Function Parameters and Argument Behavior

---

1. Default Arguments

Q1. What will be the output of the following function call, and why?

```
def func(a, b=6, c=8):
    print(a, b, c)

func(1, 2)
```

---

2. Keyword Arguments with Defaults

Q2. Predict the output and explain:

```
def func(a, b, c=5):
    print(a, b, c)

func(1, c=3, b=2)
```

---

3. Positional Variable-Length Arguments

Q3. Explain the behavior and output:

```
def func(a, *pargs):
    print(a, pargs)

func(1, 2, 3)
```

---

4. Keyword Variable-Length Arguments

Q4. What will this code print, and why?

```
def func(a, **kargs):
    print(a, kargs)

func(a=1, c=3, b=2)
```

---

5. Mixing Positional and Unpacked Tuple Arguments

Q5. Explain the result of this call:

```
def func(a, b, c=8, d=5):
    print(a, b, c, d)

func(1, *(5, 6))
```

---

6. Mutable vs Immutable Behavior

Q6. Predict the output of the variables `l`, `m`, and `n` after this code runs:

```
def func(a, b, c):
    a = 2
    b[0] = 'x'
    c['a'] = 'y'

l = 1
m = [1]
n = {'a': 0}

func(l, m, n)

print(l, m, n)
```

Bonus Challenge

Write a function that accepts:

- one required argument,
- a variable number of positional arguments,
- a variable number of keyword arguments,

Then print all three components. Call it with:

```
my_func(10, 20, 30, x=5, y=15)
```

Would you like me to combine Assignments 23 and 24 into a formatted PDF or exportable Word document?