Program GradeStats.java

Write a program that will:

- Ask the user for the scores to be entered until the user types in quit.
    - Use Integer.parseInt() to change each user input (String) into int.
- Read in the scores and store them to a 1-D array. For now, make your array big enough to handle what we will need, say 20 cells. Note, realistically, for a problem like this, we should have more cells, but we won't be typing that many in so we will use 20. For a better way that works for our level, see the note at the bottom of the page.
- After the array is completely built (input has been completely assigned),
    - Print out the data, showing the student number and corresponding score.
    - Print which students had a score below 75%.

    - Then find the following statistics and print them:
        - Number of scores,
        - Maximum (use a Math method),
        - Average score,
        - Median score, and
        - Minimum (use a Math method).

- How to compute the median score.
    - The median score is the number where half of the scores are above it and half are below it. First, you need to put the scores in order from smallest to greatest. If the number of scores is odd, then it is the middle number. If the number of scores is even, it is the average of the two middle numbers.
    - Do this last. Get everything else working first. This is a bit more involved.

Follow the formatting given in the sample run. If there are 10 scores (or any two digit number), the scores (or data) should begin at the same place for each line. When necessary, round to the nearest tenth.

Follow our standard and use main to call calculateIt(), which will then call the rest, as needed.

Remember to break down your code with methods that are short and do what the method name states. For example, you will have separate methods to calculate the scores below 75%, the max/min, the average, and the median (which needs one method to put the numbers in order and another one to find the median) in addition to other ones such as getting the user input, printing the required information and the standard ones.

- Work an example out on paper before starting (using any reasonable data that is not the example provided). This will be turned along with your pseudocode.
- Remember to follow Pandas Don't Eat Oreos conventions.

Note:

How can you deal an array that is full, but you need to add more to it?

- Start by making an array big enough to get going. When it gets full, make an array that is twice as big, store the original values into the new one, then add as needed. This process repeats each time the array is completely used.

Sample run. User input in bold.

```
< prompt info>
Type in the score.  Type "Quit" to end the program -->  95
Type in the score.  Type "Quit" to end the program -->  65
Type in the score.  Type "Quit" to end the program -->  70
Type in the score.  Type "Quit" to end the program -->  78
Type in the score.  Type "Quit" to end the program -->  85
Type in the score.  Type "Quit" to end the program -->  77
Type in the score.  Type "Quit" to end the program -->  QuiT

Here is the data you entered:
Student 1's score:   95
Student 2's score:   65
Student 3's score:   70
Student 4's score:   78
Student 5's score:   85
Student 6's score:   77

There were 2 students who scored below 75%:  student 2, student 3.

Number of scores:    6
Maximum:             95
Average:             78.3
Median:              77.5
Minimum:             65
```