Notice

- Follow Java Code Conventions; otherwise you cannot get full points. The requirement on naming (section 4) and indentation (section 9) are hard. Other parts are suggestive.
- For each question, you submission should contain a .java source file and a screenshot of its execution. If you don't know how to take a screenshot, this website may help.
- Submit on learn@polyu, no hard-copies are required.
- If you use your free coupon by submitting a text file stating "I'm using my free coupon and I'll submit this assignment later" to learn@polyu before **October 1, 2015 11:59pm**, the deadline of this assignment will be extended to **October 8, 2015 11:59pm**. You have only one coupon, so be wise to use it.

Mandate questions

1. (25 points) Translate the pseudo-code I used in lecture 3 for binary search into a Java method, and test it.

   Hint: you may reuse the `SortingTester.java` in the lab 2.

   Warning: you'd better write it before checking online, otherwise your code will be surely similar to those online, and fail the plagiarism test.

2. (25 points) Finish `BinarySort.java`. You are ask to sort the students according to their gender, which can be either 0 (female) or 1 (male), such that all female students come before male students. Your algorithm must run in linear-time $O(n)$, and you are *not* allowed to use another array. The extra storage used by your algorithm, besides the given array, should be no more than constant size (this is also called *in place* sort).

Challenging questions

You're suggested to try these to get bonus points and better understanding.

1. Modify the binary search program you implemented such that if there are more than one elements of the same value, return the one in the middle of them. For example, if you're asked to find a queen from 13 cards that are all queens, you should return 7. Your algorithm should still run in $O(\log n)$ time.

2. Given an array of integers, you are asked to give an algorithm to find the second smallest of them with at most $n + \log n - 2$ comparisons. Note that this requirement is stronger than linear-time: We're not playing Big-Oh here, and $2n$ comparisons wouldn't count.

   Hint: divide and conquer to find the smallest; where is the second smallest?