2. Binary Search Tree, 4 points. Draw a picture below of the binary search tree that would result from inserting the following words into an empty binary search tree in the following order: France, Canada, Italy, USA, Germany, England, Japan. Assume the search tree uses alphabetical ordering to compare words.

7. Comparable class, 20 points. Define a class called AdmissionsEntry that keeps track of information for an admissions candidate, how that candidate is rated by reviewers (real numbers between 0.0 and 5.0), and whether or not a candidate will be discussed. The class has the following public methods:

AdmissionsEntry(id) constructs an AdmissionsEntry object with given ID
rate(rating) records a rating for the candidate
flag() indicates that the candidate should be discussed
getRating() returns the average rating (0.0 if no ratings)
toString() returns a String with ID and average rating

Below is an example for a candidate that has been reviewed four times:

    AdmissionsEntry entry = new AdmissionsEntry("2222222");
    entry.rate(3.75);
    entry.rate(3.65);
    entry.rate(3.8);
    entry.rate(3.75);
    entry.flag();

After these calls, the call entry.getRating() would return 3.7375 (the average of the ratings). The toString method should return a string composed of the ID, a colon, and the average rating rounded to 2 digits after the decimal point ("2222222: 3.74" for this example). If there are no ratings, then getRating and toString should indicate a rating of 0.0.

Each AdmissionsEntry object should keep track of whether that candidate should be discussed by the admissions committee. Any candidate who receives an individual score of 4.0 or higher should be discussed even if their average rating is below 4.0. Notice also that the flag method can be called, as in the example above, in which case the candidate will be discussed even if none of the ratings are 4.0 or higher.

The AdmissionsEntry class should implement the Comparable<E> interface. Define the method so that when sorted, a list of entries will have students to be discussed appearing first followed by students not to be discussed. Within those groups, students with higher average ratings should appear earlier in the list. Students with the same discussion status and the same average rating should appear in increasing order by ID. Recall that values considered "less" appear earlier in a sorted list.