* 1. Use the first argument of commandline input as the input file name.
		1. For example, args[0] will be the name of your input file. (The first line of your main method is: **public static void main(String [ ] args)**, so array element args[0] contains a string which is the name of your input file.)
		2. The name of your input file should be typed into the *Run Arguments* text box at the top of jGRASP. (You should test your program with all three of the input files listed at the bottom of this webpage. Therefore, the name of your input file is: electricity.txt, groceries.csv, or 1000.txt.)
		3. In the main() method, make sure that your error-checking includes checking for commandline arguments. If the user does not enter the commandline arguments, your program should show an error message. Your program should not crash.
	2. Store each integer in the input file in an array of integers.
		1. Write a method to accomplish this.
		2. The parameter of the method is the input file name.
		3. The method should return an array of integers with all the integers in the input file.
		4. You need to make a connection to the input file using the Scanner class. See the [Java API of the Scanner class](http://docs.oracle.com/javase/7/docs/api/java/util/Scanner.html) for more details on the Scanner constructor, hasNextLine(), next(), and nextInt() methods.
		5. Loop through the file using the method hasNextLine() of the Scanner class.
		6. Use the method nextInt() of the Scanner class to read integers from the file and store them into an array. For example: x = inputFromFile.nextInt(); (where inputFromFile is a Scanner object). (Alternative code: You could also use Scanner method next() to get one word from the file, and Integer.parseInt() to change strings into integers. And NumberFormatException to catch exceptions.)
		7. Use a *try-catch block* to determine which words are integers and which are not by using the InputMismatchException class, where you need to remove the non-integer string from the input stream:
		String word = inputFromFile.next();
		8. You also need to use NoSuchElementException class for blank lines in the file.
	3. Print out the number of integers in the file, and all the integers in the array.
		1. Write a method to accomplish this.
		2. The parameters of the method are the array of integers and the file name.
		3. Print the name of the file.
		4. Print the number of integers in the file.
		5. Print all of the integers that are stored in the array.
		6. Your output should be similar to the output below.
		7. You will have extra null values in your array, so you will need to loop from zero to the number of integers stored in the array, or write an if-statement, so that null values are not displayed.
	4. Your main method should be *very short*, as it only has two method calls, and error checking for commandline input.
	5. Write your original comments every 3-5 lines of code.
	6. Make sure your code follows the [ICS 211 Java Coding Standard](http://www2.hawaii.edu/~walbritt/ics211/materials/standard.htm), in particular the Java documentation (Javadoc) comments that go above each method.
	7. *WARNING: Do NOT copy my code or my comments. Use my code as a guide to write your own code.*