PROJECT DESCRIPTION

In this project, you are required to use the Huffman coding function in Matlab to encode and decode text files. The project is composed of three parts to guide you through the required task. You will need to use the following Matlab functions:

- **huffmandict()**
- **huffmanenco()**
- **huffmandeco()**

Part I

Assume a random variable X has M symbols with the following distribution:

$$f_X(x) = 1/6, \qquad \text{for } x = 1, 2, \cdots, 6$$

1) Map each symbol to a fixed-length code with the minimum number of bits needed to represent M symbols.
2) Find a Huffman code and calculate its average code length.
3) Compare the average code lengths in parts 1) and 2) above to each other and to the Entropy of the random variable. Comment on your observations.
4) Using the attached .mat file of sample data from the random variable:
   - Verify the probability mass function of the random variable.
   - Generate the source coded codewords for both the fixed-length code and the Huffman code. Compare the total number of bits to be transmitted in each case.
   - Decode the codewords back to symbols and check for any losses compared to the original data.

Part II

Repeat **Part I** for random variables Y and Z with the following distributions:

$$f_Y(y) = \begin{cases} 0.5^y, & \text{for } y = 1, 2, \cdots, 5 \\ 0.5^5, & \text{for } y = 6 \end{cases}$$

$$f_Z(z) = \begin{cases} 0.05, & \text{for } z = 1 \\ 0.10, & \text{for } z = 2 \\ 0.30, & \text{for } z = 3 \\ 0.25, & \text{for } z = 4 \\ 0.15, & \text{for } z = 5 \\ 0.15, & \text{for } z = 6 \end{cases}$$

Comment on your observations of **Part II** compared to **Part I**.

Part III

Now assume you have a text .txt file, composed of English letters, both small (a, b, ..., z) and capital (A, B, ..., Z) in addition to the space ( ), comma (,) and the period (.). You are required to compress the file using Huffman codes.

Write a Matlab function to:

- Read the .txt file
- Extract characters from the file
- Find the PMF of the characters
- Find a Huffman code
- Encode the text file into a binary one using the Huffman code
- Encode the text file into a binary one using ASCII code
- Compare the binary file sizes of the previous two parts and find the compression percentage achieved by using the Huffman code
- Decode the binary file back to a text file and check it for errors.

Note that your function should be generic such that it takes a .txt file as an input and outputs an encoded binary file and a decoded .txt file.

Deliverable

Deliver the following:

1) The written software for each part separately, with proper commenting.
2) A PDF report summarizing the outcomes of each part, including your comments.

INSTRUCTIONS

- You can work this reports in teams up to **3** members per team.
- Write a full report including all requirements of the deliverable.
- Late submissions are not allowed.
- **All team members should expect to be asked about all the report parts.**.
- Duplicate reports will be penalized with zero grade.
- The grading criteria of the **project** will be as follows:

  - **60%:** Completeness and correctness of every deliverable (as per the .pdf report).
  - **20%:** Clarity of figures, and proper labeling (as per the .pdf report).
  - **20%:** Report writing and organization.