SER 300 Programming and problem solving
Practice 5

Objective

- Practice using java Collections classes.

Instruction:

- Start by creating a test class for each problem
- Create a class representing data entities in the problem and write a method or multiple methods to solve the problems.
- Write one more method to test the functions you develop.
- Follow the java programming guidelines (attached)

Problems:

1- ~~Given two strings s and t, determine if they are isomorphic. Two strings are isomorphic if the characters in s can be replaced to get t.~~
~~For example, "egg" and "add" are isomorphic, "foo" and "bar" are not.~~
~~**Must use a data structure.**~~

2- ~~Write a program that is given a list of strings, sorts the list by the number of vowels in each string.~~
~~Example input:~~

~~List<String> words = new List<String>();~~
~~words.add("less"); words.add("code"); words.add("evenless") ; words.add("bugs");~~

~~Example output:~~
~~less~~
~~bugs~~
~~code~~
~~evenless~~

3- Implement the solution for the following problem from Practice 2. **Use the most optimal data structure. (**5pts)
Given a string, find the length of the longest substring without repeating characters

4- Write a program that is given two lists: Employees names and their salaries, return a list of names of employees whose salary occurs the fewest time in the given input. If there is a tie (two or more rarest salaries occur the same number of times), return the lowest salary among them. **Must use a data structure.**

Example input:
{"Jim","Tom","Alan","Jack","Bob","Steve","John","Justin","Mary","Larry","Sue"}
{300000,900000,900000,400000,800000,300000,800000,400000,400000,450000,800000}

Output:
{"Steve"}

Bonus Problem

5- Write a method `splitStack` that takes a stack of integers as a parameter and splits it into negatives and non-negatives. The numbers in the stack should be rearranged so that all the negatives appear on the bottom of the stack and all the non-negatives appear on the top. In other words, if after this method is called you were to pop numbers off the stack, you would first get all the nonnegative numbers and then get all the negative numbers. It does not matter what order the numbers appear in as long as all the negatives appear lower in the stack than all the non-negatives. You may use a single queue as auxiliary storage.

Grading criteria:

Criteria	**#1**	**#2**	**#3**	**#4**	**Bonus**	
Solution is correct	10	10	10	10		
At least four tests are defined and run successfully	8	9	9	9		
Code follows guidelines provided	5	5	5	5		
Problem #1 submission by end of class	5					
Total	28	24	24	24	+10	**100**