Test Strategy: Code Snippet with Arrays

Write a code snippet that executes the specification below.

A. Given: a double array named numbers that has been declared and loaded elsewhere.
B. Calculate the average only of elements > the number stored in double variable cutoff, also declared and loaded elsewhere.
C. Print out the array, one element per line with the index of the element followed by a dash (hyphen) followed by the element.
D. But, if the element is > variable cutoff, add to the end of the line the phrase, " *** > cutoff".
E. After printing the array, print the average your code produced with a message identifying the average is the sum of elements above the cutoff.
F. Print a blank line and then your name.

My Approach:

To make sure I only write code that I actually need, I'll ask a few questions to make sure I've broken the problem down as far as I can. Once I know the answers, it will be much easier to build a solution.

One: What is given to me?	double[] **numbers** = [some array of numbers]; double **cutoff** = [some number];
Two: What are my goals? What should my code do?	• Calculate the average of only elements greater than the number stored in **cutoff** • Print out each element of the array on its own line, in the format "Index - Value" • If an element is greater than **cutoff**, print the string " *** > cutoff" next to it • Print the average with a message explaining how I calculated it • Print a blank line and my name
Three: What operations will I need to perform?	• Look at each item in the array **numbers**, one at a time • Test whether an element is larger than a fixed variable • Calculate an average (add up some numbers, divide the sum by the quantity of numbers I added) • Print some things
Four: What new variables will I need?	• Something to hold my sum • Something to hold the quantity of numbers I add up • Something to hold the average once it's calculated

Ok, let's code!

First, I'll declare my new variables. Since my elements are all doubles, I'll need the sum and average variables to be doubles, too. There are no partial entries in the array, so the quantity can be an integer. I'll start them all off at zero.

```
double sum = 0.0;
double average = 0.0;
int numAdded = 0;
```

Next, since I'm going to need to look at every element in the array, I'll set up a loop to do that. The length of the array has already been decided for me, which means I can know in advance how many times to repeat, so I'll use a **for** loop. Every array starts at index 0, so that's where I'll start my counter. I'll increase the counter by one every time, so I can look at the next element, and keep going as long as the counter points to an index lower than the total length of the array.

```
for (int i = 0; i < numbers.length; i++) {
```

For the element I encounter at array index **i**, I'll need to do two things for sure: Print the number and its index (same as the current value of my loop counter **i**), and check to see if it's greater than the value of **cutoff**. If it is, I'll have three more tasks. Since I might be printing a second thing on the same line as the element, I'll start with just a basic **print()** statement.

```
    System.out.print(i + " - " + numbers[i]);
    if (numbers[i] > cutoff) {
```

If I do find that an element is greater than **cutoff**, I need to add its value to my sum, then increase the quantity-of-numbers-added counter by 1. Then, I need to print a special string. Every element needs its own line, but the special string only applies to some elements, so I'll use another basic **print()**.

```
       sum += numbers[i];
       numAdded++;
       System.out.print(" *** > cutoff");
    } // End if
```

That takes care of the things I have to do for every element. I'm almost ready to close my loop, but first, since I haven't told my output to end each element's print(s) with a new line, I'll throw in an empty **println()**.

```
    System.out.println();
} // End for
```

All that's left now is to calculate my average and print. If the number of things I counted is greater than zero, I'll divide the sum by that number of things, but if not, there is no average to calculate, and **average** can stay at its initial value of 0.0. Then I'll print it out with an explanation, print a blank line, then my name.

```
if (numAdded > 0) {
   average = sum / numAdded;
} // End if
System.out.println("Average of values > cutoff: " + average);
System.out.println();
System.out.println("Dr. Tor Coolguy");
```

Hey, I'm allowed to dream. Don't judge me.