# PROJECT 6 (35 + 10 +10 points)

**HUFFMAN ENCODING/ DECODING AND GREEDY ALGORITHM**

**Huffman Coding** is an [entropy encoding](http://en.wikipedia.org/wiki/Entropy_encoding) [algorithm](http://en.wikipedia.org/wiki/Algorithm) used for [lossless data compression](http://en.wikipedia.org/wiki/Lossless_data_compression). It refers to the use of a [variable-length code](http://en.wikipedia.org/wiki/Variable-length_code) table for encoding a source symbol (such as a character in a file) where the variable-length code table has been derived in a particular way based on the estimated probability of occurrence for each possible value of the source symbol.

1. **Variable length coding:**
2. **Input: a text of 3 or more lines including 26 characters, comma, spaces and period.**

**Display the input texts.**

**Pre-determine frequencies of each character by looking up dictionary.**

Using Huffman variable length encoding and decoding, implement a

program in Java or Python to encode and decode a given text,

1. Determine variable length binary code for each character and

**display the compressed encoded codes for the entire input text.**

1. Use the compressed binary code as input, decode it back to the

original text **and display the same.**

**II. Fixed length coding:** (use 5 bits or 8 bits coding to represent each character; If you use 5 bits, you may create your own character representation. If you use 8 bits, you may use ASCII representation.)

**Do the same as above in part I.**

1. **Compare the numbers of bit representation of each text in both variable length and fixed length by inputting three different paragraphs of texts. For convenience, you may use the digit “1” to represent the bit “1”, and the digit “0” to represent the bit “0”.**

**Your program output must show proper information to be understood well by the reader/viewer.**