**1. What is the result of the code, and why?**

**>>> def func(a, b=6, c=8):**

**print(a, b, c)**

**>>> func(1, 2)**

Ans: The output will be 1 2 8. In this function call, the value of a is provided as 1, the value of b is provided as 2, and the default value of c remains 8.

**2. What is the result of this code, and why?**

**>>> def func(a, b, c=5):**

**print(a, b, c)**

**>>> func(1, c=3, b=2)**

Ans: The output will be 1 2 3. In this function call, the values of a, b, and c are provided explicitly as 1, 2, and 3 respectively. The default value of c is overridden by the provided value 3.

**3. How about this code: what is its result, and why?**

**>>> def func(a, \*pargs):**

**print(a, pargs)**

**>>> func(1, 2, 3)**

Ans: The output will be 1 (2, 3). The first argument a is provided with the value 1. The \*pargs parameter collects the additional positional arguments into a tuple, so pargs becomes (2, 3).

**4. What does this code print, and why?**

**>>> def func(a, \*\*kargs):**

**print(a, kargs)**

**>>> func(a=1, c=3, b=2)**

Ans: The output will be 1 {'c': 3, 'b': 2}. The \*\*kargs parameter collects the additional keyword arguments into a dictionary, so kargs becomes {'c': 3, 'b': 2}.

**5. What gets printed by this, and explain?**

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

**>>> func(1, \*(5, 6))**

Ans: The output will be 1 5 6 5. The first argument a is provided with the value 1, the second argument b is unpacked using \*(5, 6) which is equivalent to 5, 6, and the default values of c and d are 8 and 5 respectively.

**6. what is the result of this, and explain?**

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

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

**>>> func(l, m, n)**

**>>> l, m, n**

Ans: The output will be 1 ['x'] {'a': 'y'}. When l is passed to the function, it's not mutable, so modifying it inside the function doesn't affect the original value. However, m and n are mutable (a list and a dictionary respectively), so modifications inside the function affect the original objects.