

CSC 108H1 S 2009 Test 2  
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 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.

# 1: \_\_\_\_\_/10

# 2: \_\_\_\_\_/ 6

# 3: \_\_\_\_\_/ 8

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

TOTAL: \_\_\_\_\_/24

---

**Question 1.** [10 MARKS]

**Part (a)** [2 MARKS] Complete the following function according to its docstring description. Your answer must be a single line of code.

```
def add_to_month(month, n):  
    '''Month is an integer between 1 and 12 inclusive; n is an integer  
    that is at least 0 and represents a number of months. Return an integer  
    between 1 and 12 inclusive that represents what month it will be if we add  
    n to month.'''
```

**Part (b)** [3 MARKS] Complete the following function according to its docstring description. Again, your answer must be a single line of code.

```
def rotate(L, n):  
    '''L is a list and n is an integer. 0 <= n <= len(L). Return a new list which is  
    L rotated to the left by n positions.  
    For example, rotate([1, 2, 3, 4, 5, 6], 2) returns [3, 4, 5, 6, 1, 2].'''
```

**Part (c)** [5 MARKS] Complete the following function according to its docstring description.

```
def short_strings(L, n):  
    '''Return a new list that contains all the elements of list L whose  
    length is less than n.'''
```

**Question 2.** [6 MARKS]

Here is a mystery program:

```
def mystery(s):
    ans = []
    while s != "":
        if len(s) == 1:
            ans.append(s)
            s = ""
        else:
            ans.append(s[0] + s[1])
            s = s[2:]
    print "ans is", ans
    return ans

if __name__ == "__main__":
    var1 = "Pooh-bear!"
    var2 = mystery(var1)
    print "var1 is", var1
    print "var2 is", var2
```

What is the output of this program?

**Question 3.** [8 MARKS]**Part (a)** [4 MARKS] What is the output of the following program?

```
def loopy(s):
    new = ""
    i = 0
    while i < len(s):
        print "Outer", i, new
        if s[i] == "-":
            while s[i] == "-":
                print "Inner", i, new
                i = i + 1
            new = new + "-"
        else:
            new = new + s[i]
            i = i + 1
    print "final result", new

if __name__ == "__main__":
    loopy("a---ha")
```

**Part (b)** [4 MARKS] Write a docstring for loopy. (Pretend the print statements aren't there — they were temporary.)

*[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__`:

- `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.startswith(prefix)` --> bool  
Return True if S starts with the str 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.