**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 result of the code is "1 2 8". When the function is called with arguments 1 and 2, the value of a is assigned as 1, the value of b is assigned as 2 due to the positional argument, and the default value of c is used as 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 result of the code is "1 2 3". When the function is called with arguments 1, b=2 and c=3, the value of a is assigned as 1, the value of b is assigned as 2 and the value of c is assigned as 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 result of the code is "1 (2, 3)". When the function is called with arguments 1, 2 and 3, the value of a is assigned as 1 and the remaining arguments 2 and 3 are collected into a tuple and assigned to pargs.

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

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

**print(a, kargs)**

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

Ans-The code prints "1 {'c': 3, 'b': 2}". When the function is called with keyword arguments a=1, b=2 and c=3, the value of a is assigned as 1 and the remaining arguments b and c are collected into a dictionary and assigned to kargs.

**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 code prints "1 5 6 5". When the function is called with arguments 1 and the tuple (5, 6), the value of a is assigned as 1, the first value of the tuple (5,) is assigned to b and the remaining value (6,) is assigned to c.

**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 code will print: 2 ['x'] {'a': 'y'}. In this function call, the variable 'l' is passed as an argument and assigned to the parameter 'a'. The variable 'm' is also passed and assigned to the parameter 'b', which is a list. Inside the function, the value of 'a' is changed to '2'. The first element of the list 'm' is modified to 'x' because lists are mutable objects and changes made inside the function are reflected outside as well. The variable 'n' is also passed and assigned to the parameter 'c', which is a dictionary. The dictionary 'n' is modified by assigning the value 'y' to the key 'a'. Finally, outside the function, the values of 'l', 'm', and 'n' are printed, resulting in '2', "['x']", and "{'a': 'y'}" respectively.