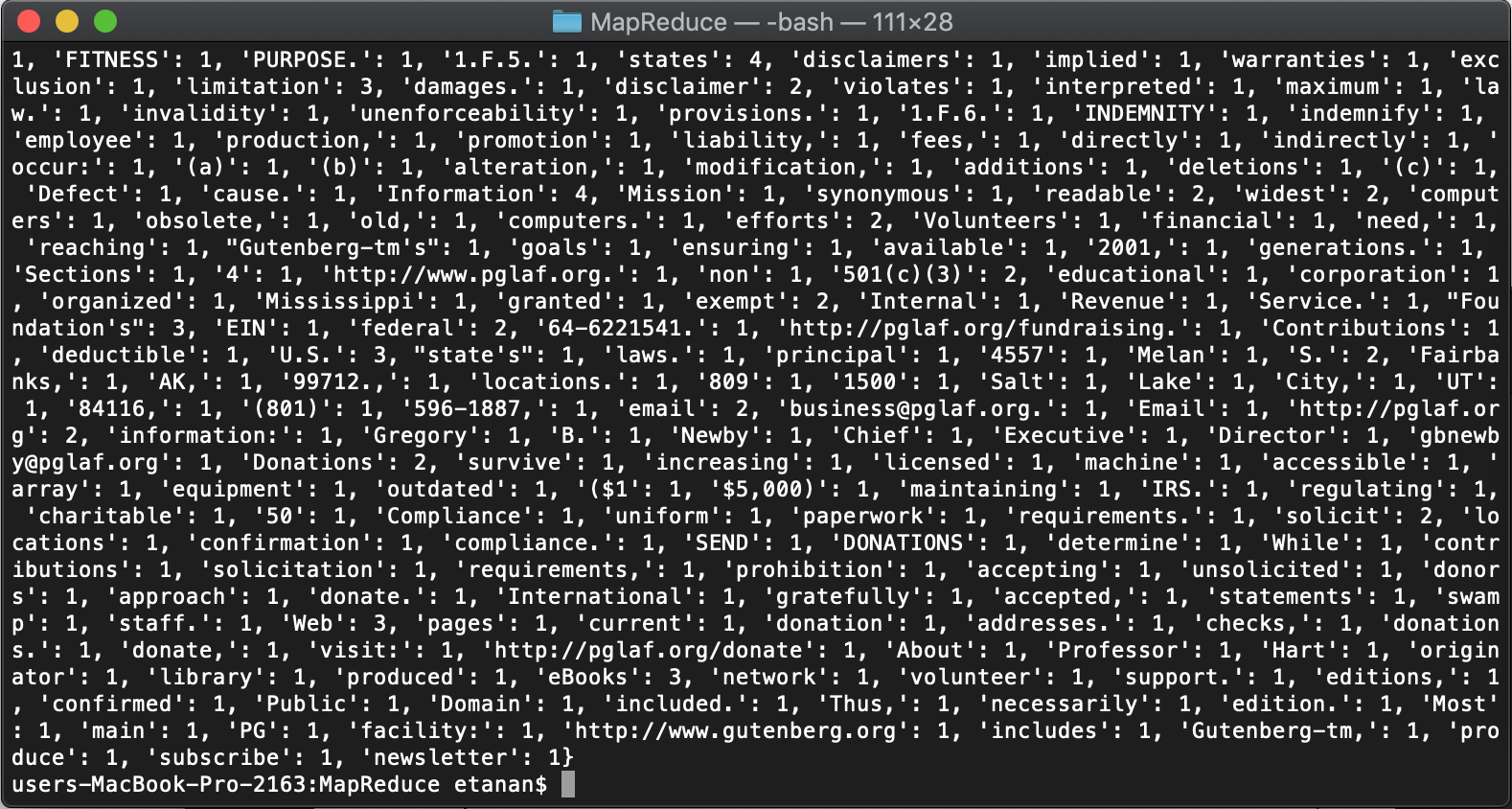
Exercise – MapReduce Example with Python

