You are required to carry out the tasks below in order to develop an optimal **Huffman Tree** with which to compress a file of ASCII text. **You should make use of the relevant template classes in the STL.**

You do not need to actually compress the file, just build the Huffman tree.

These tasks will encode and decode the file into a string of 0's and 1's using a Huffman tree

1. Given a text file, determine the frequency of each character in the text (**map** of character and frequency).
2. Build an optimal Huffman tree to represent these characters with these frequencies (maintain a **priority queue of trees**, removing and joining trees until only one tree remains – the final Huffman tree required) Hint: each Hoffman Tree should have a frequency data member, as well as a pointer to its root node.
3. Use the tree to map each character to the string of 0's and 1's needed to encode it (pre-order traversal of tree, store results in a **map** of character and string).
4. Use the map to encode the text to a string of 0's and 1's and write to an encoded file
5. Use the Huffman Tree to decode the text and write to another file.