Part 1 Time limit: 15 mins

Multiple Choice [17 pts]

1. Which of the following creates a tuple containing the number 5? [3 points]

A. `tuple(5)`
B. `(5)`
C. `(5,)`
D. `tuple(list(5))`

Note: The next 4 questions are based on the animation problem from the last quiz.

Say `app.dots` is a list of (cx, cy) tuples representing the centers of different dots. If the left or right arrow key is held, all the dots in `app.dots` should move 5 pixels in the direction of the arrow key. If both the left and right arrow keys are held at the same time, the dots should not move. Below is some partially written code for `onKeyPress` to implement this functionality.

```
def onKeyHold(app, keys):
    dx = 0 # Don't miss this line!
    if 'right' in keys:
        _________ # BLANK 1
    if 'left' in keys:
        _________ # BLANK 2

    for i in range(len(app.dots)):
        cx, cy = _________ # BLANK 3
        app.dots[i] = _________ # BLANK 4
```

2. What code should go in the first blank? [2 points]

A. `dx = 5`
B. `dx = -5`
C. `dx += 5`
D. `dx += -5`

3. What code should go in the second blank? [2 points]

A. `dx = 5`
B. `dx = -5`
C. `dx += 5`
D. `dx += -5`

4. What code should go in the third blank? [2 points]

A. `app.dots`
B. `app.dots[i]`
C. `i`
D. `dot`

5. What code should go in the fourth blank? [2 points]

A. `(cx, cy)`
B. `(dx, dy)`
C. `(cx + dx, cy)`
D. `(cx + dx, cy + dy)`

The next 2 questions are based on the `takeStep` function below that we wrote in lecture for the Snake Game Case Study. As a reminder, `app.snake` is a list of `(row, col)` tuples representing the row and column of each dot in the snake.

```
def takeStep(app):
    headRow, headCol = app.snake[0]
    drow, dcol = app.direction
    newHeadRow, newHeadCol = headRow + drow, headCol + dcol
    if (isOnBoard(app, newHeadRow, newHeadCol) and
        (newHeadRow, newHeadCol) not in app.snake):
        _________ # BLANK 1
        if (newHeadRow, newHeadCol) == app.foodPosition:
            placeFood(app)
        else:
            _________ # BLANK 2
    else:
        app.gameOver = True
```

6. What code should go in the first blank? [3 points]

A. `app.snake.insert(0, (newHeadRow, newHeadCol))`
B. `app.snake.append(0, (newHeadRow, newHeadCol))`
C. `app.snake.append((newHeadRow, newHeadCol))`
D. `app.snake += (newHeadRow, newHeadCol)`

7. What code should go in the second blank? [3 points]

A. `app.snake.pop(0)`
B. `app.snake.pop()`
C. `app.snake.remove(0)`
D. `app.snake.remove((headRow, headCol))`

Code Tracing [18 pts]

What does the following code print?

```
import copy

def ct(L):
    A, B = copy.copy(L), copy.deepcopy(L)
    A[0][1] *= 2
    B[0][1] *= 3
    print(L == A, A == B)
    print(L is A, A is B)
    x = A.pop(1)
    x[0] = 10
    print(x)
    A.extend(x)
    B[1].sort()
    print(A)
    print(B)

L = [[5, 1],
     [4, 3]]
ct(L)
print(L)
```

Answer:

Reasoning Over Code [15 pts]

Find an argument for the function `roc(L)` that makes it return `True`.

```python
def roc(L):
    M = []
    for i in range(len(L)):
        M.insert(0, [])
        for j in range(len(L[i]) - 1, -1, -1):
            M[0].append(L[i][j])
    return M == [ [17, 2, 16], [6], [8, 12] ]
```

Answer: