CSC 108H1 S 2012 Test 2
Duration — 45 minutes
Aids allowed: none

Student Number: ⌊_⌊_⌊_⌊_⌊_⌊_⌊_⌊_⌊_⌋

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

Lecture Section: L5101   **Instructor: Campbell**

---

*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 midterm consists of 4 questions on 6 pages (including this one). *When you receive the signal to start, please make sure that your copy is complete.* Comments and docstrings 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. No error checking is required: assume all user input and all argument values are valid.

If you use any space for rough work, indicate clearly what you want marked.

1: _____/ 3

2: _____/ 3

3: _____/ 8

4: _____/10

TOTAL: _____/24

Question 1.   [3 MARKS]

For each block of code in this question, write its output in the box below it. If it would generate an error, say so, and give the reason for the error.

Part (a)   [1 MARK]

```
inner2 = ['pizza', 'party']
nested = [inner2[:], inner2[:]]
nested[0].insert(1, 'birthday')
print nested
```

Part (b)   [2 MARKS]

```
inner2 = ['x', 'y', 'z']
nested2 = [inner2, inner2]
print nested2[1]
```

```
# continued from above
nested2[1].append('new')
print nested2
```

Question 2.   [3 MARKS]

In A2, a `board` was a `list of lists of strs` where each inner list represented a row on the board. Complete the following function, according to its docstring description.

```
def display_column(board, col):
    '''(list of lists of strs, int) -> NoneType
    Print column col of the board.  You may assume: 0 <= col < len(board)'''
```

Question 3.   [8 marks]

Part (a)   [5 marks] Complete the following function according to its docstring description.

```
def is_first_letter(sentences, letter):
    '''(list of lists of strs, str) -> bool
    The strs in the lists in sentences are lowercase words and letter is a single
    lowercase letter. Return True if one or more words in sentences starts with letter,
    and return False otherwise.'''
```

Part (b)   [3 marks] In the question below, fill in the box with python code that will make the program behaviour match the comments. You may **not** make any other changes to the code.

```
if __name__ == '__main__':
    sentences = [['hi', 'there'], ['this', 'is', 'fun'], ['hooray']]

    # Continually prompt the user (using raw_input) to enter a letter until they enter
    # a letter that one or more words in sentences starts with.
```

```



```

```
    print "One or more words starts with %s.", % (letter)
```

Question 4.   [10 MARKS]

A file is used to keep track of phone numbers where each line is of the form:
DDD-DDD-DDDD
where D is a digit (a number between 0 to 9, inclusive).

Here is a sample "phone file":

416-555-1111
905-555-4444
416-555-2222
647-555-8888

A "phone list" is generated based on a "phone file" like the one shown above.

The "phone list" for the file above is:
['(416) 555-1111', '(416) 555-2222', '(905) 555-1111', '(647) 555-8888']
Note: the format in the file differs from the list (e.g., 416-555-1111 as compared to (416) 555-1111).
The first three digits of a phone number are the area code.

Part (a)   [6 MARKS] Complete the following function according to its docstring description.

```
def get_phone_list(phone_file):
    '''(file) -> list of strs
    Return the "phone list" for the "phone file" phone_file.'''
```

Part (b)   [4 marks] Complete the following function according to its docstring description.

```
def count_area_code(phone_list, area_code):
    '''(list of strs, str) -> int
    phone_list is a "phone list" and area_code is a 3-digit number.
    Return the number of phone numbers from phone_list with area code area_code.'''
```

You may use the space below for rough work. This section will not be marked unless you clearly indicate the part of your work that you want us to mark.

Short Python function/method descriptions:

```
__builtins__:
  len(x) -> int
    Return the length of the list, tuple, dict, or string x.
  open(name[, mode]) -> file object
    Open a file.
  range([start], stop, [step]) -> list of integers
    Return a list containing the integers starting with start and ending with
    stop - 1 with step specifying the amount to increment (or decrement).
  raw_input([prompt]) -> str
    Read a string from standard input.  The trailing newline is stripped.
file (also called a "reader"):
  F.close() --> NoneType
    Close the file.
  F.read([size]) -> string
    Read at most size bytes; with no size, read until EOF.
  F.readline([size]) -> string
    Read next line, retaining newline; return empty string at EOF.
float:
  float(x) -> float
    Convert a string or number to a floating point number, if possible.
int:
  int(x) -> int
    Convert a string or number to an integer, if possible.  A floating point
    argument will be truncated towards zero.
str:
  str(x) -> str
    Convert an object into its string representation, if possible.
  S.find(sub[,i]) -> integer
    Return the lowest index in S (starting at S[i], if i is given) where the
    string sub is found or -1 if sub does not occur in S.
  S.replace(old, new) -> string
    Return a copy of string S with all occurrences of the string old replaced
    with the string new.
  S.split([sep]) -> list of strings
    Return a list of the words in S, using string sep as the separator and
    any whitespace string if sep is not specified.
  S.startswith(prefix) -> boolean
   Return True if S starts with the specified prefix and False otherwise.
  S.strip() --> string
    Return a copy of S with leading and trailing whitespace removed.
list:
  L.append(x) --> NoneType
    Append x to the end of the list L.
  L.index(value) -> integer
    Return the lowest index of value in L.
  L.insert(index, x) --> NoneType
    Insert x at position index.
```