MPP Final Exam

Programming Hints:

```
//To convert an integer array to List<Integer>, use
Integer[] arr = {2,4,6};
List<Integer> list1 = Arrays.asList(arr);

//To generate hash code for int[], use
int[] arr2 = {2,4,5};
int hashcode = Arrays.hashCode(arr2);

//To covert an int[] to String, use
int[] arr3 = {1,4};
String str = Arrays.toString(arr3);
```

Problem 1. [Lambdas and Streams]

In prob1 package, a few classes have been provided to you which model part of a library system. A LibraryMember has first name, last name, age, address and a list of book copies he/she has checked out.

Each book copy has unique copy number and its availability. And a book may have multiple copies. A TestData class has been provided to you. You can get library member data and book data from the two methods below respectively.

```
public static List<LibraryMember> getLibraryMembers(){
            loadLibraryMemberData();
            return members;
}
public static List<Book> getAllBooks() {
            return allBooks;
}
```

Provide your solution to Part A – D in either Main class or LambdaLibrary class depending on the requirement.

NOTE: None of the expressions in LambdaLibrary are allowed to have hard-coded values.

A. [6 points] Create a stream pipeline that prints to the console the full names (first name and last name separated by a space) of Library Members who live in the state of CA and whose first name starts after the letter 'M', sorted by age in ascending order.
B. [4 points] Turn the stream pipeline from Part A into a LambdaLibrary item and evaluate it in the Main class.
C. [4 points] Replace all lambda expressions in the stream pipeline from Part A with instances of inner classes/functors.
D. [**Extra Credit** 4 points] Create a stream pipeline that uses the reduce method to get the Library member who has checked out the greatest number of book copies. You may assume there is exactly one such member.

Problem 2. [Unit testing stream pipelines – 10 points]

In prob2 package, there are classes Main and Customer. The main method in Main loads a list of Customers and then attempts to print, in sorted order, the names of those Customers who live in Fairfield.

This question asks you to refactor the stream pipeline in the main method so that it can be unit tested, as in the Complex case mentioned in the class. Do the following:

Refactor the lambda expressions in the stream pipeline with method references so that method references are used to call auxiliary methods. Now in order to test the stream pipeline, we can test the auxiliary methods instead. Create a Unit test that tests these auxiliary methods.

Problem 3. [Problem Solving – 10 points]

In a certain startup company, the owners decided to make the process of paying their employees as simple as possible by providing to each employee one of three possible monthly salaries: $5000, $6000, $7000. Once in awhile, an employee who currently receives one of the lower two salaries, will get a raise; when this happens, the raise is always in the amount of $1000. So, for example, if an employee is receiving $5000 per month and he gets a raise, then he will start to receive $6000 per month from then on.

The IT Department wishes to track the salaries of employees on a quarterly basis. Code has been provided for you that shows an implementation of this. There is a QuarterlySalary class that contains a list of possible salaries for a 3-month period.

The QuarterlyReport class provides a HashMap that matches each possible QuarterlySalary with a list of employees whose salaries over a particular quarter match this QuarterlySalary. For instance, if a QuarterlySalary consists of [5000, 5000, 6000] and Employee "Joe" has had monthly salaries of 5000, 5000, and 6000 over the past 3 months, then this QuaterlySalary will be a key that is matched with "Joe" in the HashMap.

The implementation, however, does not work. When you try to look up all employees matched with the QuarterlySalary consisting of [5000, 5000, 6000], nothing is found.

Explain why the implementation does not work and then fix it. Place your explanation in a text file readme.txt in the prob3 package, and modify the code in prob3 so that the main method produces a correct output.

Problem 4. [Generic Programming - 15 points]

In the prob4 package, a class Product has been provided for you. It has a method `computeProduct` which takes two int arrays A, B as input and outputs all possible pairs (a,b) for which a belongs to A and b belongs to B. Here, a "pair" is a 2-element array. Therefore, the output of the `computeProduct` method is a two-dimensional, and each row of this 2-D array has length 2. See the output of the main method in Product for an example.

Generalize the `computeProduct` method as much as possible to List types. Your method should accept two lists L, M of any type and output a list of pairs (a,b), where a belongs to L and b belongs to M. You must represent "pairs" using a generic Pair class. A non-generic Pair class has been provided for you; you may make any changes you want to this Pair class.

Problem 5. (**SCI – 3 points**) Write one or two paragraphs relating a point from the course to a principle from SCI. Write your answer in answer.txt in the sci package.