Noah Buchen
Midterm CS362 Winter 2019

Question 1:

Positive Test Cases

1. Test that paraphrase returns complete and correct lines when in.txt strings are present
   - Create my own output file grepout.txt using a tool like grep
   - Compare grepout.txt with the paraphrase output (using something like memcmp) to verify correct lines have outputted

Negative Test Cases

1. Test that paraphrase returns no lines when in.txt strings are not present
   - feed paraphrase in.txt files with no matches
2. Test case sensitivity for int.txt and out.txt
   - prepare special int.txt and out.txt files with mismatched case comparisons
3. Test that paraphrase returns lines only when in.txt string is a complete match (ex. in.txt = “hot” out.txt = “I ate a hotdog today”)
   - run example above and verify no lines print

Boundry Test Cases

1. Test that paraphrase returns no lines when in.txt has no strings in it
   - pass empty int.txt
2. Test that paraphrase returns no lines when out.txt has no strings in it
   - pass empty out.txt
3. Test that paraphrase can handle really long in.txt strings
   - fill in.txt with a really long random string
   - create file using grep and use something like memcmp to compare the output from paraphrase
4. Test that paraphrase can handle really long out.txt strings
   - fill out.txt with a really long random string
   - create grepout.txt file using grep and use something like memcmp to compare it to the output from paraphrase
5. Test that paraphrase can handle missing parameters (0 or 1)
   - run twice with missing parameters
6. Test that paraphrase can handle an in.txt with a large multitude of strings
   - fill in.txt with a huge group of random strings
   - fill out.txt with a small selection of the strings plus lots of junk
   - create grepout.txt and then compare to paraphase output

Test Coverage

- Test coverage data should be gathered using a tool like gcov
- For a such a simple system, 100% branch coverage would be the goal