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

**&gt;&gt;&gt; def func(a, b=6, c=8):**

**print(a, b, c)**

**&gt;&gt;&gt; func(1, 2)**

**2. What is the result of this code, and why?**

**&gt;&gt;&gt; def func(a, b, c=5):**

**print(a, b, c)**

**&gt;&gt;&gt; func(1, c=3, b=2)**

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

**&gt;&gt;&gt; def func(a, \*pargs):**

**print(a, pargs)**

**&gt;&gt;&gt; func(1, 2, 3)**

**4. What does this code print, and why?**

**&gt;&gt;&gt; def func(a, \*\*kargs):**

**print(a, kargs)**

**&gt;&gt;&gt; func(a=1, c=3, b=2)**

**5. What gets printed by this, and explain?**

**&gt;&gt;&gt; def func(a, b, c=8, d=5): print(a, b, c, d)**

**&gt;&gt;&gt; func(1, \*(5, 6))**

**6. what is the result of this, and explain?**

**&gt;&gt;&gt; def func(a, b, c): a = 2; b[0] = &#39;x&#39;; c[&#39;a&#39;] = &#39;y&#39;**

**&gt;&gt;&gt; l=1; m=[1]; n={&#39;a&#39;:0}**

**&gt;&gt;&gt; func(l, m, n)**

**&gt;&gt;&gt; l, m, n**

***SOLUTIONS***

1. *The code will output "1 2 8". The function func has three parameters a, b, and c, with default values of 6 and 8 for b and c respectively. When func(1, 2) is called, a takes the value 1, b takes the value 2, and c takes its default value of 8.*
2. *The code will output "1 2 3". The function func has three parameters a, b, and c with a default value of 5 for c. When func(1, c=3, b=2) is called, a takes the value 1, b takes the value 2, and c takes the value 3.*
3. *The code will output "1 (2, 3)". The function func has two parameters, a and \*pargs. pargs is a variable-length argument list. When func(1, 2, 3) is called, a takes the value 1, and pargs takes the tuple (2, 3).*
4. *The code will output "1 {'c': 3, 'b': 2}". The function func has two parameters, a and \*\*kargs. kargs is a variable-length keyword argument dictionary. When func(a=1, c=3, b=2) is called, a takes the value 1, and kargs takes the dictionary {'c': 3, 'b': 2}.*
5. *The code will output "1 5 6 5". The function func has four parameters, a, b, c, and d, with default values of 8 and 5 for c and d respectively. When func(1, \*(5, 6)) is called, a takes the value 1, b takes the first value of the tuple (5, 6), which is 5, c takes the second value of the tuple (5, 6), which is 6, and d takes its default value of 5.*
6. *The code will output "1 ['x'] {'a': 'y'}". The function func has three parameters, a, b, and c. a, b, and c are assigned the values of l, m, and n respectively. In the function body, a is assigned the value 2, but this does not affect the value of l outside the function. The first element of the list m is modified to be 'x', and the value of m becomes ['x']. The value of the dictionary n is modified to be {'a': 'y'}. When the code is executed, l, m, n will print "1 ['x'] {'a': 'y'}".*