1. What are the most challenging aspects of the coursework task?

There were a lot of the challenges that I faced during the completion of the program. Error handling was one of the major problem that I faced, file names, non-integer shift values, and invalid mode selection was among the major problems that I faced. The another big challenge for me was to maintain the flow of the code and call the functions inside another function. In addition to this I also had some issues while debugging the mistake and catching errors in different part of my code.

2. How did you go about completing the task?

I completed the task in many small steps. I started with breaking down the problem into many smaller parts.at first I list down the set of instruction to the program to execute, than I addressed the requirement for the code and then created different functions for each tasks, which helped me to keep my code organized and handle the user errors easily in each functions. The main logic of the Caesar Cipher was kept inside encrypt and decrypt function which were connected by the systematic flow of code. I took a step by step approach to keep the program flow clear and focused on error handling. This made everything more organized and program could run smoothly. Finally, I tested my program with almost all the possible ways to check the available errors through the user side inputs.

3. What have you learned over the course of completing this coursework task?

This coursework helped me to understand the several important programming concepts. I got a better understanding on error handling, especially using try and except blocks for exceptions like FileNotFoundError and ValueError. It also clarifies file input and output using Python's built-in functions, opening the file in read and write modes. Working on the Caesar Cipher algorithm solidified my understanding of ASCII value manipulation. I also learned a lot about program flow control, especially with loops and conditions to guide user choices. For example, the message_or_file() function was crucial in figuring out whether the user wanted to read from a file or input directly into the console.

Overall, I improved my skills in structuring programs with multiple functions, which made my code organized. I also learned the importance of writing clear steps to run the program and thoroughly testing my program to ensure it works well in different situations. I realized the importance of testing and debugging for creating a smooth execution program.