Assignment 23

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

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

print(a, b, c)

>>> func(1, 2)

Output will be 1 2 8 because the function uses the default value of c.

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)

Output will be 1 2 3 because the function will use default values only when a value for an argument is not provided.

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

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

print(a, pargs)

>>> func(1, 2, 3)

Output will be 1 (2, 3)because argument a will be 1 whereas pargs is used when we are not sure about the no of arguments to be passed to a function. all the values under this argument will be stored in a tuple.

4. What does this code print, and why?

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

print(a, kargs)

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

Output will be 1 {'c': 3, 'b': 2} because \*\*kargs stands for variable length keyword arguments. This format is used when we want pass key value pairs as input to a function. All these key value pairs will be stored in a dictionary.

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))

Output will be 1 5 6 5 because the function will expand the \*(5,6) and consider the value of b as 5 and value of c as 6. since the default value of d is provided in function declaration d value will be 5.

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

Output will be (1, ['x'], {'a': 'y'}) because When l,m,n are provided as inputs to the function. its modifies the values of l,m,n and sets the value of l=1 ,m=['x'] and n={'a':'y'}