**Lab WordGen Resubmission**

Mistakes in Original Implementation:

1. The program didn’t quit after the user ran the program without any arguments. The usage message wasn’t helpful as it simply said ‘java WordGen’. The usage message was in the starter code, and I forgot to modify it.
2. Program execution was slow, likely due to inefficient handling of the ‘band-aid’ situation in which the sequence of k letters hasn’t been seen before in the input text. I thought that trying to choose the next letter based on the last k-1 letters might lead to a solution, but this is unlikely and simply slows the program as add() and choose() need to be run again in all three classes.
3. Main Method is very convoluted. At the start of the course, I didn’t realize the importance of helper methods to facilitate understanding and debugging.
4. If the argument in user input isn’t an integer, an exception is thrown. I didn’t test this because I thought only integer inputs would be used for assessing the quality of the work.

Corrections in Resubmitted Implementation:

1. The rest of the main method has been placed in the else statement of the condition on number of arguments == 0. Thus, the program doesn’t execute any more code once the condition is met.
2. The band-aid situation has been modified such that when choose() returns null, the first k letters of the input text are added to the generated text and the temporary loop variable ‘i’ is adjusted so that the number of letters in the generated text remains similar to the that of the original text. Adjusting ‘i’ also ensures that on the next iteration of the loop, there will be a valid letter returned by choose().
3. The main method has been further split into getK(), getText() and formNewText().
4. In getK() a try-catch statement is used so that if the program is unable to parseInt(args[0]), an appropriate error message is printed instead of an exception thrown.