CS 3610 Huffman Coder/Decoder

Design a Huffman code tree to encode and decode text files. The tree must be constructed based on a set of alphabets and their probabilities. Your objective is to assign the shortest codes to the characters with the highest probabilities. The alphabets and their probabilities will be stored in a file called "alphabet.txt". The alphabet file is organized as follows:

```
a .2
b .3
c .1
d .05
e .05
f .3
```

Lower case letters and uppercase letters are treated the same. You may convert all uppercase letters to lowercase before processing.

Write a C++ program that allows the user to encode a text file using the Huffman code tree created. Your program must be able to encode a file and decode it (restore it to its original English text). Punctuations and new lines should not be encoded or decoded and should maintain their locations inside the data files.

Your program should use command line arguments with the following options:

`-e : encode` convert letters to codes
`-d : decode` convert codes to letters
`-p : print` all characters with their codes and the average code length
`-h : help` (usage)

Report errors and exit the program if any of the following occur:

- File does not exist
- Wrong argument entered
- Wrong number of arguments

Examples:

```
./huffman -e inputFile.txt    outputFile.txt
./huffman -d inputFile.txt    outputFile.txt
./huffman -h
./huffman -p
./huffman
```
(display the usage)

Grading:
Programs that contain syntax errors will earn zero points.
Programs that use global variables, other than constants, will earn zero points.

(65 points):

- (20 points) Creating the Tree
- (20 points) Encoding
- (10 points) Decoding
- (5 points) Printing the characters and their codes
- (5 points) Error detection
- (5 points) Style and documentations

Follow the coding style outline on GitHub:

https://github.com/nasseef/cs2400/blob/master/docs/coding-style.md