

Name: _____ Section: ____ Andrew Id: _____

15-112 Fall 2016 Quiz 4a

* Up to 30 minutes. No calculators, no notes, no books, no computers. * Show your work!

* No recursion

1. **Code Tracing** [30 pts]: Indicate what this prints. Place your answer (and nothing else) in the box beside the code.

```
def ct1(L):
    result = [ ]
    M = [L[i]*10**i for i in range(len(L))]
    for val in M:
        result.extend([val, L.pop()])
    return result
L = [2,5,3]
M = ct1(L)
print(L, M)
```

```
def ct2(L):
    result = [ ]
    M = copy.copy(L)
    if (M == L): result.append(1)
    if (M is L): result.append(2)
    if (M[0] == L[0]): result.append(3)
    if (M[0] is L[0]): result.append(4)
    return result
print(ct2([5,7,6]))
```

```
def ct3(L):
    M = L
    L += [4]
    M = M + [5]
    print(L, M)
ct3(list(range(1)))
```

2. **Reasoning Over Code** [10 pts]:

Find an argument for the following function that makes it return True. Place your answer (and nothing else) in the box below the code:

```
def rc1(M):
    assert(isinstance(M, list) and (len(M) == 5))
    for i in range(-1, 3):
        assert(M[i] == M[i-1] + i)
    return (sum(M) == 15)
```

M =

3. **Fill in the Blanks** [20 pts]

Fill in the 5 blanks with the missing code from the Locker Problem worked example that we covered in lecture.

```
def lockerProblem(lockers):
    isOpen = [ False ] * _____
    students = lockers
    for student in range(1,students+1):
        for locker in range(student, lockers+1, _____):

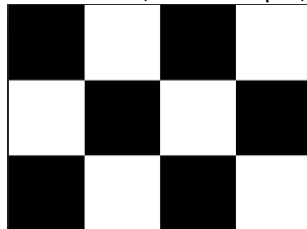
            isOpen[locker] = _____
    openLockers = [ ]
    for locker in range(1, lockers+1):

        if _____:
            openLockers.append(locker)

    return _____
```

4. **Free Response: drawCheckerboard(canvas, width, height, rows, cols)** [40 pts]

Write your solution to this problem on the empty back side of Check5 (a separate handout) . With this in mind, write the function drawCheckerboard that takes a canvas and 4 positive integers -- width, height, rows, cols – and draws a rows x cols black-and-white checkerboard into the width x height canvas. The left-top corner cell must be black, and then cells alternate from there. Here, for example, is a 3x4 checkerboard:



Your checkerboard must entirely fill the canvas, and the cells must be the same size as each other, so they might not be square.

5. **Bonus/Optional: Code Tracing** [5 pts; 2.5 pts each] What will these print? Place your answer in the boxes.

```
def bonusCt1(L):
    try:
        while True:
            L.append(L[-2] + L.pop() + L.pop())
    except: return L
print(bonusCt1(list(range(13)))[-1])
```

```
def bonusCt2(L):
    def f(L): return [(-1)**(i%2>0)*L[i] for i in range(len(L))]
    L=f(L); return sum(L[L[1]:L[-2]:-2]) if (len(L)>4) else 42
print(bonusCt2(list(range(11))))
```