Colonel Sandurz: Prepare ship for light speed!
Dark Helmet: No, no, no, light speed is too slow!
Colonel Sandurz: Light speed too slow?
Dark Helmet: Yes, we're gonna have to go right to ... ludicrous speed!

This assignment is optional—points towards this assignment will compensate for up to 10 points of lost homework credit. This is also optionally a group assignment: you may do this assignment either individually or in your assigned partnership (from PS6, not the trio/quartet from the presentations).

1. Write a HashTable class that stores integers. Of course, you should write your own hash-table class for this assignment, and not use Java’s built-in Set or Map classes. You will have to make several design decisions for this assignment, including:
   - what hash function to use.
   - the size of the table.
   - how to resolve collisions.

2. In a tester class, do the following:
   - Create an array A containing n random integers. Create a second array B also containing n integers, where B[i] is either A[i] (happens 50% of the time) or a new random integer (happens 50% of the time).
   - The idea is to search A[] for the elements found in B[]. Do this two ways: by sorting A[] and using binary search, and by using your HashTable class. For strategy #1, use Radix Sort.

3. In ps8.txt, describe the decisions that you made in the design of your hash table, plus anything unexpected that happened along the way. Furthermore, also include a comparison of the how well these two approaches worked in terms of empirically observed running time. You may find the following code fragment helpful in your evaluation:

```java
double time = System.currentTimeMillis();
// do something.
System.out.println("That took " + (System.currentTimeMillis() - time) + " ms");
```

You should try testing the above strategies on more than one pair of arrays A[] and B[].