COMP281 Assignment 2: Brief Report

Introduction

This report outlines the methodologies and strategies employed to address the programming tasks outlined in Part 1 and Part 2 of the coursework. The focus was on developing efficient and robust solutions that met the functional requirements and adhered to good coding practices.

Part 1: Run Length Encoding of ASCII Art

- **Task:** Implement a program that compresses and expands ASCII Art images using Run Length Encoding (RLE).
    - **Strategy:** For compression (C mode), the program reads each line of the ASCII Art, identifies sequences of repeating characters, and encodes them using RLE. For expansion (E mode), it decodes the RLE-compressed lines back to their original form. The compressLine and expandLine functions handle these tasks, utilising loops, conditionals, and string manipulation techniques to achieve the desired transformations. The program efficiently handles file I/O operations to read from and write to files, ensuring that the output aligns with the specified format.

Part 2: Searching a piece of text for a word

- **Task:** Develop a program that searches for a specific word in a given text, employing appropriate data structures and algorithms.
    - **Strategy:** This part uses a binary search tree (BST) to store and count unique words from the input text after cleansing them to remove punctuation and convert them to lowercase. The insert function builds the tree by placing words in alphabetical order, and the search function looks up the tree for the search word. The program outputs the count if the word is found; otherwise, it displays a "not found" message. This approach ensures efficient word lookup and counting, demonstrating effective use of dynamic memory allocation and tree data structures.

Part 2: Conclusion

Completing Part 1 and Part 2 of the assignment deepened my understanding of various aspects of C programming, from file handling and memory management to more advanced topics like data structures and network programming. Each solution focused on efficiency,

readability, and robustness, ensuring that the programs met the specified requirements and were structured to support maintainability and scalability. This exercise has significantly enhanced my programming skills, particularly in problem-solving, algorithmic thinking, and applying C programming concepts to real-world scenarios.