**Module 5 Assignment: XOR Encryption**

Zane Milo Deso

Southern New Hampshire University

**XOR Encryption Summary**

**Introduction**

For CS 405 Module Five, I was tasked with implementing XOR-based encryption on a text file, loading the file into a string, and saving both encrypted and decrypted outputs in the specified format. This brief summary outlines my approach, the bug encountered during file reading, and the correction I applied.

**Process**

I began by writing three principal routines in Encryption.cpp. In encrypt\_decrypt, I looped over each character of the input string and applied an XOR with the character ‘z’, guarding against empty input or key lengths with assertions. In read\_file, I opened “inputdatafile.txt” and accumulated every line (including its terminating newline) into a single std::string so that the full Lorem Ipsum text would be available for processing.

In save\_data\_file, I constructed the output by concatenating the student’s name, the hard-coded date (2025-06-11), the key, and the transformed data, then wrote it to the specified filename. Finally, in main(), I chained these functions: reading the source file, extracting the student name, encrypting the complete text into “encrypteddatafile.txt,” decrypting it back, saving to “decrypteddatafile.txt,” and logging each step to the console.

### **Bug**

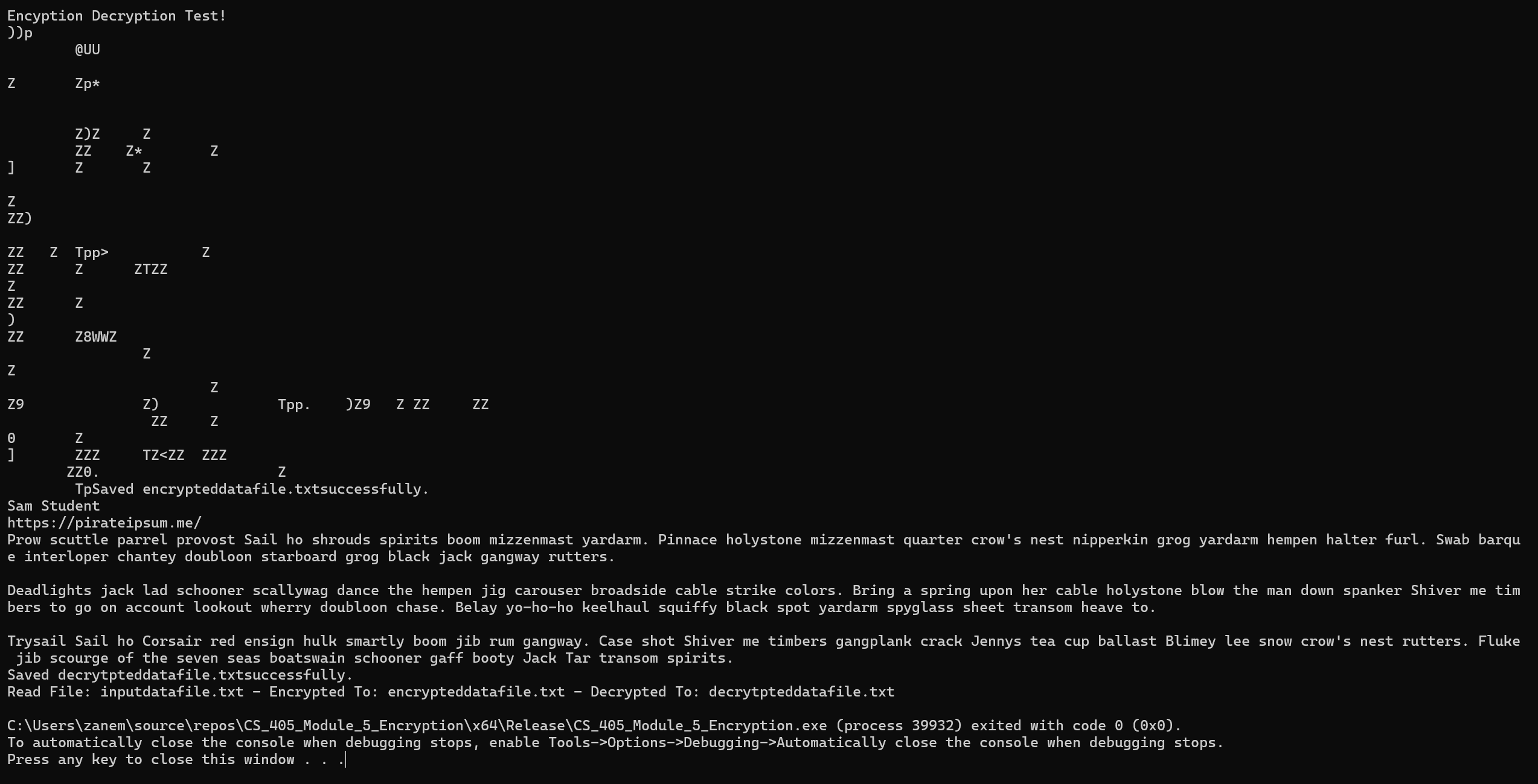
While testing, I noticed that both the encrypted and decrypted files lacked the Lorem Ipsum content. Investigation revealed that my original read\_file loop printed each line but never accumulated it—plus I mistakenly used std::getline(std::cin, file\_text) instead of appending file text. As a result, source\_string was always empty.

**Fix**  
 I replaced the faulty logic in read\_file with a single while (std::getline(f, line)) { file\_text += line; file\_text += '\n'; } loop and removed the extraneous std::getline(std::cin, ...) call. This ensured that source\_string correctly contained the entire contents of the input file before encryption.

## **Conclusion**

In conclusion, by refactoring read\_file to properly accumulate file contents, the program now fulfills all assignment requirements: XOR encryption accounting for key length, loading the specified file format, and saving both encrypted and decrypted files correctly.

## **Screenshot of Console Output**

Figure 1: Console output showing successful encryption and decryption.