Laboratory Exercise 3

cpe 357 Fall 2022

Due by (or before) 11:59pm, Monday, October 17th.
The Laboratory Exercise is to be done individually (because it's really the first part of Asgn3).

Problems

There are no written exercises this week. Stay tuned for next week.

Laboratory Exercises

This laboratory exercise is to create an initial version of Assignment 3, Huffman encoding and decoding. For this lab, you are to create a program, `htable` that will generate the table of encodings appropriate for a given file.

Usage:

```
htable filename
```

Your program must:

- Read the input file—given on the command line—and build the Huffman code tree according to the rules given in Assignment 3; and
- Write this encodings described by this code tree to standard out according the the following format:
  - Only bytes that are present in the file are included in the table.
  - Bytes are included in the table in numerical order.
  - Each line of the table consists of the byte as a two-digit hexadecimal number followed by a colon and a space, followed by the binary encoding represented by the characters '`0`' and '`1`'.
    Example: `0x61:  101`

You may use any kind of IO you like for this, the restrictions for Assignment 3 do not apply.

What to turn in

For the Laboratory Exercise: Submit via `handin` in the CSL to the `lab03` directory of the `pn-cs357` account:

- your source files.
- A makefile (called `Makefile`) that will build your program when given the command `make htable`.
- A README file that contains:
  - Your name(s). In addition to your names, please include your Cal Poly login names with it, in parentheses. E.g. `(pnico)`
  - Any special instructions for running your program.

– Any other thing you want me to know while I am grading it.

The README file should be **plain text,** i.e, **not a Word document**, and should be named "README", all capitals with no extension.

Sample Runs

I have placed a runable version of `htable` in the CSL in `~pn-cs357/demos` as `htable`.

```
% cat test
aabbccddd
% htable test
0x0a: 100
0x61: 101
0x62: 00
0x63: 01
0x64: 11
%
```