1. The return types of sumAll methods is long.
2. The relation between arr and max is that they must both the maximum number possible with which is being dealt with
3. Similarly, to sumAll, since we are printing out a number that is greater than 32 bytes, we are using long which is for 8 bytes compared to integers 4 bytes( less information)
4. By having init(arr) before printing out the written statements, the value of the arr is initialized to a variable so that the index and its value are recorded.
5. As similarly as sumAll, the return type of fib methods will be long again due to the final output being more than 32 bytes of information. Meaning int would not work here.
6. The Fibonacci output starts at 2 because if position 0 and 1 were added together the value of position 2 would be 1. This a number that we start with because if not, any number 0 plus any number would give the starting number of 1.
7. These returns are long as well because the results that are printed out are more than 32 bytes, meaning that the final output is larger than what a integer can handle.
8. Factorial starts at output 2 because if it started at 1 the multiplied numbers would 1 \* 1. So, by starting at 2, 1\*2 gives us 2.
9. Long []arr is initialized as a long because a integer would not be able to hold on the information that we are handling. Since the array is of length 100 and most methods are being multiplied by itself, that value is too large to be contained in just integer.
10. If MAXSIZE is changed to 10, the code will not run smoothly. To fix this, we need to either delete the statements that call on any value greater than or equal to 10. This is because when these number are calculated, it usually takes into account the number after itself, but if we limit the max number to 10, then the methods will stop once they reach the max number of 10.