HACETTEPE UNIVERSITY

Department of Computer Engineering

BBM 203 – SOFTWARE LAB I

FALL 2020

Assignment IV Report

Name: Mert Emre

Surname: Öztürk

Number: 21986578

Note:I have two days extension and I am using them for this assignment.

My program compile this way---> make
And make command generate Main then we will use this executable files with command order by order.

EXPLANATION OF ENCODING ALGORITHM AND CODE:

I am running program this way---> ./Main -i input.txt -encode.

First of all, we read file and store character and its frequency in the map,then all of them and their frequency added ordered linked list through addToList function.*AddToList* function provide adding new element according to its frequency.Thus,all characters with frequency will store ordered in linkedlist. After all process, we can make huffman tree with using linkedlist.*MakeTree* function provide them.It get element from front of linked list(front element is smaller frequency than others) and add them to Huffman tree left and remove that element through *removeFromList* function from front of linkedlist .Same process are made right and top pointer.We will continue until linked list only has one element.If our linked list has only one element finish process and this function call *encodingTree* function.This function provide if it move the right add "1" to string or if it move the left add "0" to string after this string stored in the encodingTable map according the node's character.After encodingTree function, we open txt file with named "outForDecode" and saved tree with using *writeTree* function for decoding process and saved encoding table to output.txt for **./Main -s character** command.Finally,we print encoded string in the console with using *findEncodingStr* function.

EXPLANATION OF DECODING ALGORITHM AND CODE:

I am running program this way---> ./Main -i input.txt -decode

In this part,we will use saved tree created by encoding part.Firstly read file and store data to vector, secondly we will declare root pointer by HuffmanTree and create tree according to saved tree from previous section through *readTree* function.*ReadTree* function will create tree according to element of vector came from the file and we use with *decodeTree* function together take input from input.txt (this file should contain encoding string).*DecodeTree* function will continue along with string size.We will traverse string one by one and if bit is '1' we will move the right otherwise move the left.If we arrive the leaf node,the node's left and right child should be null,therefore this node is a character for decoding,we write that step by step result string.Finally we print result decoded string.This string is the same string from first input.txt.

EXPLANATION OF LIST TREE COMMAND:

I am running program this way---> ./Main -l list tree

In this part,we will use saved tree created by encoding part.This tree saved in "outForDecode.txt" and we will create a new Huffman Tree through *readTree* function.This function took element from vector one by one and added tree on suitable position.Finally we send the root pointer to *printTree* function and this function preorder traverse in the tree,then print console each node's char,frequency.