

Python Interview Questions and Answers

1. Compress String (Run-Length Encoding) – Variation

Input: aaabbca

Output: 3a2b1c1a

```
s = "aaabbca"  
  
result = ""  
  
count = 1  
  
for i in range(len(s)):  
    if i < len(s)-1 and s[i] == s[i+1]:  
        count += 1  
    else:  
        result += str(count) + s[i]  
        count = 1  
  
print(result)
```

2. Reverse each word in a sentence

Input: "hello world"

Output: "olleh dlrow"

```
s = "hello world"  
  
words = s. split ()  
  
result = " ".join(word[::-1] for word in words)  
  
print(result)
```

3. Find first non-repeating character

Input: "aabbcddde"

Output: c

```
s = "aabbcddde"  
  
for ch in s:  
    if s.count(ch) == 1:
```

```
print(ch)
```

```
break
```

✓ 4. Count vowels in a string

Input: "engineering"

Output: e=3, i=2

```
s = "engineering"
```

```
vowels = "aeiou"
```

```
freq = {}
```

```
for ch in s:
```

```
    if ch in vowels:
```

```
        freq[ch] = freq.get(ch, 0) + 1
```

```
print(freq)
```

✓ 5. Remove duplicates but keep order

Input: "banana"

Output: "ban"

```
s = "banana"
```

```
result = ""
```

```
for ch in s:
```

```
    if ch not in result:
```

```
        result += ch
```

```
print(result)
```

✓ 6. Check if two strings are anagrams

Input: "silent", "listen"

Output: True

```
s1 = "silent"
```

```
s2 = "listen"
```

```
print(sorted(s1) == sorted(s2))
```

✓ 7. Frequency of each character

Input: "malayalam"

Output: {'m':2,'a':4,'l':2,'y':1}

```
s = "malayalam"
```

```
freq = {}
```

```
for ch in s:
```

```
    freq[ch] = freq.get(ch, 0) + 1
```

```
print(freq)
```

✓ 8. Find max occurring character

Input: "success"

Output: s

```
s = "success"
```

```
freq = {}
```

```
for ch in s:
```

```
    freq[ch] = freq.get(ch, 0) + 1
```

```
print(max(freq, key=freq.get))
```

✓ 9. Swap two numbers without third variable

```
a = 5
```

```
b = 10
```

```
a, b = b, a
```

```
print(a, b)
```

✓ 10. Reverse a string using loop (interview favourite)

```
s = "suresh"
```

```
rev = ""
```

```
for ch in s:
```

```
    rev = ch + rev
```

```
print(rev)
```

✓ 11. Palindrome check (string)

Input: "radar" — **Output:** True

```
s = "radar"
```

```
print(s == s[::-1])
```

✓ 12. Check if number is prime

Input: 29 — **Output:** True

```
n = 29
```

```
if n <= 1:
```

```
    print(False)
```

```
else:
```

```
    for i in range(2, int(n**0.5)+1):
```

```
        if n % i == 0:
```

```
            print(False)
```

```
            break
```

```
    else:
```

```
        print(True)
```

✓ 13. Fibonacci (n-th) using iteration

Input: n=7 — **Output:** 13 (sequence: 0,1,1,2,3,5,8,13)

```
n = 7  
a, b = 0, 1  
for _ in range(n):  
    a, b = b, a + b  
print(a)
```

✓ 14. Remove duplicates from list (preserve order)

Input: [1,2,2,3,1,4] — **Output:** [1,2,3,4]

```
lst = [1,2,2,3,1,4]  
seen = set()  
res = []
```

```
for x in lst:  
    if x not in seen:  
        seen.add(x)  
        res.append(x)
```

```
print(res)
```

✓ 15. Two-sum (indices) — O(n)

Input: nums=[2,7,11,15], target=9 — **Output:** [0,1]

```
nums = [2,7,11,15]; target = 9  
d = {}  
for i, v in enumerate(nums):  
    if target - v in d:  
        print([d[target-v], i]); break
```

```
d[v] = i
```

