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, default value of parameters are being set in function definition, so if no value is provided for that parameter then default value will be used.

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, function is being called using named parameter. So value is being mapped to that parameter only.

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), one value is mapped with a parameter and other get mapped with args parameter in form of tuple.

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}, one value is mapped with a parameter and other get mapped with args parameter in dictionary because of named parameters get passed.

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, 1 is being mapped to a parameter, 5,6 with b and c, and since d value is not passed so it used default value.

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'}), data structures get initialized first and being passed it function so value get stored in data structures.