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

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

print(a, b, c)

>>> func(1, 2)

**Answer:**

**1,2,8**

**1 will be assigned to a variable, 2 will be assigned to variable b and it will override the value 6.**

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)

**Answer:**

**1,2,3**

**since the values are assigned to the variable , paramter position doesn't matter here.**

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

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

print(a, pargs)

>>> func(1, 2, 3)

**Answer:**

**1 (2,3)**

**1 is assigned to the variable and rest of the inputs are assigned to \*pargs take input as tuples.**

4. What does this code print, and why?

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

print(a, kargs)

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

**Answer:**

**1 {'c': 3, 'b': 2}**

**1 is assigned to the variable a and rest of the inputs are assigned to \*\*kargs and it will takes inputs as 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))

**Answer:**

**1 5 6 5**

**1 is assigned to variable a, tuple values are assigned to respective variable 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

**Answer:**

**Since l,m,n are not assigned any values. it throws an error**