

## Assignment #7 – Morse Code Decoder

**Date Assigned:** Thursday, November 12, 2009

**Date Due (Part I):** Thursday, November 19, 2009

**Date Due (Part II):** Tuesday, November 24, 2009

**Points:** 75

Write a C program that will decode files of Morse Code into plain English. You are to implement the solution in two parts.

### Part I

For Part I you are to implement the Binary Tree ADT, found in the CS300 Public directory on zeus as a project called BinaryTree. Set up this module as all others with folders Headers, Sources, ... as before. You are to include a driver called binarytreedriver.c that tests all of the functions that are a part of the Binary Tree module found in the header file binarytree.h. Save this project as a tarball called 07yourlastnameI.tar.gz. When extracted, your tarball is to produce a project called BinaryTree. I will change into the BinaryTree directory, type **make clean**, then **make**, then **./binarytreedriver**.

The Binary Tree module has a print function to print the binary tree just constructed on its side. The data content of each node in the tree is to be printed so that the spatial relationship among the nodes is shown on the output. In particular, we choose to display the tree "on its side." For example:

```

                Z
              E   T
            I   A   N   M
```

will be displayed this way:

```

                M
              T   N
            Z   A
          E
        I
```

Add lines to this tree by hand before you turn in your final output.

Note1: Your tree display is built by using the following rules:

- 1) Each datum is printed on a separate line
- 2) A datum at level  $i$  is preceded by  $4(i-1)$  blanks. This means that all values at the same tree level will appear in a column on the printed output.
- 3) The right subtree of each node is printed on lines above the line containing the node, which in turn appears before lines containing the left subtree

## Part II

For this part of the assignment you are to build a project called MorseCode that takes a Morse Code file and prints the decoded Morse Code in English. Set up this module as all others with folders Headers, Sources, ... as before. You are to include a driver called **morsecodedecoder.c** that builds a Morse Code Tree using `morsecode.txt` and then decodes a file **message.txt** in the folder Testcases. Your driver is to include and use the Binary Tree module created in Part I to build the Morse Code Tree. Save this project as a tarball called `07yourlastnameII.tar.gz`. When extracted, your tarball is to produce a project called MorseCode and any other projects needed to make and run `morsecodedriver`. I will clean all other directories first and then I will change into the MorseCode directory, type **make clean**, then **make**, then **./morsecodedriver** at which point your program will open the file `message.txt` and print the decoded text.

The file to be decoded contains morse code that is to be decoded. Each set of morse code characters representing one encoded value will be separated by one space or one or more newline characters. Whenever you read a newline, output a newline in your outputted message.

Along with printing the source code and decoded message, print out the Morse Code Tree and fill in the lines by hand.

Using the file **morsecode.txt** found in the CS300Public folder to create your Morse Code Tree. Each line in the file contains a character, followed by a space, followed by the character's morse code. We will talk in class about how to build the tree.