

I. Requirements: Restate the problem specification and any detailed requirements in your own words.

The goal of this assignment was to create a program that would take parameters from the command line and then perform an operation using those parameters. We were tasked with reversing a portion of a string based on the parameters passed on the command line.

II. Design: How did you attack the problem? What choices did you make in your design, and why? Show class diagrams for more complex designs.

This was a simple enough implementation. We were required to have one sub method that would take two character pointers. Knowing that the original string was not passed, I had to make use of creative array indexing to make a working solution.

III. Security Analysis: State the potential security vulnerabilities of your design. How could these vulnerabilities be exploited by an adversary? What would be the impact if the vulnerability was exploited?

One potential security flaw that could be exploited by an adversary would be the array indexing. In order to get the element in the string prior to the end character, I used `array[-1]`. This is not good practice, but it works because c-strings are contiguous in memory.

IV. Implementation: Outline any interesting implementation details in your solution. I used recursion to implement the string reversal method. I made it so that it would call the method again with the next characters inward and continue until the pointers passed were either equal or the first was greater than the last. As it worked back up the call stack it would reverse the string a character at a time.

V. Testing: Explain how you tested your program, enumerating the tests if possible. Explain why your test set was sufficient to believe that the software is working properly, i.e., what were the range of errors for which you were testing.

I used a total of ten test cases. There were eight edge cases that I had to test for. I then tried two that I expected the program to output a result. The edge cases were as follows: too many arguments, too few arguments, front index not a number, rear index not a number, front and rear index less than one, front index greater than rear index, and rear index greater than the length of the string.

VI. Summary/Conclusion: Present your results. Did it work properly? Are there any limitations? NOTE: If it is an analysis-type project, this section may be significantly longer than for a simple implementation-type project.

This program seems to work correctly. All the edge cases are handled well, and it outputs the correct values for correctly formatted input. As far as I know there are not any limitations within spec.

VII. Lessons Learned: List any lessons learned. For example, what might you have done differently if you were going to solve this problem again?

I learned a bit more about passing variables by reference. Using this I was able to process a string even though I didn't have the original string to use.