Python Tuples, Lists, and Dicts - Selected Questions

- 1. What happens when you multiply a tuple by a constant (e.g., tup * 3)?
- a) It creates multiple copies of the objects in the tuple.
- b) It creates multiple references to the same objects in the tuple.
- c) It creates a new tuple with mutable objects copied.
- d) It creates a shallow copy of each object.
- 6. Given the following code:

$$a = 92, 34$$

$$b = 76, 54$$

$$a, b = b, a$$

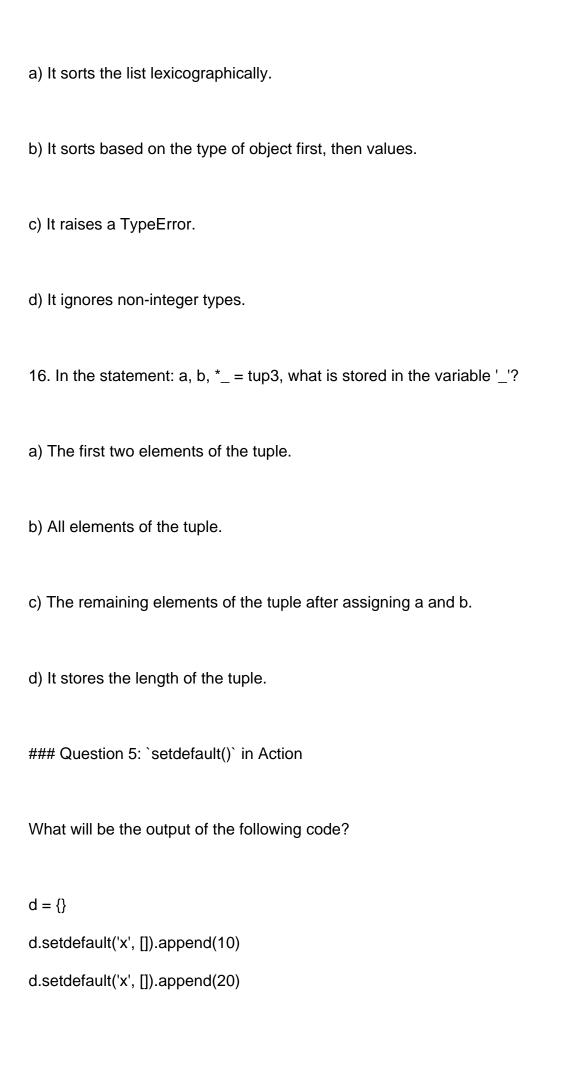
What is the value of 'a' and 'b' after swapping?

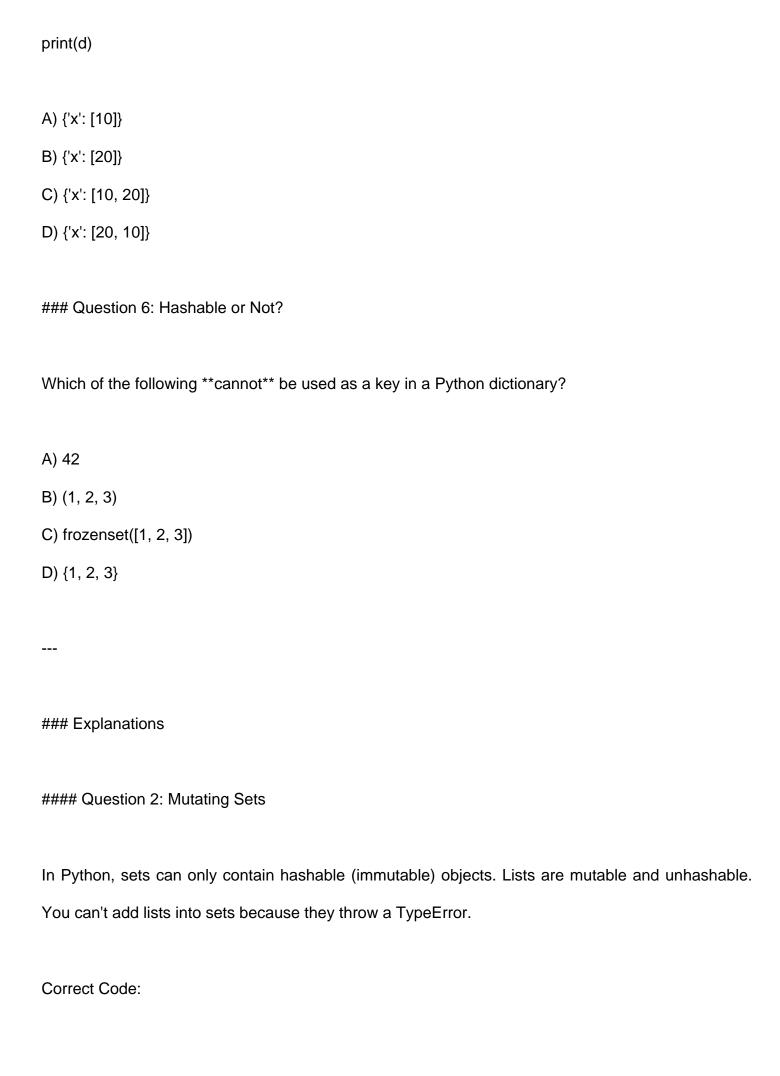
a)
$$a = (92, 34), b = (76, 54)$$

b)
$$a = 76, 54$$
 and $b = 92, 34$

d)
$$a = [76, 54], b = [92, 34]$$

- 8. Which of the following methods is generally faster for adding elements from one list to another?
- a) Using extend() method
- b) Using the + operator
- c) Using append() in a loop
- d) Using insert() at index 0
- 11. Which of the following is the correct way to unpack a tuple while ignoring some of its values?
- a) a, b, c = tup
- b) a, *_, b = tup
- c) a, b, *_ = tup
- d) a, b, *rest = tup
- 13. What happens if you try to sort a list that contains both integers and strings?





```
a = \{1, 2, 3\}
```

 $b = \{(4, 5, 6)\}$ # Tuples are hashable and allowed in sets

Question 8: Comprehension Creation

We need to get the length of each word and collect them in a set to ensure uniqueness.

Code:

```
words = ["python", "java", "go", "swift"]
lengths = {len(word) for word in words}
print(lengths)
```

Output:

 $\{2, 4, 6\}$

Question 15: One-Liner Challenge

We need to double every number in `a` without using an explicit loop.

Code Option 1 (List comprehension):

$$a = [1, 2, 3, 4]$$

result = [x * 2 for x in a]

Code Option 2 (Using map and lambda):

$$a = [1, 2, 3, 4]$$

result = list(map(lambda x: x * 2, a))

Both will output:

[2, 4, 6, 8]