CSC 108H1 S 2009 Test 3
Duration — 35 minutes
Aids allowed: none

**Student Number:** _____

**Lab time, room:** _____

**Family Name:** _____ **First Name:** _____

---

*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 3 questions on 7 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.
If you use any space for rough work, indicate clearly what you want marked.

# 1: _____/ 8
# 2: _____/10
# 3: _____/ 7

TOTAL: _____/25

## Question 1. [8 MARKS]

Each row in the table below shows a series of statements to be interpreted by the Python shell. For each, describe any requirements that must be met in order for the statements to execute without error. For example, you might say the "x must be a float that is greater than 1." If there is no way for the code to run without error, write "impossible" and say why.

| **Python statements** | **Requirements (or "impossible" plus reason)** |
|---|---|
| `for (a, b) in m.items():`<br>`    print a % b` | |
| `d = {1: 5, "abc": "hello", 3: 77}`<br>`d[p] = q` | |
| `L = [9, 11, 3, 4, 8, 23, -5, 14]`<br>`L[i] = j` | |
| `T = ("Spring", "almost", "here")`<br>`T[s] = t` | |

## Question 2. [10 MARKS]

**Part (a)** [6 MARKS] Complete the following function according to its docstring description. Note that the word maternal means "relating to a mother."

```
def maternal_list(mother_of, name):
    Return a list where the first item is str 'name' and the subsequent items are the names
    of maternal ancestors of 'name', in order from youngest to oldest, as found in dict
    'mother_of'. Each key in 'mother_of' is the name of a child and the corresponding value
    is that child's mother.  For example, if the dictionary is
            {"jo": "ani", "jane": "di", "lina": "jo", "ani": "frida", "di": "cate"}
    and name is "lina", then the result should be
            ["lina", "jo", "ani", "frida"]'''
```

**Part (b)** [4 MARKS] Fill in the table below with four good test cases for `maternal_list`. For each, give the specific values for `mother_of` and `name` that you propose, and the purpose of the test case, *e.g.,* "name with even length" (although that's not a good test case for this function!).

Your cases should be different form each other, and should each test something significant.

| `mother_of` | `name` | Purpose of this test case |
|---|---|---|
| | | |
| | | |
| | | |
| | | |

## Question 3. [7 MARKS]

Complete the following function according to its docstring description.

```
def num_comments(f):
    '''f is an file that has been opened for reading, and it contains python code.  Return
    the number of lines in the file whose first non-whitespace character is "#".'''
```

*[Use the space below for rough work. This page 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) -> integer
    Return the length of the list 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 stop and ending with
    stop - 1 with step specifying the amount to increment (or decrement).
    If start is not specified, the list starts at 0.  If step is not specified,
    the values are incremented by 1.
dict:
  D.get(k) --> value
    Return the value associated with the key k in D.
  D.has_key(k) --> boolean
    Return True if k is a key in D and False otherwise.
  D.keys() --> list of keys
    Return the keys of D.
  D.values() --> list of values
    Return the values associated with the keys of D.
  D.items() -> list of 2-tuples.
    Return a list of D's (key, value) pairs.
file (also called a "reader"):
  F.close()
    Close the file.
  F.read([size]) -> read at most size bytes, returned as a string.
    If the size argument is negative or omitted, read until EOF is reached.
  F.readline([size]) -> next line from the file, as a string. Retain newline.
    A non-negative size argument limits the maximum number of bytes to return (an incomplete
    line may be returned then).  Return an empty string at EOF.
str:
  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) -> bool
    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)
    Append x to the end of the list L.
  L.index(value) -> integer
    Returns the lowest index of value in L.
  L.insert(index, x)
    Insert x at position index.
  L.sort()
    Sorts the list in ascending order.
```