✓ 16. Merge two sorted lists (like merge step)

Input: [1,3,5], [2,4,6] — **Output:** [1,2,3,4,5,6]

```
a, b = [1,3,5], [2,4,6]  
i=j=0; out=[]  
while i<len(a) and j<len(b):  
    if a[i] < b[j]:
```

```
out.append(a[i]); i+=1  
else:  
    out.append(b[j]); j+=1  
out.extend(a[i:]); out.extend(b[j:])  
print(out)
```

✓ 17. Check balanced parentheses

Input: "([])" — **Output:** True

```
s = "([ ])" .replace(" ", "")  
pairs = {'(': ')', '[': ']', '{': '}'}  
stack = []  
for ch in s:  
    if ch in pairs:  
        stack.append(ch)  
    else:  
        if not stack or pairs[stack.pop()] != ch:  
            print(False); break  
    else:  
        print(not stack)
```

✓ 18. Flatten nested list (single level)

Input: [[1,2],[3,4],5] — **Output:** [1,2,3,4,5]

```
nested = [[1,2],[3,4],5]  
flat = []  
for item in nested:  
    if isinstance(item, list):  
        flat.extend(item)  
    else:  
        flat.append(item)  
print(flat)
```

✓ 19. Count words in sentence

Input: "hello hello world" — **Output:** {'hello':2,'world':1}

```
s = "hello hello world"
```

```
freq = {}
```

```
for w in s.split():
```

```
    freq[w] = freq.get(w,0) + 1
```

```
print(freq)
```

✓ 20. Check if list is palindrome

Input: [1,2,3,2,1] — **Output:** True

```
lst = [1,2,3,2,1]
```

```
print(lst == lst[::-1])
```

✓ 21. Find kth largest (using heapq)

Input: nums=[3,2,1,5,6,4], k=2 — **Output:** 5

```
import heapq
```

```
nums=[3,2,1,5,6,4]; k=2
```

```
print(heapq.nlargest(k, nums)[-1])
```

✓ 22. Generate primes up to n (Sieve of Eratosthenes)

Input: n=15 — **Output:** [2,3,5,7,11,13]

```
n = 15
```

```
is_prime = [True]*(n+1)
```

```
is_prime[0]=is_prime[1]=False
```

```
p=2
```

```
while p*p<=n:
```

```
    if is_prime[p]:
```

```
        for i in range(p*p, n+1, p):
```

```
            is_prime[i]=False
```

```
p+=1
```

```
print([i for i,v in enumerate(is_prime) if v])
```

✓ 23. Rotate string by k (left rotate)

Input: s="abcdef", k=2 — **Output:** "cdefab"

```
s = "abcdef"; k = 2  
k %= len(s)  
print(s[k:] + s[:k])
```

✓ 24. Singleton pattern (simple OOP interview Q)

Output: Demonstrates single instance

class Singleton:

```
_instance = None  
  
def __new__(cls, *args, **kwargs):  
    if not cls._instance:  
        cls._instance = super().__new__(cls)  
    return cls._instance
```

```
a = Singleton(); b = Singleton()
```

```
print(a is b) # True
```

✓ 25. Generator: infinite Fibonacci (yield)

Usage: print first 8 numbers — **Output:** 0 1 1 2 3 5 8 13

```
def fib_gen():
```

```
    a, b = 0, 1
```

```
    while True:
```

```
        yield a
```

```
        a, b = b, a + b
```

```
g = fib_gen()
```

```
for _ in range(8):
```

```
    print(next(g), end=" ")
```

✓ 26. Count Frequency of Elements in a List Using a Dictionary

```
data = ['apple', 'banana', 'apple', 'orange', 'banana', 'apple']
```