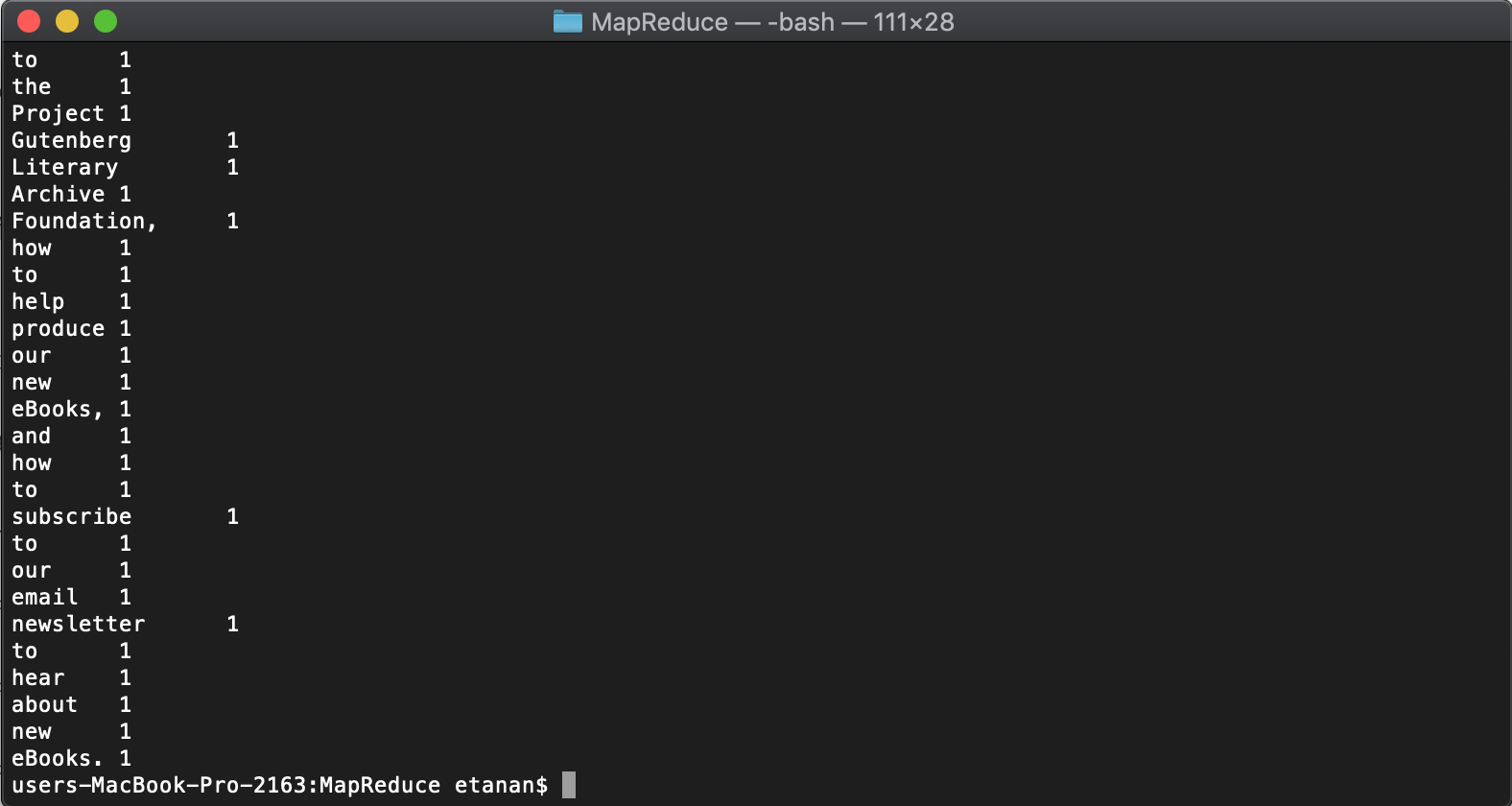
1. users-MacBook-Pro-2163:MapReduce etanan$ python counts.py < doyle.txt

This code line runs counts.py and uses doyle.txt as its standard input. counts.py counts words and prints the results along with the number of times the word had occurred.



