Specify, design, and implement a class called Statistician. After a statistician is initialized, it can be given a sequence of double numbers. Each number in the sequence is given to the statistician by activating a method called `nextNumber`. For example, we can declare a statistician called `s` and then give it the sequence of numbers `1.1, -2.4, 0.8`, as shown here:

```java
Statistician s = new Statistician();
s.nextNumber(1.1);
s.nextNumber(-2.4);
s.nextNumber(0.8);
```

After a sequence has been given to a statistician, there are various methods to obtain information about the sequence. Include methods that will provide the length of the sequence, the last number of the sequence, the sum of all the numbers in the sequence, the arithmetic mean of the numbers (i.e., the sum of the numbers divided by the length of the sequence), the smallest number in the sequence, and the largest number in the sequence. Notice that the length and sum methods can be called at any time, even if there are no numbers in the sequence. In this
case of an “empty” sequence, both length and sum will be zero. The other methods should return 
`Double.NaN` if they are called for an empty sequence.

Notes: Do not try to store the entire sequence (because you don’t know how long this sequence will be). Instead, just store the necessary information about the sequence: What is the sequence length; what is the sum of the numbers in the sequence; and what are the last, smallest, and largest numbers? Each of these pieces of information can be stored in a private instance variable that is updated whenever `nextNumber` is activated.

3 Write a new static method to allow you to “add” two statisticians from the previous project. If `s1` and `s2` are two statisticians, then the result of adding them should be a new statistician that behaves as if it had all of the numbers of `s1` followed by all of the numbers of `s2`. 