Sharon Kudenko

Part 1
- 18/20
  - 10/10
  - 8/10

Part 2
- 9/10 (output of most common words is printing a tuple, not formatted text)

Part 3
- Only works when given a file, so could be 20/20 for effective automated decryption, or 10/20 for half of the required functionality, as it is limited to ½ cases.

Part 4
- 0/10 – no #!# comments found or references to extensions in report.

General implementation
- 13.5/30 – 15.5/30
  - 2.5 – *mostly* sensible names but very inconsistent naming conventions (camelCase, underscores, etc.)
  - 3 – or 4 – using the right data structures
  - 5 – sensible comments
  - 0 – for elegant implementation
  - 3 – or 4 for program structure
  - 0 – for robustness

Report
- 10/10 (or close to it, covers lots of detail, and without the waffle, it is 2 pages)

Total: 60.5/100 – 63.5/100 – I think this could be higher, I think they've achieved a high 60s mark overall, and with such a strong report, possibly hitting 70, but there are some strange things in the code.

General comments:
Lots of use of lambda functions, which seems suspect.
Also_some_underscore naming used, and other times, nocaseatall.

Feedback:
This was a great implementation of this project, having achieved a highly-functional implementation of Parts 1-3. The only thing that was missing for Part 3 was that the automated decryption did not work without giving your program an input file. Ideally, this should work for either user input, or file input, in the same way. You have made excellent use of functions in parts of your project, adding new functions when appropriate, but have also seemed to introduce duplicate functions that extend the functionality of previous ones without using the originals (multiple decrypt functions, one of which prints output, and another returns it). I think this could be better structured with a bit of planning. The general program structure is very good, though, with a nice amount of detail provided in the comments throughout. Most of your variable and function

names are quite sensible, but sometimes inconsistent, and your naming conventions are very inconsistent, using camelCase and underscores in different places. I would have liked to see some error handling when negative rotation values are entered, or the file is missing from the arguments list, or not enough arguments are given, but generally your program works great for most of the arguments expected.
The report was very detailed, covering both your design decisions and problems/solutions that you worked on during the implementation. This was excellent, well done.