Ans. # Create an empty dictionary

```

frequency = {}

# Loop through each element in the list
for item in data:

    if item in frequency:

        frequency[item] += 1 # Increase count if item exists

    else:

        frequency[item] = 1 # Initialize count if new item

print(frequency)

 Output:
{'apple': 3, 'banana': 2, 'orange': 1}

```

27. **Variable = 'suresh' reverse a string with out using reverse and use for loop**

Ans.

```

variable = 'suresh'

reversed_string = ""

for char in variable:

    reversed_string = char + reversed_string # add each char in front

print(reversed_string)

```

28. **num = [15, 50, 5, 9, 20] find the 3rd largest number**

```

num = [15, 50, 5, 9, 20]

third_largest = sorted(num, reverse=True)[2]

print("3rd largest number:", third_largest)

```

OR

```

def third_larg(nums):

    nums = sorted (nums, reverse = True)

    return nums[2]

num = [15,50,5,9,20]

print("third largest number is ::",third_larg(nums))

```

29. **Find the 3rd Largest Number in a List**

```
numbers = [15, 50, 5, 9, 20]
numbers.sort(reverse=True)
print("3rd largest:", numbers[2])
# Output: 3rd largest: 15
```

✓ 30. Reverse a String Using a Loop

```
text = "suresh"
reversed_text = ""
for ch in text:
    reversed_text = ch + reversed_text
print(reversed_text)
# Output: hserus
```

✖ ✓ 31. Find Duplicate Elements in a List

```
data = [1, 2, 3, 4, 2, 3, 5, 1]
duplicates = []
for i in data:
    if data.count(i) > 1 and i not in duplicates:
        duplicates.append(i)
print("Duplicates:", duplicates)
# Output: Duplicates: [1, 2, 3]
```

✓ 32. Altrimatik --list=[1, 2, [3, 4], 5, [6, 7, 8]] - single list

O/P = lst = [1,2,3,4,5,6,7,8]

```
--data = [1, 2, [3, 4], 5, [6, 7, 8]]
result = []
```

```
for item in data:
```

```
    if isinstance (item, list):
```

```
    result.extend(item)
else:
    result.append(item)
print(result)
```

Explanation:-

```
if isinstance(item, list):
    isinstance(item, list) returns True if item is a list
```

It returns False if the item is not a list.

Item Is it a list? Result

1		No	else block
2		No	else block
[3, 4]		Yes	extend block
5		No	else block
[6, 7, 8]		Yes	extend block

If it is a list → add its elements to result

```
result.extend(item)
```

- extend() adds **each element** of the list individually.

Example:

result.extend([3,4]) → adds 3 and 4 into result, not the whole list.

6 If it is NOT a list → add it normally

```
else:
```

```
    result.append(item)
```

append() adds the value as a **single element**

Senario :Altrimatik (Company) :-

33.write a Python program to conduct match one country with other countries

```
list= ['USA','India','Australia','Pakistan','South Africa']
```

Ans . countries = ['USA', 'India', 'Australia', 'Pakistan', 'South Africa']

```
for i in range(len(countries)):
```

```
for j in range(len(countries)):  
    if i != j:    # avoid matching the same country with itself  
        print(countries[i], "matches with", countries[j])
```

OR

```
countries = ['USA', 'India', 'Australia', 'Pakistan', 'South Africa']  
for i in range(len(countries)):  
    for j in range(i+1, len(countries)):
```

```
        print(countries[i], "matches with", countries[j])
```

✓ 34. Altrimatik (Company) :-

Q. Sorting a list of dictionaries in Python:

```
students = [  
    {"name": "Alice", "age": 20, "grade": "A"},  
    {"name": "Bob", "age": 22, "grade": "C"},  
    {"name": "Charlie", "age": 19, "grade": "B"},  
    {"name": "David", "age": 20, "grade": "A+"},  
    {"name": "Eve", "age": 21, "grade": "B-"},  
]
```

--- Sorting by 'age' (ascending order) –

Ans .

```
sorted_students = sorted(students, key=lambda x: x['age'])  
print(sorted_students)
```

