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:1 2 8

While calling the func we are passing values 1 and 2 which are stored in a and b and c already contains 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: 1 2 3

While calling the func we are passing the values for a,b and c so they get printed accordingly

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

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

print(a, pargs)

>>> func(1, 2, 3)

Ans: 1 (2,3)

1 is assigned to a and 2,3 is taken as positional argument

4. What does this code print, and why?

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

print(a, kargs)

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

Ans: 1 {‘c’:3,’b’:2}

1 is taken as positional argument and c and b as keyword argument

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: 1 5 6 5

\*(5,6) is taken as positional argument for b and 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: 1 [‘x’] {‘a’:’y’}

Value of l is not changing as we cant assign new value to an int but we are trying to change index value of m and n so it changes basically we are changing the values at memory location of a list and dict