### What gets printed, Round 1

<table>
<thead>
<tr>
<th>a = 0</th>
<th>a = 0</th>
<th>a = 0</th>
<th>a = 0</th>
<th>a = 0</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>print(a)</code></td>
<td>a = a + 1</td>
<td><code>if a == 0:</code></td>
<td>a = a + 1</td>
<td><code>if a == 0:</code></td>
</tr>
<tr>
<td></td>
<td><code>print(a)</code></td>
<td>if a == 0:</td>
<td><code>a = a + 1</code></td>
<td>if a == 0:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>a = a + 1</td>
<td><code>print(a)</code></td>
<td>a = a + 1</td>
</tr>
<tr>
<td></td>
<td></td>
<td><code>print(a)</code></td>
<td></td>
<td><code>print(a)</code></td>
</tr>
</tbody>
</table>

(Let's look at these one by one.)
What gets printed, Round 1, Solutions

```python
a = 0
print(a)  # 0

a = a + 1
print(a)  # 1

if a == 0:
    a = a + 1
print(a)  # 1

if a == 1:
    a = a + 1
print(a)  # 0

if a == 0:
    a = a + 1
    print(a)  # 2
```

0 1 1 0 2
What gets printed? (Question)

```python
a = 0
if a == 0:
a = a + 1
if a == 0:
a = a + 2
a = a + 1
print(a)
```

A: 0
B: 1
C: 2
D: 3
E: I do not know
What gets printed? (Solution)

```
a = 0
if a == 0:
    a = a + 1
if a == 0:
    a = a + 2
a = a + 1
print(a)
```

A: 0
B: 1
C: 2  CORRECT
D: 3
E: I do not know
What gets printed, Round 2

\[
\begin{align*}
\text{Let's look at these one by one.}
\end{align*}
\]
What gets printed, Round 2

\[
\begin{align*}
\text{a} &= 0 \\
\text{if } a == 0: & \quad \text{if } a == 1: \\
& \quad \quad a = a + 1 \\
& \quad \quad a = a + 1 \\
\text{else:} & \quad \quad a = a + 2 \\
& \quad \quad a = a + 2 \\
\text{print}(a) & \quad \quad \text{print}(a) \\
\end{align*}
\]

1

\[
\begin{align*}
\text{a} &= 0 \\
\text{if } a == 1: & \quad \text{if } a == 1: \\
& \quad \quad a = a + 1 \\
& \quad \quad a = a + 1 \\
\text{else:} & \quad \quad a = a + 2 \\
& \quad \quad a = a + 2 \\
\text{print}(a) & \quad \quad \text{print}(a) \\
\end{align*}
\]

2

\[
\begin{align*}
\text{a} &= 0 \\
\text{if } a == 1: & \quad \text{if } a == 1: \\
& \quad \quad a = a + 1 \\
& \quad \quad a = a + 1 \\
\text{else:} & \quad \quad a = a + 2 \\
& \quad \quad a = a + 2 \\
\text{a} &= a + 1 \\
\text{a} &= a + 1 \\
\text{print}(a) & \quad \quad \text{print}(a) \\
\end{align*}
\]

3

\[
\begin{align*}
\text{a} &= 0 \\
\text{if } a == 1: & \quad \text{if } a == 1: \\
& \quad \quad a = a + 1 \\
& \quad \quad a = a + 1 \\
\text{else:} & \quad \quad a = a + 2 \\
& \quad \quad a = a + 2 \\
\text{a} &= a + 1 \\
\text{a} &= a + 1 \\
\text{print}(a) & \quad \quad \text{print}(a) \\
\end{align*}
\]

3
What does the call frame look like next? (Q)

def max(x, y):
    if x > y:
        return x
    return y

max(0, 3)

Current call frame:

A:

max

<table>
<thead>
<tr>
<th>x</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>y</td>
<td>3</td>
</tr>
</tbody>
</table>

