**Assignment 23**

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: The output will be ‘1 2 8’ because although we didn’t give three arguments to the function but two of the arguments of the function are already specified. Therefore, passing 1,2 will override the a and b values, hence the value of a =1, b=2, c=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: The output will be ‘1 2 3’ because we are specifying a=1, b=2 and c=3.

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

def func(a, \*pargs):

print(a, pargs)

func(1, 2, 3)

Ans: The output will be 1 (2,3) because \*pargs argument will take 2 and 3 values and convert it into tuple.

4. What does this code print, and why?

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

print(a, kargs)

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

Ans: The output will be 1 {‘b’:2, ‘c’: 3} because \*\*kargs argument will convert the b and c values to dictionary elements.

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: The output will be ‘1 5 6 5’ because \*(5,6) will break into two values and get assigned to b and c of function arguments.

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)

Ans: The output will be 1 [‘x’] {‘a’:’y’} because the value of l is not changeable in the function and value of b[0] will becomes m[0] which will change it to x and then dictionary value of ‘a’ will change into ‘y’