ACS-2947-050
Lab #7
Due by Friday November 23 at 11:59 pm
Submit your `.java` files to 2947L-070@acs.uwinnipeg.ca or 2947L-071@acs.uwinnipeg.ca

Recall that In lab 3 we used a stack to test if a word is a palindrome (reads the same forward and backward). In this lab you will write a program that uses a map to test whether or not the letters forming a string can be permuted to form a palindrome.

E.g.

- omm can be permuted to form mom
- edified can be permuted to form deified
- dobby has no palindrome permutations

Use an unsorted tablemap in your solution. Using the `Map` and `Entry` interfaces, produce the code for the the `AbstractMap` class that provides the base for the `UnsortedTableMap` given here.

Create a driver class called `LastLab` with a static method `canFormPalindrome` that returns the boolean result of this test. Display the results of the examples above.

Try to solve this problem with minimal hints!

Hint #1
Hint #2
Hint #3

EXTRA WORK: Do not submit

A generic method for `insertionSort` is given here. Add the generic methods `selectionSort` and `bubbleSort` to this class. Illustrate the use of these methods by

a. using the default comparator from As3 and sorting the integers 8 6 7 5 3 0 9.
b. using the TeamComparator and Team objects from Lab 6

Hint #1

All characters must occur in pairs for a string to be permutable into a palindrome, with 1 exception: if the string is of odd length.

Hint #2

You will need to map characters to frequencies: count the occurances of each character of the string, as you did with words in Lab 7.

Hint #3

Test that at most one character appears an odd number of times.