question
What is the expected output of the following code?

```
x = [1, 2, 3, 4, 5, 6, 7, 8, 9]
x[::2] = 10, 20, 30, 40, 50, 60
print(x)
```

options

a) The code is erroneous
b) [1, 2, 10, 20, 30, 40, 50, 60]
c) [1, 10, 3, 20, 5, 30, 7, 40, 9, 50, 60]
d) [10, 2, 20, 4, 30, 6, 40, 8, 50, 60]

answer
option **a**

Original List
x = [1, 2, 3, 4, 5, 6, 7, 8, 9]
This is a list of 9 elements.

Slicing and Assignment:
x[::2] = 10, 20, 30, 40, 50, 60

Here, x[::2] means:
Start from the beginning (start is omitted, defaults to 0), Go till the end (stop is omitted), Step by 2.

So x[::2] selects every second element, starting from index 0:

Indices selected: 0, 2, 4, 6, 8
So: x[0], x[2], x[4], x[6], x[8] → currently these are [1, 3, 5, 7, 9]
So, you're trying to assign:

x[0] = 10
x[2] = 20
x[4] = 30
x[6] = 40
x[8] = 50
But there are 6 values on the right (10, 20, 30, 40, 50, 60) and only 5 positions being targeted in the list (due to the step of 2).

So what happens?
Python raises a ValueError because the number of elements you're assigning doesn't match the number of positions you're slicing.

Final Answer
ValueError: attempt to assign sequence of size 6 to extended slice of size 5
Let me know if you want to explore how to fix or work around this!

what should be the correct one ??
To make it correct, the number of elements you're assigning must exactly match the number of elements selected by the slice.

```
x = [1, 2, 3, 4, 5, 6, 7, 8, 9]
x[::2]  # selects indices 0, 2, 4, 6, 8 → total 5 elements
So, you must assign exactly 5 values.

x = [1, 2, 3, 4, 5, 6, 7, 8, 9]
x[::2] = 10, 20, 30, 40, 50
print(x)

x[0] = 10
x[2] = 20
x[4] = 30
x[6] = 40
x[8] = 50

Final list
[10, 2, 20, 4, 30, 6, 40, 8, 50]
```