**The Fabulous Morse Code Encoder and Decoder:**

Back in the day, primative earthlings communicated with each other by throwing potatoes at a distant drum. But we will leave that simulation for another time. You will be writing a Morse Code Encoder (English to Morse Code) and a Morse Code Decoder (Morse Code to English).

**Specifications:**

You will find in this folder a file called *MorseCode.java*. It contains two parralel arrays of strings. One contains English characters and the other contains their Morse Code equivalents.

You will also find a file called *english.txt* which contains text in English, as well as *code.txt* which containst text in Morse code. Each encoded word in *code.txt* is separated by a ‘/’ character.

Your task is to write a program (object definitions and driver) that will ask if the user wants to encode or decode.

If the user chooses to encode, it will then ask for the file name that contains text in English. Your program must write out to the screen the direct translation in Morse Code, with each word separated with a ‘/’.

If the user chooses to decode, it will then ask for the file name that contains text in Morse Code. Your program must write out to the screen the direct translation in English. Get it?

You must use a type of Map Object. Read from the static final arrays into the Map at the beginning of the program. Think about which type of Map to use (hash map or tree map). Your choice is vital to the efficiency of the program. You have complete freedom as to how you do the rest, but PLAN IT FIRST.

**Grading:**

15% of the grade will be internal documentation (pre/post conditions, efficiencies, commenting variables and cryptic code)

25% of the grade will be programming style. Did you create objects and methods that will be easy to use in ANY encode/decode application? Would the same methods work if we wanted to send all output to a file instead of the screen? Was the right type of Map data structure chosen? Is it as efficient as possible?

**Extra credit:**

Give the user the option to send the output to a file instead of the screen.

Give the user the option to hear the Morse code read in or the Morse code translated.

Accommodate the backspace command. To indicate that a mistake has been made and for the receiver to delete the last word, send **........** (eight dots).

The timing of sounds:

If the duration of a dot is taken to be one unit of sound then that of a dash is three units of sound. The space between the components of one character is one unit, between characters is three units and between words seven units.