Work in a group of **two** to simulating map-reduce programming via multi-threading.

1- In your application, you should have three components: the main, the map, and the reduce.

2- Create two threads for the reduce component in map-reduce. The reduce component should be programmed to receive a set of numbers. Once the map part is over, you should print the largest number transmitted from the map component.

3- The map component should initially use four threads. However, it could expand its thread usage to eight when required. The threads in the map component should be designed to receive a set of words from the main component. Once a thread receives a word, calculate its length and send the result to the reduce component.

4- The main component should hold the main function, in which you should generate a string of random words and send them to the map component.

-0.5 Bad code separation. The code for each component should be in a separate file.
-1 The algorithm for generating random words cannot be found in the code.
-1.5 Not creating a thread pool for the map component. The core size and the max size should be set as 4 and 8, respectively.
-2 The main component does not send the data to the map's threads correctly. You need to consider that the number of threads is not fixed, it can be 4 or 6 or even 8. Thus, Splitting the data 1/4 is not flexible. What about the case in which you need to send the data to 7 threads or 8.
-1 Not using a thread-safe data structure between the map and main components.
-1 The lengths of the words are not transmitted correctly to the reduce threads.
-1 Not creating two threads for the reduce component.
-2 The reduce component does not printing the maximum length ONCE the data has been processed.