CSC 148H1 Winter 2008 Midterm Test
Duration — 50 minutes
Aids allowed: none

Student Number: |_|_|_|_|_|_|_|_|_|_|_|

Lab day, time, room: _____

Last Name: _____ **First Name:** _____

Lecture Section: L5101 **Instructor:** Gries

---

*Do **not** turn this page until you have received the signal to start.*
(Please fill out the identification section above, **write your name on the back of the test**, and read the instructions below.)
Good Luck!

This test consists of 4 questions on 8 pages (including this one). *When you receive the signal to start, please make sure that your copy is complete.*
Comments and docstring are not required except where indicated, although they may help us mark your answers. They may also get you part marks if you can't figure out how to write the code.
If you use any space for rough work, indicate clearly what you want marked.

1: _____/10
2: _____/10
3: _____/10
4: _____/10

TOTAL: _____/40

Question 1. [10 MARKS]

The textbook discussed two ways to represent binary trees: using nested lists, and using node objects. For example, a tree with `'A'` as the root value, `'B'` as `'A'`s left child, and no right child, could be represented using this list: `['A', ['B', None, None], None]`. An empty tree is just `None`.

Here is a `Node` class:

```
class Node:
    def __init__(self, v, L=None, R=None):
        '''A new Node with value v, left child Node L, and right child Node
        R.'''
        self.value = v
        self.left = L
        self.right = R
```

Complete the recursive function below.

```
def to_nodes(tree):
    '''Convert tree, which is a nested list representing a binary tree, to a
    binary tree made up of Node objects and return the root Node. tree[0] is
    the root, tree[1] is the left subtree, and tree[2] is the right
    subtree.'''
```

Question 2. [10 MARKS]

Part (a) [6 MARKS]

Follow the Huffman tree-building process for the message `"abracadabra"`. (Don't bother with EOF.)

- Initial forest (just draw circles with values inside to represent the nodes):

- Forest after one step in the tree creation (after two nodes are combined):

- Continue drawing the forests until the process is finished. Separate each step with a horizontal line.

Part (b) [2 MARKS] What is the binary representation for the letter `'c'`? _____

Part (c) [2 MARKS] Is your Huffman tree the only possible one for this message? Circle the answer:

`Yes` `No`

Question 3. [10 MARKS]

Consider the following method.

```
def average(L):
    '''Return the average of the numbers in L.'''
    sum = 0.0
    count = 0
    for value in L:
        sum = sum + value
        count = count + 1
    return sum / count
```

Part (a) [3 MARKS]

There are at least two different possible errors that might occur during execution of `average`. One is when the list is empty. Describe another one.

Part (b) [5 MARKS] Here is the output when you call `average` with an empty list:

```
ZeroDivisionError: integer division or modulo by zero
```

That isn't very helpful. Rewrite `average` (including the docstring) to raise an `EmptyListError` with a better error message when given an empty list; also define class `EmptyListError`.

Part (c) [3 MARKS]

Write two `nose` tests that test whether the appropriate exceptions are raised in the two error situations as expected. (Note: we don't expect you to use all the space on this page.)

Question 4. [10 MARKS]

Part (a) [4 MARKS] Consider the following code.

```
def mystery(s):
    if len(s) == 0:
        return ""
    elif len(s) == 1:
        return s
    elif s[0] == s[1]:
        return mystery(s[1:])
    else:
        return s[0] + mystery(s[1:])
```

What does `mystery` do? **Describe in one English sentence.**

Part (b) [6 MARKS]

Write a recursive function called `to_words` that takes a single `int` parameter and returns a `str` that contains the digits in the `int` in English.

For example, the call

```
to_words(23561)
```

would result in the following:

```
'two three five six one '
```

Assume the following list of `str`s is defined in the file that contains the function you are writing.

```
NUMBERS = ['zero', 'one', 'two', 'three', 'four', 'five', 'six', 'seven',
       'eight', 'nine']
```

Hint: Use the `/` and `%` operators.

Use this page for rough work and for any answers that didn't fit.

Last Name: _____ **First Name:** _____