Creativity Challenge – Code Review Template

Coder Name: Nolan Downey

Reviewers Names: Connor Ruff, Kelly Buchanan

Creativity Challenge Number: 3

Part 1 - Correctness and Defects – 150 points towards Creativity Challenge Grade – Good faith documentation of these approach will receive full credit.

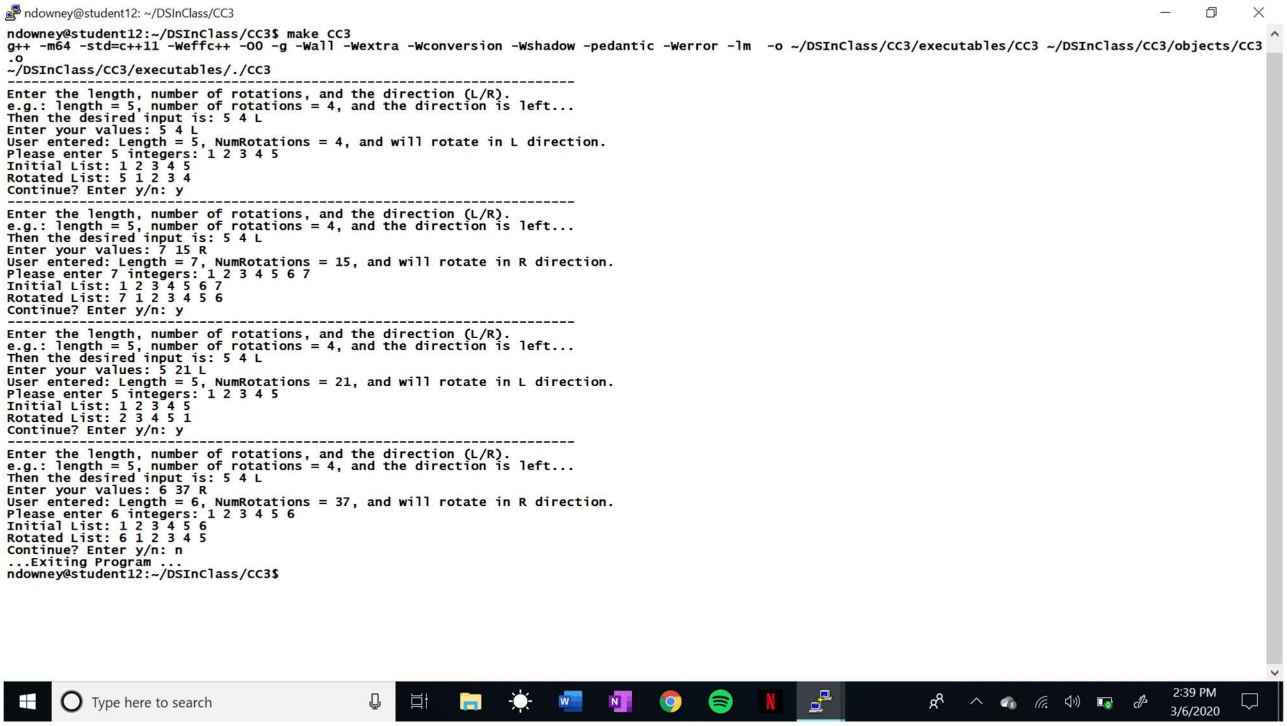
Does the program do what you expect?

- If the code compiles and produces the correct output, to earn the 150 points:
- Describe in detail why your reviewers believe it works
    - Describe their assessment of your proposed solution based on problem statement and the result generated just for correctness (efficiency and memory are in Parts 2 and 3)
    - Show the code run for several test cases

- If the code does not compile or produce the correct output, to earn the 150 points:
    - Describe the currently existing compiler/Makefile linker errors, if any exist.
        - Describe the code segments that are potentially contributing to these errors.
    - Describe the currently incorrect outputs and logical defects, if any exist
        - Describe the code segments that are potentially contributing to these errors.

Reviewer Comments:

