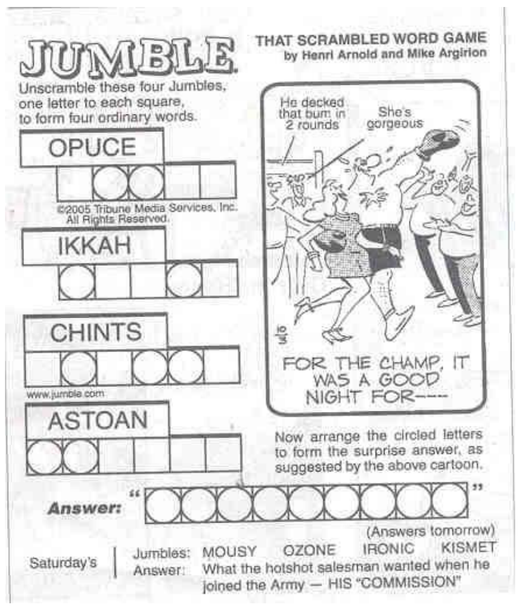
Program #5. HashMap version of Jumbles



Program5.java Starter File

Your program should take the dictionary and the jumbles file on the command line like below. I will only test the big files since your code should be lighting fast!.

```
C:\>java Program5  dictionary.txt   jumbles.txt
```

- dictionary.txt
- jumbles.txt

Your output must be sorted vertially and horizontally to match my output EXACTLY.

YOU MUST USE A HASHMAP. ANY SOLUTION REQUIRING MORE THAN 3 SECONDS IS A FAIL

You must have your entire main wrapped in a try/catch block so that any possible Exception is caught. That exception must print "EXCEPTION DETECTED" + e then exit. If you hand in a file that does not trap an Exception - and you crash my script with that Exception, you will be deducted 10% that cannot be recoved in subsequent submissions!

```
addej: jaded
ahicryrhe: hierarchy
alvan: naval
annaab: banana
baltoc: cobalt
braney: nearby
celer: creel
couph: pouch
cukklen: knuckle
dica: acid cadi caid
dobeny: beyond
dobol: blood
dufil: fluid
dupled: puddle
eslyep: sleepy
ettniner: renitent
ettorp: potter
genjal: jangle
gluhc: gulch
hartox: thorax
hoybis: boyish
hucnah: haunch
iddec: diced
irrpo: prior
kutbec: bucket
lappor: poplar
lasia: alias
laurib: burial
lubly: bully
meefal: female
milit: limit
mopsie: impose
mycall: calmly
nekel: kneel
nokte: token
noper: prone
nwae: anew wane wean
nyegtr: gentry
perrim: primer
preko: poker
pudmy: dumpy
pypin: nippy
rebisc: scribe
rodug: gourd
rpeoims: imposer promise semipro
shewo: howes whose
wardty: tawdry
warllc:
yaldde: deadly
```

Note that the jumbled word "warllc" has no matching word in the dictionary, thus you write nothing after it and go to the next word. You should also notice that the jumbled words are being listed in alphabetical order and so are the dictionary words after it.

Implementation Details

- Read the strings from the dictionary file process the dictionary according to one of the strategies discussed in class or use your own algorithm.
- You MUST use HashMap to store or process your Strings.
- You **are** allowed to use an ArrayList to sort the strings.

Here is a sample program that produces a sorted version of a single String

```
import java.io.*;
import java.util.*;

public class Sort
{
        public static void main( String args[] )
        {
                String unsorted = "zebra";
                char cArr[] = unsorted.toCharArray();
                Arrays.sort( cArr );
                String sorted = new String(  cArr );

                System.out.println( "unsorted: " + unsorted + ",    sorted: " + sorted );
        }
}
```