

PYTHON 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)
```

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)
```

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

```
>>> def func(a, *pargs):  
    print(a, pargs)  
  
>>> func(1, 2, 3)
```

4. What does this code print, and why?

```
>>> def func(a, **kargs):  
    print(a, kargs)  
  
>>> func(a=1, c=3, b=2)
```

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))
```

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) &gt;>> l, m, n
```

SOLUTIONS

1. Result and Explanation

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

```
    print(a, b, c)
```

```
func(1, 2)
```

Output: 1 2 8

Explanation: The function `func` is called with arguments a=1, b=2, and the default value c=8.

The values are printed as specified in the function.

2. Result and Explanation

```
def func(a, b, c=5):
```

```
    print(a, b, c)
```

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

Output: 1 2 3

Explanation: The function `func` is called with arguments a=1, b=2, and c=3.

The values are printed as specified in the function.

3. Result and Explanation

```
def func(a, *pargs):
```

```
    print(a, pargs)
```

```
func(1, 2, 3)
```

```
# Output: 1 (2, 3)
```

```
# Explanation: The function `func` is called with arguments a=1 and the remaining arguments packed into pargs using *args.
```

4. Result and Explanation

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

```
    print(a, kargs)
```

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

```
# Output: 1 {'c': 3, 'b': 2}
```

```
# Explanation: The function `func` is called with keyword arguments a=1, b=2, and c=3, which are packed into the dictionary kargs.
```

5. Result and Explanation

```
def func(a, b, c=8, d=5):
```

```
    print(a, b, c, d)
```

```
func(1, *(5, 6))
```

```
# Output: 1 5 6 5
```

```
# Explanation: The function `func` is called with arguments a=1, *args=(5, 6), which unpacks to b=5 and c=6. The default values for d are used.
```

6. Result and Explanation

```
def func(a, b, c):
```

```
    a = 2
```

```
    b[0] = 'x'
```

```
    c['a'] = 'y'
```

```
l = 1
```

```
m = [1]
```

```
n = {'a': 0}
```

```
func(l, m, n)
```

```
print(l, m, n)
```

```
# Output: 1 ['x'] {'a': 'y'}
```

```
# Explanation: The function `func` modifies the values of its arguments.
```

```
# After calling the function, the values of l, m, and n are printed.
```