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

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

print (a, b, c)

>>> func(1, 2)

1, 2, 8

The function gets 2 parameters, and they will replace with a and b and for c function will use default value

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)

1, 2, 3

The function gets 3 parameters, 1 will assign to a , 3 , 2 will assign to c and b respectively as input parameters of function and these value will replace instead of default value.

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

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

print(a, pargs)

>>> func(1, 2, 3)

1 2 3

\*pargs use when we are not sure about number of variables, then 1 will assign to a and other parameter will pass into the function through \*pargs.

4. What does this code print, and why?

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

print(a, kargs)

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

1 {‘c’ : 3 , ‘b’ : 2}

\*\*kargs use for passing variable argument into function, then python will make a dictionary and save variables inside it.

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))

1 5 6 5

The parameters pass to function ordered, then a = 1 , b = 5 , c = 6 and d with get default value which is d = 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

(1, ['x'], {'a': 'y'})

In python variables pass by reference and reference pass by value, then when we pass a member of list or dictionary the reference will pass to function and when we change it inside the function it will affect outside too.