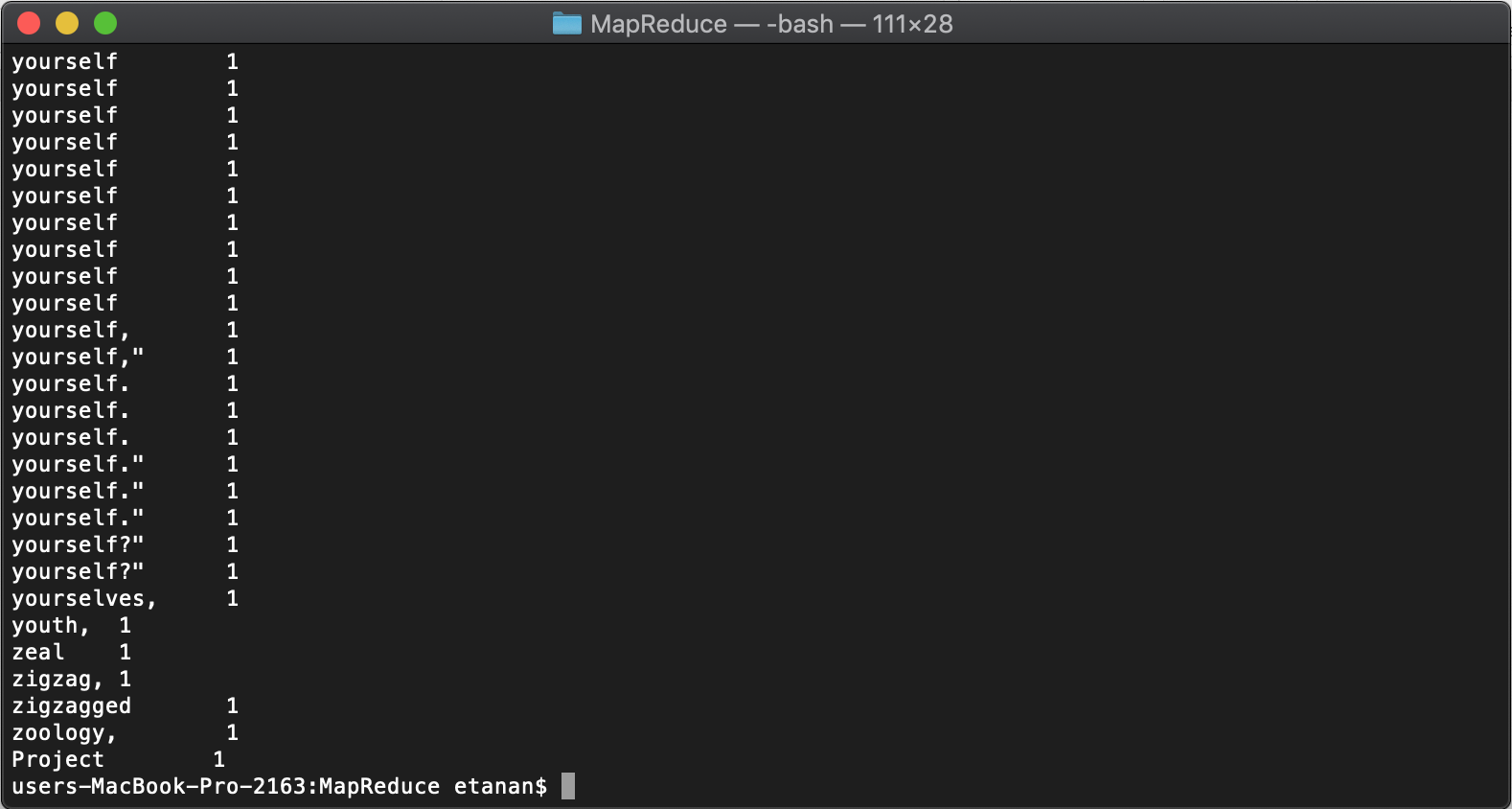
1. users-MacBook-Pro-2163:MapReduce etanan$ python map.py < doyle.txt

This line runs map.py and takes doyle.txt as a standard input value. In this case, the python file splits each word and puts value of “1” after it.