B:

max

<table>
<thead>
<tr>
<th>x</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>y</td>
<td>3</td>
</tr>
</tbody>
</table>

RETURN 0

C:

max

<table>
<thead>
<tr>
<th>x</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>y</td>
<td>3</td>
</tr>
</tbody>
</table>

RETURN 3

D:

max

<table>
<thead>
<tr>
<th>x</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>y</td>
<td>3</td>
</tr>
</tbody>
</table>
What does the call frame look like next? (A)

```python
def max(x, y):
    if x > y:
        return x
    return y
```

Current call frame:

<table>
<thead>
<tr>
<th></th>
<th>x</th>
<th>y</th>
</tr>
</thead>
<tbody>
<tr>
<td>max</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>x</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>y</td>
<td>3</td>
<td></td>
</tr>
</tbody>
</table>

**A:**

```
max(0, 3)
```

**B:**

```
max(1, 3)
```

**C:**

```
max(1, 3)
```

**D:**

```
max(1, 3)
```

✓
def max(x, y):
    if x > y:
        return x
    return y

max(0, 3):
Call Frame Explanation (2)

```python
def max(x, y):
    if x > y:
        return x
    return y
```

```
max(0, 3):
```

[Diagram showing the execution of `max(0, 3)` with `x` as 0, `y` as 3, and skipping line 2]
def max(x, y):
    if x > y:
        return x
    return y

max(0, 3):
What gets printed, Round 3

```python
a = 0
def plus(a, b):
    return a + b

if a == 0:
    b = 0
    print(b)

a = 1
if a == 0:
    b = 0
    print(b)
```
What gets printed, Round 3, Solution

```python
a = 0
if a == 0:
    b = 0
print(b)

a = 1
if a == 0:
    b = 0
print(b)
```

0

Error!
def max(x, y):
    """Returns: max of x, y""
    # note: code has a bug!
    # check if x is larger
    if x > y:
        bigger = x
    return bigger

maximum = max(3, 0)

Value of maximum?

A: 3
B: 0
C: Error!
D: I do not know
Control Flow and Variables (A1)

```python
def max(x, y):
    """Returns: max of x, y""
    # note: code has a bug!
    # check if x is larger
    if x > y:
        bigger = x
    return bigger

maximum = max(3, 0)
```

Value of `maximum`?

- A: 3  CORRECT
- B: 0
- C: Error!
- D: I do not know

- Local variables last until
  - They are deleted or
  - End of the function
- Even if defined inside if
Control Flow and Variables (Q2)

```python
def max(x,y):
    """Returns: max of x, y"""
    # note: code has a bug!
    # check if x is larger
    if x > y:
        bigger = x
    return bigger

maximum = max(0,3)
```

Value of `maximum`?

A: 3  
B: 0  
C: Error!  
D: I do not know
Control Flow and Variables (A2)

```python
def max(x,y):
    """Returns: max of x, y""
    # note: code has a bug!
    # check if x is larger
    if x > y:
        bigger = x
    return bigger

maximum = max(0,3)
```

Value of `maximum`?

- A: 3
- B: 0
- C: Error! CORRECT
- D: I do not know

- Variable existence depends on flow
- Generally terrible idea to refer to variables defined inside an `if` clause
If-Elif-Else (Question)

a = 2

if a == 2:
a = 3
elif a == 3:
a = 4
print(a)

What gets printed?

A: 2
B: 3
C: 4
D: I do not know
If - Elif - Else (Answer)

a = 2

if a == 2:
    a = 3
elif a == 3:
    a = 4
print(a)

What gets printed?

A: 2
B: 3  CORRECT
C: 4
D: I do not know
What gets printed, Round 4

```python
a = 2
if a == 2:
    a = 3
elif a == 3:
    a = 4
print(a)
```
What gets printed, Round 4

```python
a = 2
if a == 2:
    a = 3
elif a == 3:
    a = 4
print(a)
```

```python
a = 2
if a == 2:
    a = 3
if a == 3:
    a = 4
print(a)
```

3

4