

Name: _____ Section: _____ AndrewID: _____

15-112 Fall 2021 Quiz 3a

*** Up to 20 minutes. * No calculators, no notes, no books, no computers. * Show your work!**

*** No lists, list indexing, or recursion. * Fill in bubbles completely / erase cleanly.**

Multiple Choice:

MC1 [20pts total]: Select **ALL** of the following assert statements which **pass** (so, those that don't crash).

Note: s1 == 'dogs and cats' for the following lines

- a `assert('abcd' == "abcd")`
- b `assert('BEEP'.isupper())`
- c `assert((s1.find('s') == 3) and (s1.index('x') == -1))`
- d `assert(len('\nice') == 5)`
- e `assert(chr(ord('a') + 2) == 'c')`
- f `assert(isinstance(chr('A'),int))`
- g `assert(s1[5] == 'a')`
- h `assert(s1[1:7] == 'ogs and')`
- i `assert(string.ascii_lowercase('aBcD123') == 'abcd123')`
- j `assert(s1.isalnum())`

MC2 [14 pts]: Select **ONE** of the following will be printed by the following CT code:

```
def ct1(s):
    u = ''
    for i in range(len(s)):
        s = s[::-1]
        u += s[0] if (s[i].isupper()) else str(i)
    return u
print(ct1('ABcd'))
```

This code prints:

- a 'abcd'
- b '0Ad3'
- c 'ABCD'
- d '0Da3'
- e 'A12d'
- f 'D12a'

Short Answer 1: reverseStr(s) [20 pts]

Write the very short function reverseStr(s) which takes s (which is guaranteed to be a **string** of any length) and returns a string which is the reverse of s. **Do not use lists or recursion.**

```
assert(reverseStr('abcd123') == '321dcba')
assert(reverseStr(reverseStr('H0nk')) == reverseStr('kn0H'))
```

Short Answer 2: isPalindrome(i) [20 pts]

Write the very short function isPalindrome(i) which takes i (which is guaranteed to be a positive **integer**) and returns True if the number i is a palindrome, and False otherwise. A palindrome is a number which remains the same when its digits are reversed (ignoring leading zeros).

You may call your reverseStr(s) function if you wish, but you do not have to. Do not use lists or recursion.

```
assert(isPalindrome(12321)==True)
assert(isPalindrome(29892)==True)
assert(isPalindrome(1) == True)
assert(isPalindrome(12345) == False)
assert(isPalindrome(1010) == False)
```

Bonus/Optional: Code Tracing [+2pts]

Indicate what this prints. Place your answer (and nothing else) in the box.

```
def bonusCT(s):
    return s[s.index(chr(ord(s[int('a'>'Z'])))
              +s.find(str(s.count("")))]]
print(bonusCT("Bonus 112 Question?"))
```