CSC 108H1 Fall 2007 Test 4
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 2 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: _____/10
2: _____/ 6
TOTAL: _____/16

Question 1. [10 MARKS]

Complete the following functions.

Part (a) [4 MARKS]

```
def read_player_location_data(f):
    '''f is an open text file containing lines of the form:
        player_num:team_name:x:y

    Return a dictionary that contains (player_num, team_name) tuples as keys
    and (x, y) tuples as values.

    team_name is a str that may contain letters, numbers, and spaces.
    player_num, x and y are ints.
    '''
```

Part (b) [6 MARKS] You may assume that colour values are in the range 0-255, and the x and y coordinates are in the range 0-99.

```
def display_player_locations(teams, players):
    '''Return a 100x100 picture with a black background with the (x, y) location
    for each player from the dictionary players coloured according to her team's
    colour in the dictionary teams.

    Dictionary teams contains team_names as keys, and
    (red_value, blue_value, green_value) tuples as values.

    Dictionary players contains (player_num, team_name) tuples as keys,
    and (x, y) tuples as values.

    red_value, blue_value, green_value, player_num, x and y are ints.
    team_name is a str that may contain letters, numbers, and spaces.
    '''
```

Question 2. [6 MARKS]

Recall that you wrote the following function in Assignment 4:

```
def increment_count(d, key):
    '''Increment the integer associated with the str key in dict d if the key
    is already in the dictionary; add the key to the dictionary with value
    1 if key is not in d.
    '''
```

Complete the following functions. You may use `increment_count` in either function.

Part (a) [2 MARKS]

```
def count_items(L):
    '''Return a dictionary where the keys are the items of list L and the
    values are the number of times that each item occurs in L.
    '''
```

Part (b) [4 MARKS] For function `mode`, you may assume `len(L) >= 1`. You may also use `count_items`.

```
def mode(L):
    '''Return the mode of list L. (The mode is the item that occurs most
    frequently.) If there is more than one mode, return any one of them.
    '''
```

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 using the file() type and return a file object.
  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).
str:
  S.find(sub) -> integer
    Return the lowest index in S where the string sub is found or -1 if sub
    does not occur in S.
  S.index(sub) -> integer
    Like find but raises an exception 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.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.
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.
picture:
  get_blue(pixel) --> int, get_red(pixel) --> int, get_green(pixel) --> int
    Return the value of the specified color (between 0 and 255) in the given pixel.
  get_pixel(picture, x, y) --> pixel
    Return the pixel at the location (x, y) in the given picture.
  get_pixels(picture) --> list
    Takes a picture as input and returns the sequence of pixel objects in the picture.
  make_empty_picture(width, height) --> picture
    Return a blank (black) picture of the given dimensions.
  set_blue(pixel, value), set_green(pixel, value), set_red(pixel, value)
    Set the specified color component in the given pixel to the given value.
```

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