Map Problems

Java Maps are pretty much the same as Python Dictionaries.

`numberToLargestDivisor` constructs a map associating a number with its largest divisor. For example, `numberToLargestDivisior(10)` yields {2=1, 3=1, 4=2, 5=1, 6=3, 7=1, 8=4, 9=3, 10=5}

```
public static HashMap<Integer,Integer> numberToLargestDivisor(int maxNum) {
    HashMap<Integer, Integer> map = new HashMap<Integer, Integer>();
    for(int k = 2; k <= maxNum; k++) {
        int numberToFindDivisorOf = k;
        for(int j = numberToFindDivisorOf / 2; j > 0; j--) {
            if(numberToFindDivisorOf % j == 0) {
                map.put(numberToFindDivisorOf, j);
                break;
            } // end if
        } // end for
    } // end for
    return map;
} // numberToLargestDivisor
```

`doubleMap` takes a map and returns a new map of where both the keys and values are double. So far example, {A=a, BB=bb} yields {AA=aa, BBBB=bbbb}

```
public static HashMap<String,String> doubleMap(HashMap<String,String> originalMap) {
    HashMap<String, String> retVal = new HashMap<String, String>();
    for(String key : originalMap.keySet()) {
        retVal.put(key + key, originalMap.get(key) + originalMap.get(key));
    } // end for
    return retVal;
} // doubleMap
```

`findMostFrequentStartingLetter` takes an array of strings and returns the most frequent beginning letter. So for example, the strings {“ant”,”bug”,”aunt”} yield 'a'

```
public static char findMostFrequentStartingLetter(String[] strings) {
    HashMap<Character, Integer> letterToFreq = new HashMap<Character, Integer>();
    for(String current : strings) {
        char startingChar = current.charAt(0);
        if(!letterToFreq.containsKey(startingChar)) {
            letterToFreq.put(startingChar, 0);
        } // end if
        int currentFrequency = letterToFreq.get(startingChar) + 1;
        letterToFreq.put(startingChar, currentFrequency);
    } // end for

    //ok now find the most frequently appearing 1st character
    int highestFrequency = 0;
    char charWithHighest = '\0';
    for(char current : letterToFreq.keySet()) {
        if(letterToFreq.get(current) > highestFrequency) {
            highestFreq = letterToFreq.get(current);
            charWithHighest = current;
        } // end if
    } // end for
    return charWithHighest;
} // findMostFrequentStartingLetter
```

Map Reference

```
// A Java Map is like a dictionary in Python
// It associates a key with a particular value - but because this is
// java the key and the value both must be typed
//
// Format: HashMap<KeyType,ValueType> foo = new HashMap<KeyType,ValueType>();

HashMap<String, Integer> namesToWeight = new HashMap<String, Integer>();

//to add elements to the map, use put
namesToWeight.put("Buffalo", 160); // key = “Buffalo”, value = 160
namesToWeight.put("Gretchen", 130);// key = “Gretchen”, value = 130

//note that putting twice with the same key overwrites the original value in the map
namesToWeight.put("Buffalo", 165);

//to get elements out of the map, use get
System.out.println("Buffalo's weight is " + namesToWeight.get("Buffalo"));

//if you need to check if a particular key is in the map, use containsKey method
if(namesToWeight.containsKey("Steve")) {
      System.out.println("Steve is in the map!");
} // end if

//if you need iterate over all the keys in the map, use the keyset method
Set<String> keys = namesToWeight.keySet();
// Use the enhanced for loop to iterate over the entire set
for(String key : keys) {
      int value = namesToWeight.get(key);
      //do something for every key and value
} // end for
```