CSC 108H1 F 2010 Test 2
Duration — 45 minutes
Aids allowed: none

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

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

Lecture Section: L0201 **Instructors: 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 8 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: _____/ 4
2: _____/ 6
3: _____/ 6
4: _____/ 8

TOTAL: _____/24

Question 1. [4 MARKS]

Beside each code fragment below, show the output that it would create. If it would generate an error say so, and give the reason why.

Part (a) [1 MARK]

```
L = [1, 15, 4]
for a in L:
    for b in L:
        print a + b
```

Part (b) [1 MARK]

```
d = {9: 3, -3: 17, 40: 20}
sum = 0
for x, y in d.items():
    sum = sum + y
    print sum
```

Part (c) [1 MARK]

```
s = "cabbages"
print s[2:6]
```

Part (d) [1 MARK]

```
s = "Really,truly?"
print s.find(",?")
```

Question 2. [6 MARKS]

Suppose we want to know how many times an `int` at a certain position in a `list` occurs in a row. For example, with the list `[55, 8, 14, 14, 14, 9, 0, 14, 14, 14, 14, 6]` and the position 2, we would determine that the `int` 14 occurs 3 times in a row starting at that position.

Write the function below, according to its docstring. You must not use a for-loop in this question or your solution will earn zero.

```
def repeats(L, i):
    '''L is a non-empty list ints and int i is a valid index into L. Return
    the number of times that the int at index i occurs in a row beginning at
    that index.'''
```

Question 3. [6 MARKS]

Write the function below, according to its docstring.

```
def frequencies(s):
    '''s is a string.  Return a dict where each key is a character from s
    and each value is the number of times that character occurs in s.'''
```

Question 4. [8 MARKS]

Suppose we have population data such as this:

```
Edmonton: 1034000
Los Angeles: 5123000
```

Write the function below, according to its docstring.

```
def total_population(filename):
    '''str filename is the name of a file.  Each line of the file gives a city
    and its population in the form
    CITY: POPULATION
    (There is a space character after the colon.)  Return the total population
    of the cities in the file.'''
```

[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__:
  len(x) -> integer
    Return the length of the list or string x.
  max(L) -> value
    Return the largest value in L.
  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.
float:
  float(x) -> float
    Convert a string or number to a float, if possible.
list:
  x in L --> boolean
    Return True if x is in L and False otherwise.
  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.
int:
  int(x) -> integer
    Convert a string or number to an integer, if possible.  A floating point argument
    will be truncated towards zero.
```

Continued on reverse

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

```
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.index(sub [,start [,end]]) -> int
    Like S.find() but raise ValueError when the substring is not found.
  S.lower() -> string
    Return a copy of the string S converted to lowercase.
  S.lstrip([chars]) -> string
    Return a copy of the string S with leading whitespace removed.
    If chars is given and not None, remove characters in chars instead.
  S.replace(old, new) --> string
    Return a copy of string S with all occurrences of the string old replaced with the string new.
  S.rstrip([chars]) -> string
    Return a copy of the string S with trailing whitespace removed.
    If chars is given and not None, remove characters in chars instead.
  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.
```