✓ 35. Check if a String is a Palindrome

```
word = "madam"  
if word == word[::-1]:  
    print("Palindrome")  
else:  
    print("Not Palindrome")  
# Output: Palindrome
```

36. Count Vowels in a String

```
text = "hello world"  
vowels = "aeiou"  
count = 0  
  
for ch in text:  
    if ch in vowels:  
        count += 1  
  
print ("Vowel count:", count)  
# Output: Vowel count: 3
```

37. Find Even and Odd Numbers in a List

```
numbers = [1, 2, 3, 4, 5, 6]  
  
even = [n for n in numbers if n % 2 == 0]  
odd = [n for n in numbers if n % 2 != 0]  
  
print("Even:", even)  
print("Odd:", odd)  
  
# Output: Even: [2, 4, 6], Odd: [1, 3, 5]
```

38. Remove Duplicates from a List

```
data = [1, 2, 3, 2, 4, 3, 5]  
  
unique = list(set(data))  
  
print(unique)  
  
# Output: [1, 2, 3, 4, 5]
```

39. Swap Two Variables Without a Temp Variable

```
a, b = 10, 20  
  
a, b = b, a  
  
print(a, b)  
  
# Output: 20 10
```

  **40. Find Maximum and Minimum in a List**

```
data = [12, 45, 2, 10, 34, 22]
print("Max:", max(data))
print("Min:", min(data))
# Output: Max: 45, Min: 2
```

  **41. Find Common Elements Between Two Lists**

```
list1 = [1, 2, 3, 4]
list2 = [3, 4, 5, 6]
common = list(set(list1) & set(list2))
print("Common elements:", common)
# Output: [3, 4]
```

  **42: Print Fibonacci Sequence**

```
n = 10
a, b = 0, 1
for _ in range(n):
    print(a, end=" ")
    a, b = b, a + b
# Output: 0 1 1 2 3 5 8 13 21 34
```

  **43: Find Factorial of a Number**

```
num = 5
fact = 1
for i in range(1, num + 1):
    fact *= i
print("Factorial:", fact)
# Output: 120
```

  **44: Count Occurrence of Each Character in a String**

```
text = "banana"  
count = {}  
for ch in text:  
    count[ch] = count.get(ch, 0) + 1  
print(count)  
# Output: {'b': 1, 'a': 3, 'n': 2}
```

  **45: Merge Two Dictionaries**

```
dict1 = {'a': 10, 'b': 20}  
dict2 = {'c': 30, 'd': 40}  
merged = {**dict1, **dict2}  
print(merged)  
# Output: {'a': 10, 'b': 20, 'c': 30, 'd': 40}
```

  **46. Count Words in a Sentence**

```
sentence = "Python is fun and Python is powerful"  
words = sentence.split()  
count = {}  
for w in words:  
    count[w] = count.get(w, 0) + 1  
print(count)  
# Output: {'Python': 2, 'is': 2, 'fun': 1, 'and': 1, 'powerful': 1}
```



 **47: Find the Second Highest Number in a List**

```
data = [10, 35, 50, 20, 35]  
unique = list(set(data))  
unique.sort()
```

```
print("2nd highest:", unique[-2])
```

```
# Output: 35
```

✓ 48. Sort Dictionary by Its Values

```
data = {'apple': 3, 'banana': 1, 'orange': 2}  
sorted_data = dict(sorted(data.items(), key=lambda x: x[1]))  
print(sorted_data)  
# Output: {'banana': 1, 'orange': 2, 'apple': 3}
```

✓ 49. Merge Two Lists into a Dictionary

```
keys = ['name', 'age', 'city']  
values = ['John', 25, 'Hyderabad']  
merged = dict(zip(keys, values))  
print(merged)  
# Output: {'name': 'John', 'age': 25, 'city': 'Hyderabad'}
```

