AP Computer Science Name_____
Sets and Maps

As in math, a set holds objects, but no duplicates, and should be able to add an object, remove an object, and test if the set contains a certain object. Look at the API cheat sheet, or the book Pages 484-491.

As in math, a map holds a pair of Objects (called "keys" and "values"). A key maps to a value. Keys must be unique. Maps should be able to add pairs, remove keys, and test for keys. Look at the API cheat sheet.

How do you iterate over a map? Write some code: _____
_____
_____
_____

HashSet and TreeSet are two concrete classes that implement the Set interface.

HashMap and TreeMap are two concrete classes that implement the Map interface.

The Hash versions store objects in a (pseudo)random order. The Tree versions store objects in order. If the order of the objects is not important, use the Hash classes. If the order of the objects is important, use the Tree classes.

The Hash versions provide O(1) run times for the get and put operations. Later we'll discuss what hashing is.

The Tree versions provide O(log n) run times for the get and put operations. That's because the Tree classes are implemented as self-balancing binary search trees. Nice.

Examples

1.

```
1   Set<String> s = new HashSet<String>();
2   s.add("Mary");
3   s.add("Joan");
4   s.add("Mary");                          //duplicate!
5   s.add("Dennis");
6   System.out.println("Size:  " + s.size());
7   Iterator it = s.iterator();
8   while(it.hasNext())
9      System.out.print(it.next() + " ");
10  System.out.println();
11  Set<String> t = new TreeSet<String>(s); //from HashSet to TreeSet
12  it = t.iterator();
13  while(it.hasNext())
14          System.out.print(it.next() + " ");
15  System.out.println(s);                  //print a set, formatted
```

Output:

line 6: _____

line 9: _____

line 14: _____

line 15: [Joan, Mary, Dennis]

2.

```
1   Map<String, String> h = new HashMap<String, String>();
2   h.put("Othello", "green");
3   h.put("MacBeth", "red");
4   h.put("Hamlet", "blue");
5   if(!h.containsKey("Lear"))
6     h.put("Lear", "black");
7   System.out.println(h.containsKey("Othello"));
8   System.out.println(h.keySet());
9   Map<String, String>  t = new TreeMap<String, String> (h);
                  //from HashMap to TreeMap
10  System.out.println(t.keySet());   //print the __________
11  Iterator it = t.keySet().iterator();
13  while(it.hasNext())
14    System.out.print(t.get(it.next())); //print the __________
```

Output:

line 7: _____

line 8: _____

line 10: _____

line 14: _____

Assignment: SetsOfLetters

The given text file "declarationLast.txt" consists of several lines of alphanumeric data. Write a program to read the lines. Print each line and determine for each line the sets of:

- Lower case letters
- Capital letters
- Non-alphabetic characters.

At the end, print the sets of lower case letters, capital letters, and other characters that appeared in each and every line of the text. In other words, print the intersections of the relevant sets.

Sample Run

```
We, therefore, the Representatives of the United States of
Lower case:  a e f h i n o p r s t u v
Upper case:  R S W
Other: ,

America, in General Congress, Assembled, appealing to the
Lower case:  a b d e h i l m n o p r s t
Upper case:  A C G
Other: ,

Supreme Judge of the world for the rectitude of our intentions,
Lower case:  c d e f g i l m n o p r s t u
Upper case:  J S
Other: ,
  <...snip...>
Characters in every line:  Lower case:  e i n o p r s t
                           Other: ,
```