

Name: _____

Date: _____

Pledge: _____

Total: ____ / 75

Closed book: no textbook, no electronic devices, one sheet of paper with notes. Read each question carefully before answering, as there is **no partial credit for any response!** You may work out the solution to each question within the test itself, but **only your answers on this cover page will be graded.**

Question 1

['Why', 'is', 'raining', 'cow'] (5 points)

Question 2

is or 'is' (5 points)

Question 3

[7, 'stevens'] (5 points)

Question 4

- a) **3** (5 points)
- b) **(a) tail** (5 points)
- c) **10, 2** (5 points)
- d) **2** (5 points)

Question 5

- a) **8** (5 points)
- b) **(d) tree** (5 points)
- c) **annoy or 'annoy'** (5 points)
- d) **yonna or 'yonna'** (5 points)

Question 6

- a) **[]** (2 points)
- b) **len(lst[0])** (2 points)
- c) **[lst[0]]** (2 points)
- d) **1** (2 points)
- e) **min_length** (2 points)
- f) **lst[1:]** (2 points)
- g) **min_length** (2 points)
- h) **filter(lambda s: len(s) >= min_length, lst)** (6 points)

Question 1 (5 points)

Consider the following code:

```
L = ['Why', 'is', 'it', 'raining']
M = ['How', 'now', 'brown', 'cow']
N = L[ :2 ] + [ L[ -1 ] ] + M[ 3: ]
```

What is the value of N after these statements have executed?

Question 2 (5 points)

Consider the following code:

```
L = ['the', 'innovation', 'university', 'stevens', 'is']
M = range( 3, len(L) )
print(L[ M[ 1 ] ])
```

What is printed on the screen after these statements have executed?

Question 3 (5 points)

Consider the following code:

```
L = ['www', 'stevens', 'edu']
M = L + ['www', 'nyu', 'edu']
N = map(lambda s: [len(s), s], M)
```

What is the value of N[1] after these statements have executed?

Question 4 (20 points)

Consider the call to `mystery(58, 12)` for the function definition below.

```
def mystery(m, n):
    if n == 0:
        return m
    return mystery(n, m % n)
```

- a) Excluding the call to `mystery(58, 12)`, how many recursive calls are made before this function terminates?
- b) What type of recursion is found? Select the **best, most specific** answer.
 - (a) tail
 - (b) mutual
 - (c) linear
 - (d) tree
 - (e) nested
- c) What are the values of the parameters `m` and `n` in the second recursive call (that is, not including the initial call to `mystery(58, 12)`)?
- d) What is the final value returned by `mystery(58, 12)`?

Question 5 (20 points)

Consider the following code:

```
def confuse(s):
    if len(s) <= 1:
        return s
    x = len(s) // 2
    return confuse(s[:x]) + confuse(s[x:])

print(confuse('annoy'))
```

- a) Excluding the call to `confuse('annoy')`, how many recursive calls are made before this function terminates?
- b) What type of recursion is found? Select the **best, most specific** answer.
- (a) tail
 - (b) mutual
 - (c) linear
 - (d) tree
 - (e) nested
- c) What does `confuse('annoy')` return?
- d) Suppose the return statement is changed to:

```
return confuse(s[x:]) + confuse(s[:x])
```

What would `confuse('annoy')` return then?

Question 6 (20 points)

Implement the following function using recursion:

```
def keep_strings(lst, min_length):
    '''Assume lst is a list of strings.
    Returns a list of strings that have at least min_length characters.
    Examples:
    keep_strings([], 10) -> []
    keep_strings(
        ['wonderful', 'awesome', 'amazing', 'excellent', 'great'], 8) ->
        ['wonderful', 'excellent']
    ...
    if lst == []:
        return _____(a)_____
    if _____(b)_____ >= min_length:
        return _____(c)_____ + keep_strings(lst[____(d)____:], _____(e)_____)
    return keep_strings(_____(f)_____, _____(g)_____)
```

Implement the following function using **filter** and **lambda**.

```
def keep_strings_filter(lst, min_length):
    '''Assume lst is a list of strings.
    Returns a list of strings that have at least min_length characters.
    Examples:
    keep_strings_filter([], 10) -> []
    keep_strings_filter (
        ['wonderful', 'awesome', 'amazing', 'excellent', 'great'], 8) ->
        ['wonderful', 'excellent']
    ...
    return _____(h)_____
```

```

'''
CS 115 A, Spring 2017 - Test 1, Questions 7 and 8

Author: <your name here>
Pledge: <write pledge>
'''

from cs115 import filter
.....

' RULES: You can use Canvas to download this file and upload your solution.
' You can use Eclipse to edit and run your program. You should NOT look at
' other programs in Eclipse, you should NOT use any other programs, and you
' should NOT use any notes or books.
' According to the Honor Code, you should report any student who appears
' to be violating these rules.
.....

.....

' Question 7 (20 points)
' Implement these functions using recursion.
.....

def keep_integers(lst):
    '''Assume lst is a list of all different data types. There could be ints,
    floats, strings, booleans, nested lists, and more.
    Return a list of only the integers present in the original list. You do
    not have to worry about integers inside nested lists and can safely
    ignore them.
    You may use type(data) == int to determine if the data variable is an
    integer.
    This part is worth 20 points.'''
    if lst == []:
        return []
    if type(lst[0]) == int:
        return [lst[0]] + keep_integers(lst[1:])
    return keep_integers(lst[1:])

Rubric:
(3 points for correct if statement in base case,
2 points for correct return statement in base case,
3 points for correct if statement with call to type(),
2 points for returning [lst[0]],
5 points for each keep_integers(lst[1:]) )

.....

' Question 8 (10 points)
' Implement this function using the Python's built-in 'filter' and 'lambda'.
' DO NOT USE recursion.
.....

def keep_integers_filter(lst):
    '''Assume lst is a list of all different data types. There could be ints,
    floats, strings, booleans, nested lists, and more.
    Return a list of only the integers present in the original list. You do
    not have to worry about integers inside nested lists and can safely

```

```
ignore them.  
You may use type(data) == int to determine if the data variable is an  
integer.  
This part is worth 10 points.'  
return filter(lambda x: type(x) == int, lst)
```

Rubric:

(5 points for correct use of lambda,
5 points for correct use of filter)