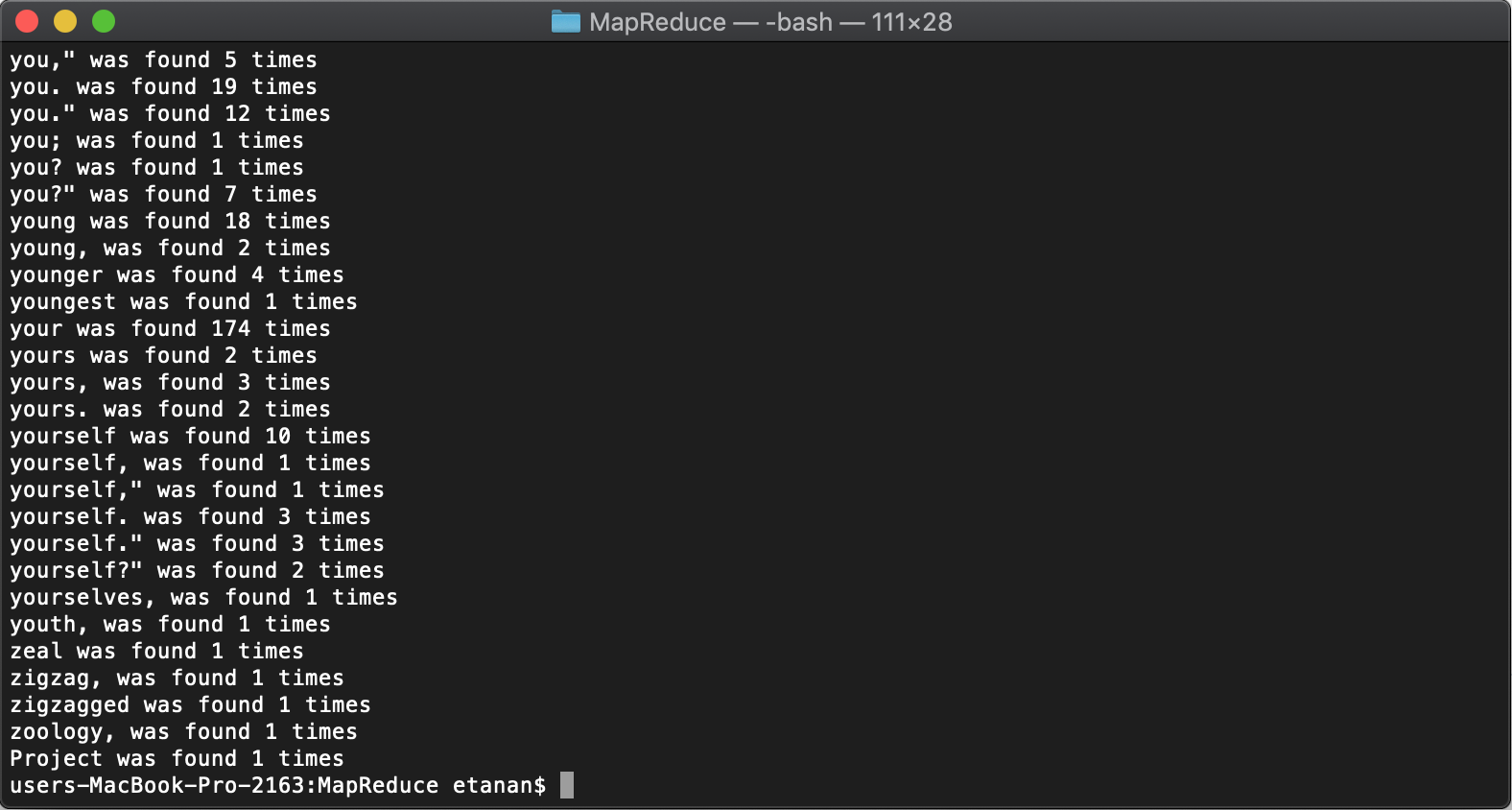
1. users-MacBook-Pro-2163:MapReduce etanan$ python map.py < doyle.txt | sort

This line of code runs map.py with doyle.txt as standard input. Furthermore, takes the outcome as an input to run the ‘sort’ command to alphabetically sort the result and displays it accordingly.