- My reviewers, after extensive testing of my code and its outputs, determined that my code produces the correct output for all test cases without any defects. I believe my code produces the correct output.
- Based on the problem statement, for a given set of n integers, a number r, and a direction d(L or R), my code performs r rotations on the n integers in the d direction and outputs the resulting data.



```
ndowney@student12:~/DSInClass/CC3$ make CC3
g++ -m64 -std=c++11 -Weffc++ -O0 -g -Wall -Wextra -Wconversion -Wshadow -pedantic -Werror -lm  -o ~/DSInClass/CC3/executables/CC3 ~/DSInClass/CC3/objects/CC3
.o
~/DSInClass/CC3/executables/./CC3
----------------------------------------------------------------------
Enter the length, number of rotations, and the direction (L/R).
e.g.: length = 5, number of rotations = 4, and the direction is left...
Then the desired input is: 5 4 L
Enter your values: 5 4 L
User entered: Length = 5, NumRotations = 4, and will rotate in L direction.
Please enter 5 integers: 1 2 3 4 5
Initial List: 1 2 3 4 5
Rotated List: 5 1 2 3 4
Continue? Enter y/n: y
----------------------------------------------------------------------
Enter the length, number of rotations, and the direction (L/R).
e.g.: length = 5, number of rotations = 4, and the direction is left...
Then the desired input is: 5 4 L
Enter your values: 7 15 R
User entered: Length = 7, NumRotations = 15, and will rotate in R direction.
Please enter 7 integers: 1 2 3 4 5 6 7
Initial List: 1 2 3 4 5 6 7
Rotated List: 7 1 2 3 4 5 6
Continue? Enter y/n: y
----------------------------------------------------------------------
Enter the length, number of rotations, and the direction (L/R).
e.g.: length = 5, number of rotations = 4, and the direction is left...
Then the desired input is: 5 4 L
Enter your values: 5 21 L
User entered: Length = 5, NumRotations = 21, and will rotate in L direction.
Please enter 5 integers: 1 2 3 4 5
Initial List: 1 2 3 4 5
Rotated List: 2 3 4 5 1
Continue? Enter y/n: y
----------------------------------------------------------------------
Enter the length, number of rotations, and the direction (L/R).
e.g.: length = 5, number of rotations = 4, and the direction is left...
Then the desired input is: 5 4 L
Enter your values: 6 37 R
User entered: Length = 6, NumRotations = 37, and will rotate in R direction.
Please enter 6 integers: 1 2 3 4 5 6
Initial List: 1 2 3 4 5 6
Rotated List: 6 1 2 3 4 5
Continue? Enter y/n: n
...Exiting Program ...
ndowney@student12:~/DSInClass/CC3$
```

Part 2 – Performance – 125 points towards Creativity Challenge Grade – Good faith documentation of these approach will receive full credit.

The team's assessment of the run-time efficiency of your proposed solution.

- Does this program use the best Data Structure for runtime given the problem constraints?
    - If your teams believe so, state why.
    - If your team believes not, state why not.

- Does this program run in the most efficient manner possible given the available resources? (i.e., not using unnecessary loops and complex approaches when a Data Structure could simply the approach)
    - If your teams believe so, state why.
    - If your team believes not, state why not.

Reviewer Comments:

- As a team, we believe that my code has the best runtime of the three of us. I decided to use a singly liked list, while the other two members of my group used a vector for the rotating. I used singly linked list to simply change the pointers of the list, allowing the memory elements to stay in place. My reviewers believe this increases runtime, as there are no unnecessary temporary elements that increase the runtime. Given the problem constraints, my reviewers believe my program does not use any unnecessary loops or complex approaches. The singly-liked list simply changes pointers and is not very complicated. However, despite this, there are still ways we believe that could simplify the approach even further than it is, but we have not discovered those yet.

Part 3 – Memory – 125 points towards Creativity Challenge Grade – Good faith documentation of these approach will receive full credit.

The team's assessment of the space efficiency of your proposed solution.

- Does this program use the best Data Structure for memory efficiency given the problem constraints?
  - If your teams believe so, state why.
  - If your team believes not, state why not.

- Does the program make effective use of the heap (i.e., not unnecessarily storing redundant copies of information to produce the solution when a better approach exists), pass/call by reference, function, and method calls?
  - If your teams believes so, state why.
  - If your team believes not, state why not.

Reviewer Comments:

- My team believes that my code efficiently uses memory better than most approaches. I do not create any temporary elements, but I do have to create a node structure for the method of the class SLList.h. Although my approach may have not been the *most* efficient for memory, but we believe that it is more efficient than most other approaches. It is more efficient than a vector because a vector requires double the memory than a list. My team also commented that my code uses effective use of the heap by not unnecessarily storing redundant copies of information to produce the solution. My team commented that I wrote a method for the class SLList.h, which helps in terms of effective use of the heap.

Part 4 – Finding Better Solutions – Optional

Let's say that you have reviewed your group mates code and determine there is a better approach than the one you are taking. A concern you might be having right now would is "if I change my answer, is that Academic Dishonesty?" Here is how you can modify your solution post Code Review and maintain academic integrity:

Describe the changes - State the change in "Correctness and Defects", "Performance", and/or "Memory", as appropriate, that contributed to your solution, and how you wrote your own version of the changed code to meet these changes.

Cite: Name your team member or classmate who gave you the idea, to give appropriate attribution.

Coder Comments:

Kelly Buchanan told me that I could have potentially used a circular linked list instead of a singly linked list. I attempted to have a tail pointer for the last node of the list to point to and earlier node of the list, but I struggled and could not figure out how to run the code effectively with this approach.