

Multiple Choice [15 pts]

For each of the following, indicate whether they are part of the model, view, or control. [1 pt each]

1. `app.width`

- A. Model
- B. View
- C. Control

2. A helper function that draws circles

- A. Model
- B. View
- C. Control

3. `app.stepsPerSecond`

- A. Model
- B. View
- C. Control

4. `redrawAll`

- A. Model
- B. View
- C. Control

5. `onAppStart`

- A. Model
- B. View
- C. Control

6. If `points = [100, 100, 50, 200, 300, 300, 250, 50]`, which of the following draws a polygon with vertices at those points? [2 points]

- A. `drawPolygon(points)`
- B. `drawPolygon(*points)`
- C. `drawPolygon([points])`
- D. `drawPolygon(*[points])`

7. Which of the following creates a tuple containing the number 5? [2 points]

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

8. Which of the following prints the letters 'a', 'b', 'c', and 'd' each on their own line? [2 points]

A.

```
L = [('a', 'b'), ('c', 'd')]
for i in range(len(L)):
    print(L[i])
```

B.

```
L = [('a', 'b'), ('c', 'd')]
for i in range(len(L)):
    print(i)
    print(L[i])
```

C.

```
L = [('a', 'b'), ('c', 'd')]
for v in L:
    print(v)
```

D.

```
L = [('a', 'b'), ('c', 'd')]
for v1, v2 in L:
    print(v1)
    print(v2)
```

9. What does the following code print? [2 points]

```
L = ['abcd', 'ef', 'ghi', 'j']
print(L[2:])
```

- A. ['cd']
- B. 'cd'
- C. ['ghi', 'j']
- D. 'ghi', 'j'

10. What will happen if you run the following code? [2 points]

```
L = [1, 2, 3, 4]
for i in range(len(L)):
    print(L.pop(i))
```

A. It will print:

```
1
2
3
4
```

B. It will print **None**

C. It will print **[]**

D. It will crash

Code Tracing [30 pts]

CT1 [15 pts]

What does the following code print?

```
def ct(L):  
    M = [ L[i] % 10 for i in range(len(L))]  
    print(M)  
    N = [ ]  
    for v in M:  
        if v in L:  
            N.append(v)  
    return N + [ N.count(5) ]  
  
print(ct([15, 112, 5, 3]))
```

Answer:

CT2 [15 pts]

What does the following code print?

```
import copy
def ct(L):
    L.extend([1,2])
    L = copy.copy(L)
    L[1] += 2
    print(L.pop(len(L) // 2))
    print(L)
    M = L + ['a']
    print(M.remove(1))
    print(M)
    print(L)

L = [1, 2, 3]
ct(L)
print(L)
```

Answer:

Reasoning Over Code [15 pts]

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

```
def roc(L):
    if type(L) != list:
        return False
    M = []
    for v in L:
        if v % 2 == 0:
            M.append(v)
    for i in range(1, len(M)):
        if M[i] != 2 * M[i-1]:
            return False
    return ((len(M) == 3) and
            (len(L) == 2 * len(M)) and
            (L[0] == 3) and
            (0 not in L))
```

Answer: