FOOMO AI Interview – Easy Coding Round

Show the following code snippets to candidates and ask questions about logic, output, and possible optimizations. These questions target easy-level screening for fresher software engineer roles.

Question 1 — Two Sum

```
arr = [2, 7, 11, 15]
target = 9
seen = {}

for i, num in enumerate(arr):
    diff = target - num
    if diff in seen:
        print(seen[diff], i)
        break
    seen[num] = i

Ask:
1. What is the output?
2. Why use dictionary?
3. Time complexity?
4. Brute-force complexity?
```

Question 2 — Move Zeros to End

```
arr = [0, 1, 0, 3, 12]
result = []

for num in arr:
    if num != 0:
        result.append(num)

zeros = len(arr) - len(result)
result += [0] * zeros

print(result)

Ask:
1. Output?
2. Why count zeros?
3. Can this be done in-place?
4. Complexity?
```

Question 3 — Valid Palindrome

```
s = "madam"
is_pal = True

for i in range(len(s)//2):
    if s[i] != s[-1-i]:
        is_pal = False

print(is_pal)

Ask:
1. Output?
2. Why loop half?
3. Python shortcut?
4. Complexity?
```

Question 4 — First Non-Repeating Character

```
s = "aabbcddee"
freq = {}

for ch in s:
    freq[ch] = freq.get(ch, 0) + 1
```

```
for ch in s:
    if freq[ch] == 1:
        print(ch)
        break

Ask:
1. Output?
2. Why two loops?
3. What does get() do?
4. Complexity?
```

Question 5 — Binary Search

```
arr = [1,3,5,7,9]
target = 7
l, r = 0, len(arr)-1

while l <= r:
    mid = (l + r) // 2
    if arr[mid] == target:
        print("Found")
        break
    elif arr[mid] < target:
        l = mid + 1
    else:
        r = mid - 1

Ask:
1. Output?
2. Why must array be sorted?
3. Time complexity?
4. Compare with linear search.
```

Question 6 — Maximum Subarray Sum

```
arr = [-2,1,-3,4,-1,2,1,-5,4]
max_sum = arr[0]
curr = 0

for num in arr:
    curr = max(num, curr + num)
    max_sum = max(max_sum, curr)

print(max_sum)

Ask:
1. Output?
2. What does curr represent?
3. Complexity?
4. Real-world use case?
```

Question 7 — Remove Duplicates

```
arr = [1, 2, 2, 3, 4, 4, 5]
unique = []

for num in arr:
    if num not in unique:
        unique.append(num)

print(unique)

Ask:
1. Output?
2. Why check not in unique?
3. Optimize using set?
4. Complexity?
```

Question 8 — Character Frequency

```
s = "banana"
freq = {}
```

```
for ch in s:
    freq[ch] = freq.get(ch, 0) + 1

print(freq)

Ask:
1. Output?
2. What does get() do?
3. Most frequent character?
4. Complexity?
```

Question 9 — Array Intersection

```
a = [1, 2, 3, 4]
b = [3, 4, 5, 6]

result = []

for x in a:
    if x in b:
        result.append(x)

print(result)

Ask:
1. Output?
2. Optimize using set?
3. Complexity?
4. Use case?
```

Question 10 — Reverse Words

```
s = "hello world foomo"
words = s.split()
words.reverse()
print(" ".join(words))

Ask:
1. Output?
2. What does split() do?
3. What does join() do?
4. Complexity?
```

Question 11 — Check Sorted Array

```
arr = [1, 3, 5, 7, 9]
sorted_flag = True

for i in range(len(arr) - 1):
    if arr[i] > arr[i + 1]:
        sorted_flag = False

print(sorted_flag)

Ask:
1. Output?
2. Condition for sorting?
3. Complexity?
4. Modify for descending order?
```