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

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

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

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

Answer 1: The code prints: 1 2 8. In the function call func(1, 2), a takes the value 1, b takes the value 2, and the default value of c (8) is used.

**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)**

Answer 2: The code prints: 1 2 3. In the function call func(1, c=3, b=2), a takes the value 1, b takes the value 2 (from the keyword argument), and c takes the value 3 (from the keyword argument).

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

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

**print(a, pargs)**

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

Answer 3: The code prints: 1 (2, 3). The \*pargs collects additional positional arguments into a tuple. In the function call func(1, 2, 3), a takes the value 1, and pargs becomes the tuple (2, 3).

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

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

**print(a, kargs)**

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

Answer 4: The code prints: 1 {'c': 3, 'b': 2}. The \*\*kargs collects additional keyword arguments into a dictionary. In the function call func(a=1, c=3, b=2), a takes the value 1, and kargs becomes the dictionary {'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))**

Answer 5: The code prints: 1 5 6 8. The \*(5, 6) unpacks the tuple into the function's parameters. So, a takes the value 1, b takes the value 5, and c takes the value 6. The default value of d (8) is used.

**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**

Answer 6: The result is: 1 ['x'] {'a': 'y'}. The function modifies the values of a, the first element of list m, and the value associated with the key 'a' in dictionary n. After the function call, l remains 1, m becomes ['x'], and n becomes {'a': 'y'}.