1. users-MacBook-Pro-2163:MapReduce etanan$ python map.py < doyle.txt | sort | python reduce.py

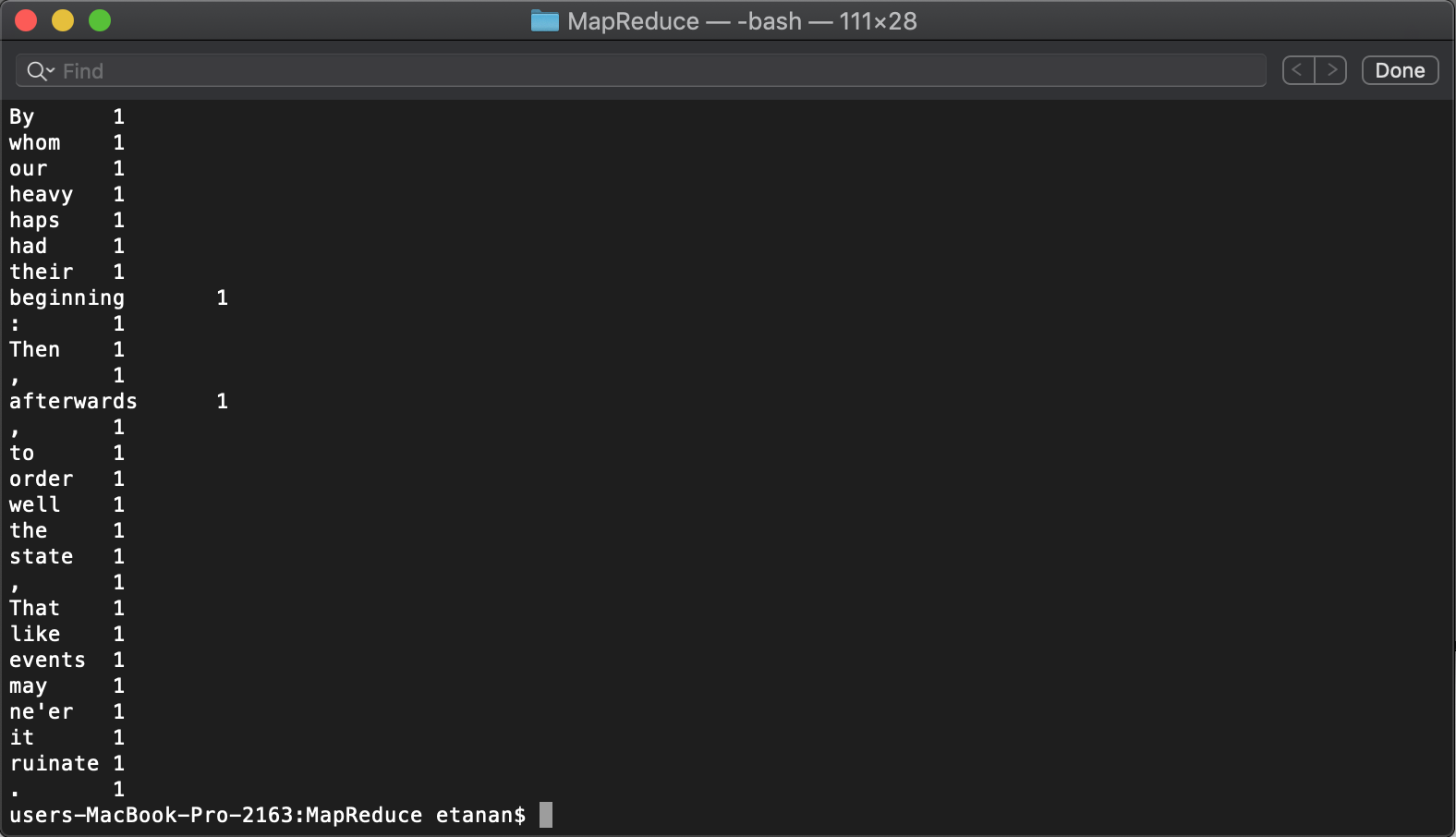
Here, map.py is run doyle.txt as a standard input. The output from this is sorted alphabetically and utilized as a standard input once again when we run reduce.py. What happened here is that, map.py maps each word and reduce.py counts the number of times the word had occurred and displays the results. This is a basic mapreduce process.



The above process is replicated with shakespeare.txt as the standard input file.

