

1. What will be the output of the following code?

```
word = "Pythonista"  
print(word[-4])
```

a) n

b) i

c) s

d) t

Explanation of Question 1:

এখানে word[-4] মানে হলো স্ট্রিংয়ের শেষ দিক থেকে চতুর্থ অক্ষর।

word = "Pythonista" এর ইন্ডেক্স (নেগেটিভভাবে):

P	y	t	h	o	n	i	s	t	a
-10	-9	-8	-7	-6	-5	-4	-3	-2	-1

তাহলে -4 ইন্ডেক্সে আছে 'i'।

2. What will the code print?

```
s = "HelloWorld"  
print(s[2:8:2])
```

a) loWo

b) lowo

c) loWr

d) loo

Explanation of Question 2:

এখানে s[2:8:2] মানে শুরু হবে index 2 থেকে এবং শেষ হবে index 8 এর আগ পর্যন্ত, প্রতি স্টেপে index 2 করে বাড়বে।

H	e	l	l	o	W	o	r	l	d
0	1	2	3	4	5	6	7	8	9

শুরু index 2 তে যাবে যার ফলে 'l' প্রিন্ট হবে

এরপর 2 ধাপ বেড়ে index 4 এ যাবে যার ফলে 'o' প্রিন্ট হবে

এরপর 2 ধাপ বেড়ে index 6 এ যাবে যার ফলে 'o' প্রিন্ট হবে

3. What will be the output?

```
text = "banana bandana"  
print(text.count("ana"))
```

a) 2

b) 3

c) 1

d) 4

Explanation of Question 3:

স্ট্রিং এবং ইন্ডেক্স:

b	a	n	a	n	a		b	a	n	d	a	n	a
0	1	2	3	4	5	6	7	8	9	10	11	12	13

Python-এর str.count() overlapping substring গণনা করে না। অর্থাৎ, "banana"-এর মধ্য "ana" আছে কি না চেক করলে, Python শুধুমাত্র non-overlapping বার গণনা করে।

"banana" এর মধ্যে index 1–3 তে "ana" পেয়েছে সেইটি count করবে।

index 3–5 তে "ana" আছে এটি overlapping, Python count করবে না।

"bandana" এর মধ্যে index 10–12 তে "ana" পেয়েছে সেইটি count করবে।

মোট non-overlapping: 2

4. What will be the output?

```
msg = "red,green,blue"  
colors = msg.split(",")  
print("-".join(colors[::-1]))
```

a) red-green-blue

b) blue-green-red

c) red-blue-green

d) green-red-blue

Explanation of Question 4:

`msg.split(",")` এর মাধ্যমে `msg` string টি কমা দিয়ে ভাগ হয়ে list এর cover হয়ে যাবে। যেইটি `colors` variable এর মধ্যে assign করা হয়েছে। যারা `colors` variable টি এইরকম দেখাবে।

```
colors = ["red", "green", "blue"]
```

এরপর `colors[::-1]` দিয়ে list টি reverse করা হয়েছে, যেইটিকে

`"-".join(colors[::-1])` দিয়ে list এর items - দিয়ে যুক্ত করে একটি string তৈরি হয়েছে সেইটিকে প্রিন্ট করা হয়েছে।

5. What will be the output?

```
nums = [2,4,6,8,10]
nums[2] = nums[2] * nums[0]
print(nums)
```

a) [2, 4, 12, 8, 10]

b) [2, 4, 6, 8, 10]

c) [4, 8, 12, 16, 20]

d) Error

Explanation of Question 5:

এইখানে

`nums[2]` মানে ৩য় এলিমেন্ট → 6

`nums[0]` মানে ১ম এলিমেন্ট → 2

তাহলে `nums[2] = nums[2] * nums[0]` এর জন্য

`nums[2] = 6 * 2 = 12`

2 index এর মধ্যে 12 assign হয়ে যাবে।

তাই নতুন লিস্ট হবে: [2, 4, 12, 8, 10]

6. What is the output?

```
letters = ['a','b','c','d','e']
print(letters[1:4][::-2])
```

a) ['b','c','d']

b) ['d', 'b']

c) ['e','d','c']

d) ['b', 'd']

Explanation of Question 6:

letters[1:4] মানে index 1 থেকে 3 পর্যন্ত → ['b','c','d'] নেয়া।

এখন[::-2] মানে উল্টে গিয়ে প্রতি দুই index আগানো তাহলে ['b','c','d'] উল্টালে হয় ['d','c','b'], তারপর প্রতি 2 স্টেপে নিলে → ['d','b'] হয়।

7. What will be printed?

```
stack = []  
for i in range(1,4):  
    stack.append(i)  
stack.pop()  
print(stack[-1])
```

- a) 1
- b) 2**
- c) 3
- d) Error

Explanation of Question 7:

for i in range(1,4) লুপটি i = 1, 2, 3 পর্যন্ত চলবে।

তাহলে লুপ শেষে stack = [1, 2, 3] হবে।

তারপর stack.pop() দিলে — শেষটা (৩) বের হয়ে যাবে।

এখন stack = [1, 2] হবে।

stack[-1] মানে শেষ element যেইটি হচ্ছে 2।

8. What will be printed?

```
queue = [10,20,30,40]  
queue.pop(1)  
queue.append(50)  
queue.pop(0)  
print(queue[0])
```

- a) 10
- b) 20
- c) 30**
- d) 50

Explanation of Question 8:

এখন queue = [10, 20, 30, 40]

queue.pop(1) হলে index 1 মানে 20 বাদ যাবে → তাহলে এখন queue তে [10, 30, 40] এইগুলো থাকে।

queue.append(50) হলে শেষে 50 অ্যাড হয়ে যাবে → তাহলে এখন queue তে [10, 30, 40, 50] এইগুলো থাকে।

queue.pop(0) হলে index 0 মানে 10 বাদ যাবে → তাহলে এখন queue তে [30, 40, 50] এইগুলো থাকে।

এখন queue[0] মানে প্রথম element → 30

9. What will be the output?

```
nums = [1,2,3,4]
squares = [x**2 for x in nums if x%2==0]
print(squares)
```

a) [1,4,9,16]

b) [4,16]

c) [2,4]

d) [1,9]

Explanation of Question 9:

nums = [1,2,3,4]

if x%2==0 এর জন্য শুধু জোড় সংখ্যা নেবে → 2, 4 , x variable হিসেবে।

তারপর প্রতিবার x square করা হবে।

$2**2 = 4$, $4**2 = 16$

ফলে squares = [4,16] হবে।

10. What is the output?

```
x = [1,2,3,2,4]
x.remove(4)
x.pop()
print(x)
```

a) [1,3,2]

b) [1,3,4]

c) [1,2,3]

d) [1,3]

Explanation of Question 10:

`x = [1,2,3,2,4]`

`x.remove(4)` এর ফলে 4 মুছে যাবে , এরপর list টি হবে→ `[1,2,3,2]`।

`x.pop()` এর ফলে শেষ element (শেষের 2) বাদ পড়ে যাবে, এরপর list টি হবে→ `[1,2,3]`