In homework 17 and homework 18, you will learn two important parts of Huffman compression.

In homework 17, you need to write a program that takes the post-order traversal description of a Huffman compression tree. The program rebuilds the tree and outputs the code book.

In homework 18, your program extends homework 17 and compresses the end of a file. We consider only a short portion of a file so that debugging can be easier.

What does homework 17 need to do? Consider this example. The input is the description of a code tree. For example, the input is

One ei, one M, one #, one gee, zero, zero, zero. One cee, one ass, zero, zero, zero.

The code tree is shown here.

From the code tree, your program needs to create the code book. What is the code book? It maps the characters to the codes.

In this example, the code for ei is zero zero. The code for M is zero, one, zero. The code for # is zero, one, one, zero.

For homework 17, there is no need to manipulate individual bits.

Homework 18 takes two inputs: the description of the compression tree and part of an article that includes the characters in the tree. In homework 18, your program needs to handle individual bits.

Suppose the article is #, ei, cee, gee, M, ass, #, ei, cee.

Based on the code book, the code should be

Zero, one, one, zero for #;

Zero, zero for ei;

One zero for cee;

Zero one, one, one for gee;

Zero, one, zero for M;

One, one for ass;

Zero, one, one, zero for #;

Zero, zero for ei;

One, zero for cee.

The program needs to put these bits together into bytes.

The unused bits of the last byte are zeros.

You can use the Linux command X X dee minus bee to see the bits in a file for debugging.