

2. **Fill in the Blank: selectionSort(a)** [25 pts]

Fill in the 5 blanks with the missing code from the example in our notes on sorting

```
def selectionSort(a):  
    n = len(a)  
    for startIndex in range(n):  
        minIndex = _____ #1  
        for i in range(_____, n): #2  
            if (_____): #3  
                minIndex = i  
        swap(a, _____, _____) #4+5
```

[Continue to next page for Quiz6]

Name: _____ Section: ____ Andrew Id: _____

15-112 Fall 2016 Quiz 6a

1. **Code Tracing** [10 pts]: Indicate what this prints. Place your answer (and nothing else) in the box below the code.

```
def ct1(L):
    a = L
    b = copy.copy(L)
    c = copy.deepcopy(L)
    a[0] = b[1]
    b[1][1] = c[0]
    c[1].append(b[1])
    a[0][0] += (b[1].pop())[0]
    return (a,b,c)
# Be careful to get the brackets
# and commas right!
for val in ct1([[1],[2,5]]):
    print(val) # prints 3 lines
```

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(n):
    assert(isinstance(n, int) and (n > 0))
    (d, total) = (0, 0)
    while (total < 100):
        (d, total) = (d+1, 0)
        a = [d] * d
        b = [a] * d # note: bad way to make a 2d list!
        for i in range(d):
            for j in range(d):
                total += b[i][j]
    return (n == d)
```

n =

3. **Multiple Choice** [15 pts]

Circle all the correct answers. Some questions may have more than one correct answer.
All these questions refer to the run function in our animation framework.

- a. What does root.mainloop do?
 - i. Displays the window
 - ii. Listens for mouse and keyboard events
 - iii. Listens for timer events
 - iv. None of the above

- b. What do the event wrapper functions (keyPressedWrapper, etc) do?
 - i. Call redrawAll
 - ii. Call init
 - iii. Call run
 - iv. None of the above

- c. What does canvas.after do?
 - i. Redraws the canvas after a delay
 - ii. Exits the app after a delay
 - iii. Calls a function (provided as a parameter) after a delay
 - iv. None of the above

Name: _____ Section: ____ Andrew Id: _____

4. **Free Response: dotDot** [40 pts]

Using our animation framework and assuming `run()` is already written, write `dotDot`, an app that works as such:

- a. When the app launches, a red dot of radius 20 starts at the top, centered horizontally and sweeps downward. Each time it exits the bottom, it re-enters at the top. Also, a blue dot of radius 20 starts at the right, at the bottom of the screen (not the vertical center), and sweeps leftward. Each time it exits the left, it re-enters at the right.
- b. Also, a score is always displayed in the left-top corner. Initially, the score is 0.
- c. Each time the dots collide, the score increases by 1 and the red dot resets to its initial position (but the blue dot does not reset and just keeps moving).
- d. Pressing 'p' pauses the red dot, but the blue dot keeps moving. Each time you press 'p', pause is toggled from on to off or from off to on, always only affecting the red dot.
- e. A mousepress anywhere resets the app to its initial state.

5. **Free Response: isFoiled(L)** [25 pts]

Write the non-destructive function `isFoiled(L)` that takes a rectangular 2d list of ints `L` and returns `True` if `L` is foiled (a coined term) and `False` otherwise, where a list is foiled if every row in `L` is equal (`==`) to some column in `L`, where rows are read left-to-right and columns are read top-to-bottom. For example, consider this list:

```
[ [ 1, 1, 2 ],  
  [ 2, 1, 1 ],  
  [ 1, 2, 1 ] ]
```

Row0 is [1,1,2] which equals col1.

Row1 is [2,1,1] which equals col2.

Row2 is [1,2,1] which equals col0.

So this list is foiled.

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

```
def bonusCt1(L):  
    for i in range(len(L)): L[i] = [L[i]]*L[i]  
    return sum([sum(val) for val in L])  
print(bonusCt1(list(range(5,11))))
```

```
def bonusCt2(k, result=0):  
    for m in range(2**k):  
        while (m > 0): (m, result) = (m//2, result+m%2)  
    return result  
print(bonusCt2(5))
```