✓ 50. Find All Prime Numbers Between 1 and 50

```
for num in range(2, 51):  
    for i in range(2, num):  
        if num % i == 0:  
            break  
    else:  
        print(num, end=" ")  
# Output: 2 3 5 7 11 13 17 19 23 29 31 37 41 43 47
```

✓ 51.: Find Common Keys in Two Dictionaries

```
d1 = {'a': 1, 'b': 2, 'c': 3}  
d2 = {'b': 5, 'c': 7, 'd': 9}  
common_keys = list(d1.keys() & d2.keys())
```

```
print(common_keys)
```

```
# Output: ['b', 'c']
```

✓ 52: Remove Punctuation from a String

```
import string
```

```
text = "Hello, how are you?"
```

```
clean = ''.join(ch for ch in text if ch not in string.punctuation)
```

```
print(clean)
```

```
# Output: Hello how are you
```

✓ 53: Find Numbers Divisible by Both 3 and 5

```
nums = [i for i in range(1, 51) if i % 3 == 0 and i % 5 == 0]
```

```
print(nums)
```

```
# Output: [15, 30, 45]
```

✓ 54: Find Maximum Occurring Character in a String

```
text = "mississippi"
```

```
freq = {}
```

```
for ch in text:
```

```
    freq[ch] = freq.get(ch, 0) + 1
```

```
max_char = max(freq, key=freq.get)
```

```
print("Max occurring character:", max_char)
```

```
# Output: i
```

-

- ✓✓ 55: Flatten a Nested List

```
nested = [[1, 2], [3, 4, 5], [6]]
```

```
flat = [x for sublist in nested for x in sublist]
```

```
print(flat)
```

```
# Output: [1, 2, 3, 4, 5, 6]
```

56:: Find Missing Numbers from a Sequence

```
nums = [1, 2, 4, 6, 7, 9]  
missing = [i for i in range(min(nums), max(nums)) if i not in nums]  
print("Missing numbers:", missing)  
# Output: [3, 5, 8]
```

57: Calculate the Sum of Digits of a Number

```
num = 12345  
total = sum(int(d) for d in str(num))  
print("Sum of digits:", total)  
# Output: 15
```

58: Count Uppercase and Lowercase Letters

```
text = "HelloWorld"  
upper = sum(1 for ch in text if ch.isupper())  
lower = sum(1 for ch in text if ch.islower())  
print("Uppercase:", upper, "Lowercase:", lower)  
# Output: Uppercase: 2 Lowercase: 8
```

59:Find Intersection of Two Sets

```
set1 = {1, 2, 3, 4}  
set2 = {3, 4, 5, 6}  
print("Intersection:", set1 & set2)  
# Output: {3, 4}
```

60:Find the Longest Word in a Sentence

```
sentence = "Python makes coding enjoyable and powerful"  
words = sentence.split()
```

```
longest = max(words, key=len)
print("Longest word:", longest)
# Output: enjoyable
```

Advanced Python Scenario Programs

61: Read a File and Count Number of Lines

```
# Suppose "sample.txt" has some text lines
with open("sample.txt", "r") as file:
    lines = file.readlines()
print("Total lines:", len(lines))
```

 *Use case:* Count records in a log file.

62: Write a List of Strings to a File

```
lines = ["apple\n", "banana\n", "orange\n"]
with open("fruits.txt", "w") as file:
    file.writelines(lines)
print("File written successfully!")
```

 *Use case:* Write processed data to a report file.

63: Read JSON Data from a File

```
import json

json_data = '{"name": "John", "age": 30, "city": "Hyderabad"}'
data = json.loads(json_data)
print(data["city"])
# Output: Hyderabad
```

 *Use case:* Reading configuration or API response data.

  64: Sort List of Dictionaries by a Key

```
employees = [  
    {"name": "Ravi", "age": 28},  
    {"name": "Anil", "age": 22},  
    {"name": "Kiran", "age": 35}  
]
```

```
sorted_employees = sorted(employees, key=lambda x: x["age"])  
print(sorted_employees)
```

 *Use case:* Sorting employee or transaction data.

