1. First Task
   a. Write a program that will read in a series of Words from a text file. Copy a large paragraph of text from the interwebs and save to a text file. Read all of the Words in from the text file and store all of the Words in a Queue and a Priority Queue.

      You must create a Word class object to actually hold the string value of the Word.

      The Priority Queue needs the following queuing discipline:

      - Words in the Priority Queue need to be arranged in alphabetical order (ascending).

      Both the Queue and the Priority Queue words must be displayed next to each other as follows:

1st queue word	1st PQ word
2nd queue word	2nd PQ word
3rd queue word	3rd PQ word
4th queue word	4th PQ word
5th queue word	5th PQ word
.	.
.	.
.	.

   b. Modify code slightly so that the words in the Priority Queue are arranged in reverse alphabetical order (descending).

2. Task 2
    a. Create Passenger object class with following methods:
        - getLastName() returns last name of passenger
        - getFirstName() returns first name of passenger
        - flightCity() returns city passenger is flying to
        - flightTime() returns time passengers plane is scheduled to depart
        - etdCalc() returns number of hours and minutes until flight departure
        - compareTo() – priority is based on smallest etdCalc() values
          However, if etdCalc() is more than 1 hour, passengers are added to list as a normal queue (there is no priority)

    b. Given the passenger list information file and a current time of 9:03 AM, fill a queue of Passengers. Upon completion, output the list of passengers and display the following:
        Last Name, First name – Flight City – Flight Time – etdCalc

    c. Given the passenger list information file and a current time of 9:03 AM, construct a PriorityQueue of Passengers that will allow the passengers that have impending departures (flights leaving in less than an hour) to have their names posted (last name first) so that they may cut line and have their bags checked-in before they miss their flight.

        In "runner"
        i. Fill a PriorityQueue with Passengers
        ii. Display passengers that must be moved to head of the line with the following syntax:
            Last Name, First name – Flight City – Flight Time – etdCalc