Set 2: Recursion with Statements *Before* Recursive Calls (Winding Phase)

Focus: Understanding top-down processing (pre-order behavior)

MCQ 1: Print Before Recursion

```
function display(n):
    if n == 0:
        return
    print(n)
    display(n - 1)

display(3)
```

Output?
A) `3 2 1`
B) `1 2 3`
C) `3 2`
D) `2 1 3`

MCQ 2: Countdown with Two Recursive Calls

```
function call(n):
    if n == 0:
        return
    print(n)
    call(n - 1)
    call(n - 1)

call(2)
```

Output?
A) `2 1 1`
B) `2 2 1`
C) `2 1 2`
D) `1 2 1`

MCQ 3: Array Forward Print

```
arr = [10, 20, 30]
function printArr(i):
    if i == length(arr):
        return
    print(arr[i])
    printArr(i + 1)

printArr(0)
```

Output?
A) `10 20 30`
B) `30 20 10`

C) `10 30 20`
D) `20 10 30`

MCQ 4: Fibonacci Pre-Call Printing

```
function fib(n):
    if n <= 1:
        return n
    print(n)
    return fib(n - 1) + fib(n - 2)

fib(3)
```

Output?
A) `3 2 1`
B) `3 2`
C) `2 3`
D) `3 3`

MCQ 5: Even Index Print

```
function evenPrint(n):
    if n > 5:
        return
    if n % 2 == 0:
        print(n)
    evenPrint(n + 1)

evenPrint(0)
```

Output?
A) `0 2 4`
B) `0 2 4 6`
C) `2 4 6`
D) `0 2 4 6 8`