Program 1

Write a complete Java program in a file called `MatrixFill.java` that reads all the tokens from a file named `data.txt` that is to be found in the same directory as the running program. The program should read all tokens that can be read as an integer and double and ignore the rest. After reading all of the numbers, the program should print them all back to the screen, in two rows from the smallest to the largest. The first row should display the integers and the second row, the doubles. For example if `data.txt` contains `10 5five 10 1.5 2 2.0 20`

Then your program should print to the screen. **Notice that there are no duplicate numbers**

```
2 10 20
1.5 2.0
```

Your program should not prompt the user for anything and should print nothing except the sorted rows of numbers (if there are no numbers your program should print nothing). If the file does not exist, then your program should instead print exactly "File not found" and exit. If the file is empty, your program should print “File empty”

Your program should start by allocating a two dimensional array of doubles of size (2x2) and fill the first row with integers and the second row with doubles. If either row gets filled, it should replace the array with one of size (2 x4) and copy all the numbers already read. If either row gets filled again, it should replace the array with one of size (2 x 8) and copy all the numbers already read.Etc. You should write a method that performs this task. All helper methods should be placed in a file called `FillHelper.java` Also, be sure you read the textbook about the `Arrays` class and the java documentation about the static method `Arrays.sort(a, fromIndex, toIndex)`. You should not write your own sorting algorithm. You should

```
/**
*      Program that reads tokens from file data.txt and prints the    * tokens found in
it to the screen in increasing order.
 *
*      @author Your Name
*      @version date of submission
 */
public class MatrixFill {

    /**
*      Allocates and returns a double array twice the column size of the one
*      supplied as a parameter. The first half of the new array will
*      be a copy of the supplied array and the second half of the new      * array
will be zeros.
     *
*      @param arr the array to be copied
*      @param size the current column size, the method doubles this
*      @return array twice the column size of <tt>arr</tt> with its initial
*      elements copied from <tt>arr</tt>
```

use the Arrays methods for copying the arrays, deepEquals if necessary, and for..each loop for iterating through the array for printing. Note that while integers will be stored as doubles in the first row of the array, they will need to be displayed back as integers. A little starter code:

```
     * @throws NullPointerException if <tt>arr</tt> is null.
     */     public static int[] doubleArrayAndCopy(double[][] arr, int
size) {
    }      public static void main(String[]
args) {         double[][] data = new
double[2][2];         ...         data =
FillHelper.readData(data);
                    // readData calls doubleArrayAndcopy(data,size)
...
    }


}
```

What to submit: Submit the file `MatrixFill.java` **and** `FillHelper.java` via Canvas dropbox.

Testing: You should test your program in a variety of situations -- missing file, empty file, file with no integers, file with no doubles, file with numbers and non-numbers, long file, short file, etc. That's what I will do and if your program fails any of my tests, you will get no credit for your initial submission. **Grading:** You will get full credit for this program if it is submitted on time, is formatted properly and fulfills all functional requirements. It will receive zero credit otherwise. You will get a chance to resubmit your program if it doesn't work the first time, but you will lose 25% per round of grading (there will be at least one week between grading rounds).

Collaboration: The preparatory work for this module has allowed you to work with others, but this program is for a grade so you should write it yourself.