

This exercise has two components, it is suggested that you read over the whole exercise first to avoid wasted effort.

Part 1 - Wordcount

You are given the following interface

```
public interface CharacterReader {  
    /**  
     * Return the next character of the stream, or EOFException  
     * when the end of the stream is reached  
     */  
    public char nextCharacter() throws EOFException, InterruptedException;  
    /**  
     * Close the input stream. This must be called after the  
     * last character has been read to dispose of the stream  
     */  
    public void close();  
}
```

Implement a function that takes a CharacterReader and produces a list of words, in order of descending frequency. When two words have the same frequency, they should be alphabetical order.

For example the input “The cat sat on the mat.” would produce:

```
the - 2  
cat - 1  
mat - 1  
on - 1  
sat - 1
```

Part 2 – Slow, parallel wordcount

You are given 10 instances of the above interface, which sleep for a random amount of time before returning a character. Write code that will read these in parallel. It should output the current (combined) counts of the words every 10 seconds and final totals at the end.