

CSC 108H5 Fall 2007 Test 3

Duration — 30 minutes

Aids allowed: none

**Student Number:** \_\_\_\_\_

**Lab day, time, room:** \_\_\_\_\_

**Last Name:** \_\_\_\_\_

**First Name:** \_\_\_\_\_

**Lecture Section:** L0101

**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 test consists of 3 questions on 6 pages (including this one). When you receive the signal to start, please make sure that your copy is complete. Comments 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: \_\_\_\_\_ / 6

# 2: \_\_\_\_\_ / 8

# 3: \_\_\_\_\_ / 6

---

TOTAL: \_\_\_\_\_ / 20

**Question 1.** [6 MARKS]

Complete the following function.

```
def find_partner(partners, student):
    '''Given partners (a list of lists) and student (a string containing a
    student's username), return the username of the student's partner, or
    the empty string if the student does not have a partner. Each element
    of partners is either a one-element list containing a username (a string),
    to represent a student without a partner, or a two-element list containing
    two usernames (both strings), for partners. The student will appear
    in partners exactly once.'''

```

**Question 2.** [8 MARKS]**Part (a)** [4 MARKS]

```
def mystery(a, b=0):
    i = len(a) - 1
    c = False

    while i >= 0 and not c:
        if a[i] == b:
            c = True
        else:
            i -= 1
    return i
```

Show variable values after each line has executed:			
a:	b:		
i:	c:		
During 1st iteration	During 2nd iteration	...	During last iteration
i: c:	i: c:		i: c:
a[i]: b:	a[i]: b:		a[i]: b:
c:	c:		c:
i:	i:		i:
value returned:			

What is returned by the function call `mystery([6, 1, 4, 1, 2], 1)`?

In the table above, trace the variable values during execution of the function. For each blank in the table, fill in the value of the variable specified or write “not reached” if that line was not executed. You do **not** need to complete the section of the table labeled ‘...’. You only need to complete the section for the last iteration if the loop iterates 3 or more times.

**Part (b)** [4 MARKS]

```
def mystery(a, b=0):
    i = len(a) - 1
    c = False

    while i >= 0 and not c:
        if a[i] == b:
            c = True
        else:
            i -= 1
    return i
```

Show variable values after each line has executed:			
a:	b:		
i:	c:		
During 1st iteration	During 2nd iteration	...	During last iteration
i: c:	i: c:		i: c:
a[i]: b:	a[i]: b:		a[i]: b:
c:	c:		c:
i:	i:		i:
value_returned:			

**Note:** the code is Part(b) is the same as the code in Part(a).

What is returned by the function call `mystery([3, 5])`?

In the table above, trace the variable values during execution of the function. For each blank in the table, fill in the value of the variable specified or write “not reached” if that line was not executed. You do **not** need to complete the section of the table labeled ‘...’. You only need to complete the section for the last iteration if the loop iterates 3 or more times.

**Question 3.** [6 MARKS]

Complete the following function.

```
def extract_character(pixel_list):
    '''Given a list, pixel_list, containing 3 pixels, return the character
    hidden in the list as a string. One digit of the ASCII value representing
    the character is hidden in the one's place of the green component of each
    pixel in the list. The first pixel in the list contains the hundred's
    component of the number. The second pixel contains the ten's component,
    and the third contains the one's component.'''

```

**Short Python function/method descriptions:**

```
__builtins__:  
    len(object) -> integer  
        Return the number of items of a sequence or mapping.  
    range([start[, stop[, step]]) -> list of integers  
        Return a list containing an arithmetic progression of integers.  
        range(i, j) returns [i, i+1, i+2, ..., j-1]; start defaults to 0.  
        When step is given, it specifies the increment (or decrement).  
    open(name[, mode]) -> file object  
        Open a file using the file() type and return a file object.  
    ord(c) -> integer  
        Return the integer ordinal of a one-character string.  
    chr(i) -> character  
        Return a string of one character with ordinal i; 0 <= i < 256.  
  
list:  
    L.append(object)  
        Append object to end.  
    L.insert(index, object)  
        Insert object before index.  
    L.remove(value)  
        Remove first occurrence of value.  
  
picture:  
    get_blue(pixel) --> int  
        Return the value of blue (between 0 and 255) in the given pixel.  
    get_green(pixel) --> int  
        Return the value of green (between 0 and 255) in the given pixel.  
    get_pixels(picture) --> list  
        Takes a picture as input and returns the sequence of pixel objects in the picture.  
    get_red(pixel) --> int  
        Return the value of red (between 0 and 255) in the given pixel.  
  
str:  
    S.isalpha() -> bool  
        Return True if all characters in S are alphabetic and S contains at least one character,  
        False otherwise.  
    S.split([sep [,maxsplit]]) -> list of strings  
        Return a list of the words in the string S, using sep as the delimiter string.  
        If maxsplit is given, at most maxsplit splits are done. If sep is not specified, any  
        whitespace string is a separator.  
    S.strip([chars]) -> string or unicode  
        Return a copy of the string S with leading and trailing whitespace removed.
```

Last Name: \_\_\_\_\_ First Name: \_\_\_\_\_