

# CSE102 – Computer Programming

## Homework #12

### Huffman Coding Tree

**Due Date: 17/06/2020**

**Hand in:** A student with number 20180000001 should hand in a 'c' file named 20180000001.c for this homework.

---

Huffman is a coding method used to compress data without loss in computer science. One of the biggest advantages of Huffman coding is that it makes an encoding according to the frequencies of the characters used, so that the frequently used characters take up less space.

Huffman encoding initially takes a large text as input, containing all the characters to be used later, if possible. The frequencies of the characters included in this large text (frequency of passing in the text) will determine the structure of the binary tree to be created. There are "characters" in the leaves of this binary tree, which is created by using the frequencies of the characters. During the movement from root to leaves (i.e. characters), codes of characters will be obtained when each left branch is referenced with the bit '0' and each right branch with the bit '1'.

The frequency of a character also refers to the level of compression. The more frequently a character is used, the fewer bits can be expressed in the Huffman coding tree.

**What is expected to be done within the scope of homework is explained below:**

1. Taking a text file (reference.txt) as input, creating the Huffman coding tree and encrypting each character **[40pts]**
2. Encrypting a text requested from the user with character codes created from the coding tree and writing it in a .dat file (encoded.dat) **[20pts]**
3. To decode the binary array in encoded.dat and write it to decoded.txt. **[20pts]**
4. Calculation of the file size difference between encryption (encoded.dat) and decryption (decoded.txt). **[20pts]**

**You can use the resources below for a detailed explanation of Huffman Coding:**

**Simulation:** <https://people.ok.ubc.ca/ylucet/DS/Huffman.html>

**Detailed description:** <https://www.studytonight.com/data-structures/huffman-coding>

**General Rules:**

1. We do not give you any function prototypes. We expect that you are experienced enough to understand when to use methods and name them. These will also be graded.
2. The program must be developed on Linux based OS and must be compiled with GCC compiler, any problem which rises due to using another OS or compiler won't be tolerated.
3. Note that if any part of your program is not working as expected, then you can get zero from the related part, even it is working partially.
4. Upload your ZIP or RAR file on to Moodle to deliver your homework. The zip file must consist of one C file that contains your solution. Name format can be found on the top of this homework sheet.
5. You can ask any question about the homework by sending an email to [bbuluz@gtu.edu.tr](mailto:bbuluz@gtu.edu.tr) or by using the forum in the Moodle page of the course.