

CSC 108H5 Fall 2007 Test 2
Duration — 25 minutes
Aids allowed: none

Student Number: _____
Lab day, time, room: _____

Last Name: _____ First Name: _____

Lecture Section: L0101

Instructor: Petersen

*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 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: _____/12

2: _____/ 7

TOTAL: _____/19

Question 1. [12 MARKS]**Part (a)** [6 MARKS]

Complete the following function according to its docstring description.

```
def verify_sum(nums, total):  
    '''Return True if total (an int) is the sum of the integers in nums,  
    and False otherwise. nums is an expression with one or more single digit  
    numbers separated by + in the format "a+b+...+m".
```

Examples:

```
verify_sum("3+2+1+4", 10) should return True  
verify_sum("1+2", 4) should return False  
, , ,
```

Part (b) [6 MARKS]

Write a program that uses `pick_a_file` to prompt for a text file containing lines in the form “`a+b+...+m=T`”, one per line, and prints the lines where `a+b+...+m` is not equal to `T`. `T` is an integer, and all other integers (`a`, `b`, and so on) are single digits.

Question 2. [7 MARKS]

In A2, you implemented this function:

```
def double_my_digits(s):  
    '''Return a string consisting of the digits in str s, doubled. s must  
    consist entirely of digits. For example, double_my_digits("123456")  
    should return the string "24681012".  
    '''
```

Part (a) [3 MARKS]

Write a nose test function for `double_my_digits` that verifies that the function operates correctly if given the empty string.

```
import nose  
import warmup
```

```
nose.runmodule()
```

Part (b) [4 MARKS]

Provide two “interesting” strings other than the empty string that should be used to test `double_my_digits`. For each string, write one sentence to justify why it is of interest. (Note that you don’t need to write entire test functions.)

Short Python function/method descriptions:

`__builtins__`:

- `max(x, y, z, ...)` -> value
With two or more arguments, return the largest argument.
- `min(x, y, z, ...)` -> value
With two or more arguments, return the smallest argument.
- `len(x)` -> integer
Return the length of the list or string x.
- `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.

`float`:

- `float(x)` -> floating point number
Convert a string or number to a floating point number, if possible.

`int`:

- `int(x)` -> integer
Convert a string or number to an integer, if possible. A floating point argument will be truncated towards zero.

`str`:

- `str(x)` -> string
Convert an object into its string representation, if possible.
- `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.isdigit()` --> boolean
Return True if all characters in S are digits and False otherwise.
- `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.

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