Name: Connor Morrison
NSID: tvi340
Student Number: 11374770
Course Number: CMPT 145
Lecture Section: 04
Laboratory Section: L02

1. (2 marks) Comment on your program's correctness (see Chapter 4 of the textbook for the definition). How confident are you that your program (or the functions that you completed) is correct? What new information (in addition to your current level of testing) would raise your confidence? How likely is it that your program might be incorrect in a way you do not currently recognize?

- From the textbook, "correctness" refers to a program delivering a correct and intended result.
- I am confident that my program is correct since for any given direction (N, S, E, W), each is properly mapped to its correct position on the cartesian plane. Since the underlying logic is correct, I am confident the program is correct.
- I believe testing more files with different combinations of directions would raise my confidence.
- I don't think the program is incorrect in a way I don't currently recognize. If it is at all, it is possibly in the overall design of the code.

2. (2 marks) Comment on your program's efficiency (see Chapter 4 of the textbook for the definition). How confident are you that your program is reasonably efficient? What facts or concepts did you use to estimate or quantify your program's efficiency?

- It's difficult to comment on efficiency since we have yet to learn about time complexity, or in-depth discussions on improving memory usage. However, I do think I designed the code to be concise and arrive at the solution in a direct and efficient manner.

3. (2 marks) Comment on your program's adaptability (see Chapter 4 of the textbook for the definition). For example, what if Question 2 asked you to write a program that had additional letters, say 'U' and 'D', which moved Mechagodzilla up and down (vertically)? How hard would it be to take your work in Question 2, and revise it for 3 dimensions?

- Adaptability, as per the textbook, refers to how small changes in program behaviour can be achieved through small changes in the code.
- To add additional letters for already established coordinate movements could simply involve adding "OR" statements to take in more possible moves, although this might not be the most elegant solution.

- For adding an additional dimension, we would likely still have a series of if/elif statements but this time adding an additional third dimension to each ordered pair. This would require more implementation than just adding an additional 'U' or 'D' command.

4. (2 marks) Comment on your program's robustness (see Chapter 4 of the textbook for the definition). Can you identify places where your program might behave badly, even though you've done your best to make it correct? You do not have to fix anything you mention here; it's just good to be aware.

- An area of potential concern could be input validation. If the program receives a input that is in an incorrect format, it could cause an error. I haven't implemented comprehensive checks to handle all such cases fully.

5. (2 marks) Consider how often you were interrupted, distracted, delayed during your work for Question 2. Do you think these factors affected substantially increased the time you needed? If so, what kinds of steps can you take to prevent these factors?

- For this assignment, I tried to reduce the possible distractions. Although I think I improved at not getting distracted by my phone throughout the assignment, I still experienced interruptions from members at home.
- I think affected some of the time during my planning process, and less on the implementation.
- In the future, I can take steps to work on my assignments at the university where I can experience less interruptions.