During this week, I worked on the other compression algorithm (Lempel-Ziv-Welch (LZW)) that I would be comparing my Huffman code algorithm. The completed code that I have been able to create shows as good of a compression score as the Huffman code that I have created, also reaching a compression rate of 48%. The work this week as been going well and I have I not stumbled on the same types of backseats that I was experiencing the previous week. The main reason for this is since I used the same idea for the compression algorithm that I did last week where I gave the most occurring letter the lowest footprint placeholder. Similarly it works in the way that if the letter ‘e’ is the most common letter in the text, it is given the smallest storage occupying placeholder. Moving on from this week I am going to be working on the UI for the two compression algorithms and implementing a way that both of these codes can be compared side by side so that it will be easy for any person wanting to test them to easily be able to see how they compare to each other and how easy they both are to use/ not easy they are to use.