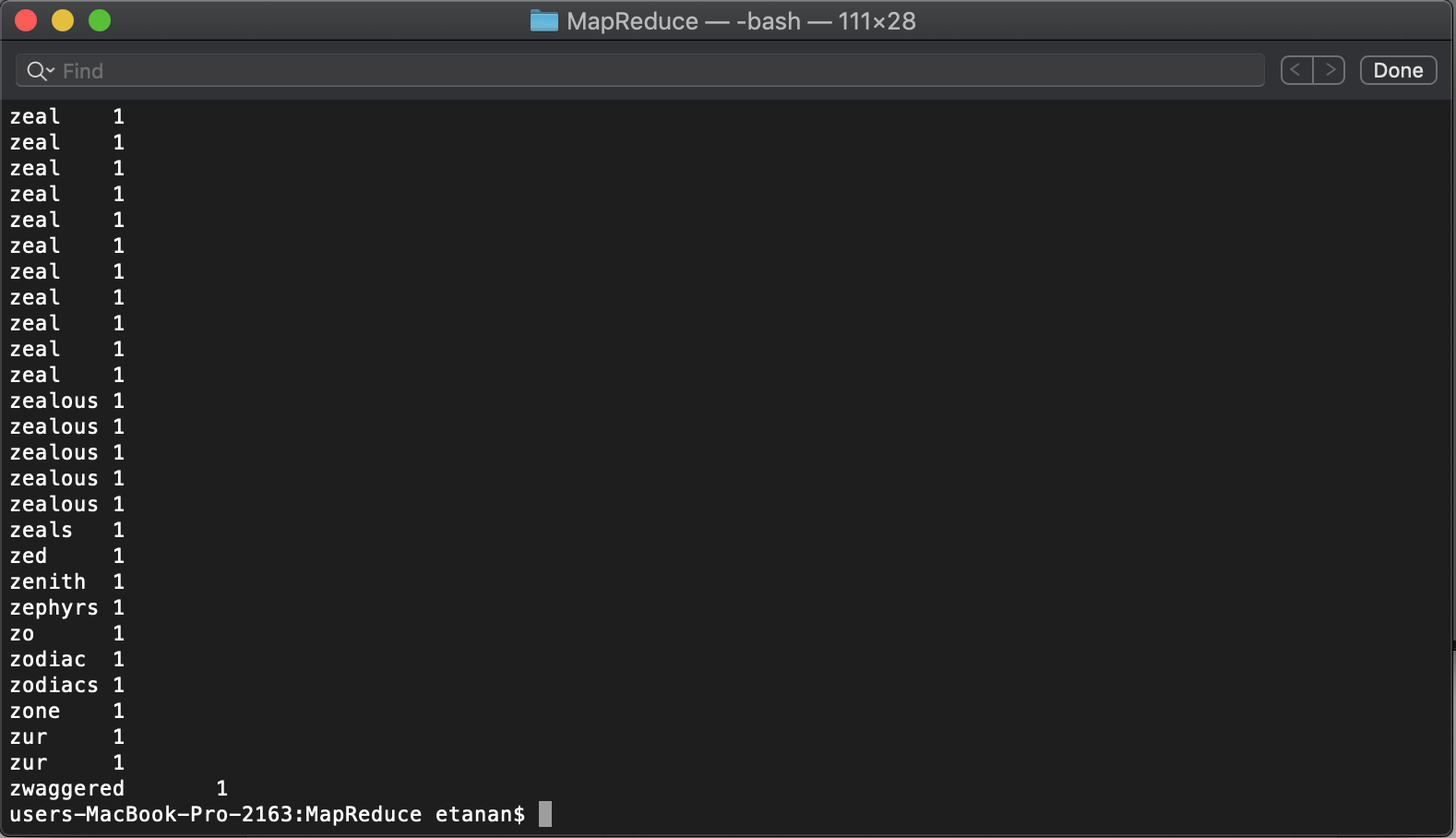
1. users-MacBook-Pro-2163:MapReduce etanan$ python map.py < shakespeare.txt

This line runs map.py and takes shakespeare.txt as a standard input value. In this case, the python file splits each word and puts value of “1” after it.



1. users-MacBook-Pro-2163:MapReduce etanan$ python map.py < shakespeare.txt | sort

This line of code runs map.py with shakespeare.txt as standard input. Furthermore, takes the outcome as an input to run the ‘sort’ command to alphabetically sort the result and displays it accordingly.



1. users-MacBook-Pro-2163:MapReduce etanan$ python map.py < shakespeare.txt | sort | phyton reduce.py

Here, map.py is run shakespeare.txt as a standard input. The output from this is sorted alphabetically and utilized as a standard input once again when we run reduce.py. What happened here is that, map.py maps each word and reduce.py counts the number of times the word had occurred and displays the results. This is a basic mapreduce process.

