Assignment 23

# Q1

The code will output: 1 2 8

Explanation:

The function func is defined with three parameters: a, b, and c.

The parameter b has a default value of 6, and the parameter c has a default value of 8.

When the function is called as func(1, 2), the argument 1 is assigned to a and the argument 2 is assigned to b.

Since the argument for c is not provided, the default value 8 is used.

Therefore, when the function is executed, it prints the values of a, b, and c as 1 2 8.

# Q2

The code will output: 1 2 3

Explanation:

The function func is defined with three parameters: a, b, and c.

When the function is called as func(1, c=3, b=2), the argument 1 is assigned to a, 2 is assigned to b, and 3 is assigned to c.

The arguments b and c are explicitly passed by keyword, so their positions can be interchanged.

Therefore, when the function is executed, it prints the values of a, b, and c as 1 2 3.

# Q3

The code will output: 1 (2, 3)

Explanation:

The function func is defined with a parameter a and \*pargs which allows for variable-length arguments (also known as varargs).

When the function is called as func(1, 2, 3), the value 1 is assigned to a, and the remaining arguments 2 and 3 are collected into pargs as a tuple (2, 3).

When the function is executed, it prints the value of a as 1 and the tuple pargs as (2, 3). The \* in front of pargs unpacks the tuple, so it is displayed without parentheses.

# Q4

The code will output: 1 {'c': 3, 'b': 2}

Explanation:

The function func is defined with a parameter a and \*\*kargs which allows for variable-length keyword arguments (also known as kwargs).

When the function is called as func(a=1, c=3, b=2), the value 1 is assigned to a, and the remaining keyword arguments c=3 and b=2 are collected into kargs as a dictionary {'c': 3, 'b': 2}.

When the function is executed, it prints the value of a as 1 and the dictionary kargs as {'c': 3, 'b': 2}. The dictionary represents the keyword arguments passed to the function, where the keys are the parameter names and the values are the corresponding values.

# Q5

The code will output: 1 5 6 5

Explanation:

The function func is defined with four parameters: a, b, c, and d.

When the function is called as func(1, \*(5, 6)), the value 1 is assigned to a, and the expression \*(5, 6) is used to unpack the tuple (5, 6). This means that 5 is assigned to b, and 6 is assigned to c.

Since the parameter d is not provided when calling the function, it takes its default value of 5.

When the function is executed, it prints the values of a, b, c, and d, resulting in 1 5 6 5.

# Q6

The code will result in the modification of the mutable objects m and n.

Explanation:

The function func takes three parameters: a, b, and c.

Inside the function, the value of a is reassigned to 2.

The first element of list b is changed to 'x' by assigning 'x' to b[0].

The value associated with key 'a' in dictionary c is modified to 'y' by assigning 'y' to c['a'].

The variables l, m, and n are assigned the values 1, [1], and {'a': 0} respectively.

When the function is called as func(l, m, n), the values of l, m, and n are passed as arguments to the function.

As a result of calling the function, the variable m is modified, changing the first element from 1 to 'x'. The variable n is also modified, changing the value associated with the key 'a' from 0 to 'y'.

However, the variable l remains unchanged because it is an immutable integer.

Therefore, after calling the function, the values of l, m, and n will be 1, ['x'], and {'a': 'y'} respectively.