Nadav Zinger
ECE 368

Project 2 Ms1

This project asks us to write two different programs. One titled huff.c and the other is unhuff.c. huff.c. Both take in a input file, huff.c takes in a uncompressed file and unhuff.c takes Ina compressed file. Huff.c compressed its input file while unhuff.c decompresses its file (they do the opposite).

The input file will be a pure ascii text file. For the first function I must convert the ascii text into a huffman encoded binary tree. From the binary tree I can compress the file into the actual text. For the second function the input file contains the actual text and I must revert back to my tree to retrive the huffman code for each character. From the huffman code I can then finally decompress the file into ascii text once again.

Characters are assigned huffman encoded bits based on how frequently they appear in the ascii text. Based on this use the greedy huffman algorithm to build the tree with every character having 0 (to the left) and 1 (to the right) “direction” path in the tree. Every character will be a leaf node so follow every root to leaf path to finally compress the file. Use this same tree in order to decompress the file.

My strategy to code this project is to look back at my ECE 264 project with huffman encoding. Based off that project I will know how to properly read the input file, convert ascii to huffman tree and finally do the reverse of that.