CSC 108H1 S 2010 Test 2
Duration — 35 minutes
Aids allowed: none

Student Number: |_|_|_|_|_|_|_|_|_|_|

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

Lecture Section: L0101 **Instructor: Horton**

---

*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 4 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. No error checking is required: assume all user input and all argument values are valid.
If you use any space for rough work, indicate clearly what you want marked.

1: _____/ 3
2: _____/ 5
3: _____/ 6
4: _____/ 9

TOTAL: _____/23

Question 1. [3 MARKS]

For each of the following snippets of code, write the output that it would produce, or write "Error" if it would cause an error.

Code	Output (or "Error")
`d1 = {[1, 2]: "a", [3]: "z"}` `print d1.keys()`	

`d2 = {1: 2, 5: 9, -3: 4}`
`ans = 1`
`for k in d2:`
`  ans = ans * k`
`print ans`	

`d3 = {"hello": "goodbye", "fee": "fi"}` `print d3["hello"].upper()[5] + "am"`	

Question 2. [5 MARKS]

Complete the following function according to its docstring description. For the purpose of this question, a word is defined to be a string that is delineated by white space (blanks, tabs and newline characters).

You must not store more than one line of the file at a time. Answers that do **will receive a mark of zero**.

```
def word_count(filename):
    '''filename is a string that is the name of a file. Open that file, read
    it, and return the number of words in it.'''
```

Question 3. [6 MARKS]

Consider the following function:

```
def remove_one(s):
    '''s is a string. Return a list of all strings you can get by removing one
    character from s.'''
```

Part (a) [1 MARK]

What should `remove_one("happy")` return?

Part (b) [5 MARKS]

Write the body of this function.

Question 4. [9 MARKS]

Consider the following function:

```
def how_many(L, n):
    '''L is a list of ints and n is an int. Return the number of list elements
    one must add up (assuming one were to start at the beginning of the list)
    in order reach or exceed the sum n. If it is not possible to reach or
    exceed n, return -1.'''
```

As an example, `how_many([10, -3, 6, 2, 7, 4, 9], 14)` should return 4 because you must add the first 4 elements of the list to reach at least the sum 14.

Write this function. You must not use any for-loops, and you must not use `break` or `continue`, which were not even taught in this course. Solutions that do **will receive a mark of zero**. Hint: Use a while loop!

[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.]

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

```
__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 witt  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[k] --> value
    Return the value associated with the key k in D.
  k in d --> 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 then be returned).  Return an empty string at EOF.
list:
  L.append(x): Append x to the end of the list L.
  L.index(value) -> integer
    Return the lowest index of value in L.
  L.insert(index, x): Insert x at position index.
  L.sort(): Sorts the list in ascending order.
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.lower() -> string
    Return a copy of the string S converted to lowercase.
  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.
  S.upper() -> string
    Return a copy of the string S converted to uppercase.
```