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

Here a and b is passed as 1 and 2 and default value of c is printed.

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

Here 1 is passed as a and b and c as passed as 2 and 3.

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

Here, 1 is passed as a and 2 and 3 is passed as \*pargs.

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}

a is passed as 1 and rest of the parameters as treated as dictionary for \*\*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. 1 5 6 5

Here 1 is passed as a and 5, 6 as passed as b and c and default value od d is passed.

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. It will throw an error because l, m and n are defined inside a function.