1. (5 pts) Fill in the homework header properly—this helps us keep the grading pipeline flowing so that you get credit for your work and get feedback more quickly.
   - writing either 4, 5, or 6 to indicate your discussion section (lab) meeting time
   - entering BOTH your name AND your umail address EVERY time.

   **Paper submissions:** One sheet of 8.5x11 paper double sided, or two DISCONNECTED SHEETS with your name on EACH. Please: NO STAPLES, NO PAPERCLIPS, NO TAPE, NO ATTACHMENT OF ANY KIND. These damage the document scanner.

   **Scanned submission:** When submitting by PDF upload: scan your pages legibly and SCAN IN THE CORRECT ORDER. Page 1 first, then Page 2, in the correct orientation. Failure to scan properly may result in zero credit, meaning you "use up" one of your five "drop the lowest grade" slots.

2. (10 pts) Suppose you have a variable `ArrayList<String> words;` which has already been instantiated filled with a list of words. Now you want to sort those words in alphabetical order. Write one line of code that will do this, using the technique described in Chapter 16.
3. The declaration of the sort method in the Collections is:

   public static <T extends Comparable<? super T>> void sort(List<T> list)

   Here's that declaration again, several times, with a part underlined each time. Explain the meaning of the underlined part.

   a. (10 pts)  public static <T extends Comparable<? super T>> void sort(List<T> list)

   b. (10 pts)  public static <T extends Comparable<? super T>> void sort(List<T> list)

   c. (10 pts)  public static <T extends Comparable<? super T>> void sort(List<T> list)

4. (5 pts) In the Java Collections API, List, Set and Map are defined as:  (Circle one):  classes interfaces