document. This pdf will allow the marker to write comments directly on your work to give you better feedback.)

   Note: name this report as follows: **YourName_CS1083_As2_Report.pdf**

ii. an archive file (**.zip**) that contains your Java source code and input files for this assignment. Make sure that your archive includes all .java files (in case the marker wishes to compile & run your code to test it). You should not include the report document or the .class files in your archive. This archive should be submitted as a **single file** to the appropriate drop box on Desire2Learn.

Note: name this archive file as follows: **YourName_CS1083_As2_Archive.zip**