  65: Filter Records Using lambda and filter()

```
numbers = [10, 25, 30, 45, 50, 60]  
divisible_by_10 = list(filter(lambda x: x % 10 == 0, numbers))  
print(divisible_by_10)  
# Output: [10, 30, 50, 60]
```

 *Use case:* Filtering data from datasets.

  66: Handle File Not Found Exception

```
try:  
    with open("unknown.txt", "r") as file:  
        print(file.read())  
    except FileNotFoundError:  
        print("File not found. Please check the filename.")
```

 *Use case:* Robust file reading with safe handling.

  67: Count Each Word's Frequency in a File

```
from collections import Counter
```

```
with open("text.txt", "r") as file:
```

```
    words = file.read().split()
```

```
word_count = Counter(words)
```

```
print(word_count)
```

 *Use case:* Word count in logs or reports (like wc in Linux).

68: Remove Empty Strings from a List

```
data = ["apple", "", "banana", "", "orange"]
```

```
cleaned = list(filter(None, data))
```

```
print(cleaned)
```

```
# Output: ['apple', 'banana', 'orange']
```

 *Use case:* Cleaning invalid data entries.

69: Find Employee with Highest Salary

```
employees = [
```

```
    {"name": "Ravi", "salary": 45000},
```

```
    {"name": "Kiran", "salary": 52000},
```

```
    {"name": "Anil", "salary": 48000}
```

```
]
```

```
highest = max(employees, key=lambda x: x["salary"])
```

```
print("Highest Paid:", highest["name"])
```

```
# Output: Kiran
```

 *Use case:* Ranking records by numeric value.

70: Convert CSV Data into Dictionary

```
import csv
```

```
with open("employees.csv", "r") as file:
```

```
reader = csv.DictReader(file)
```

```
for row in reader:
```

```
    print(row)
```

✓ *Use case:* Reading structured data files into Python objects.

✖ ✓ 71: Find Unique Words from a Sentence

```
sentence = "Python is powerful and Python is easy"
```

```
unique_words = set(sentence.lower().split())
```

```
print(unique_words)
```

```
# Output: {'and', 'python', 'is', 'easy', 'powerful'}
```

✓ *Use case:* Removing duplicates in text or logs.

✖ ✓ 72: Create a Dictionary from Two Lists (Only Even Numbers)

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

```
values = [10, 20, 30, 40]
```

```
even_dict = {k: v for k, v in zip(keys, values) if k % 2 == 0}
```

```
print(even_dict)
```

```
# Output: {2: 20, 4: 40}
```

✓ *Use case:* Conditional mapping in data pipelines.

✖ ✓ 73 Handle Division by Zero Error

```
try:
```

```
    result = 10 / 0
```

```
except ZeroDivisionError:
```

```
    result = None
```

```
    print("Cannot divide by zero.")
```

✓ *Use case:* Safe numeric operations in calculations.

✖ ✓ 74: Calculate Average Value from a Dictionary

```
marks = {"Math": 80, "Science": 90, "English": 70}  
avg = sum(marks.values()) / len(marks)  
print("Average marks:", avg)  
# Output: 80.0
```

Use case: Aggregating key-value pairs.

75:Find Top N Elements from a List

```
nums = [10, 50, 20, 80, 30, 90]  
N = 3  
top_n = sorted(nums, reverse=True) [: N]  
print("Top 3 elements:", top_n)  
# Output: [90, 80, 50]
```

76. What is the difference between list, tuple, and set?

Type Ordered Mutable Allows Duplicates

List		Yes		Yes		Yes
Tuple		Yes		No		Yes
Set		No		Yes		No

77. What is the difference between == and is?

Answer:

- == → compares values
- is → compares object identity (memory location)

78.What is the difference between shallow and deep copy?

Answer:

- **Shallow copy** → copies only top-level structure.
- **Deep copy** → copies entire structure recursively.

79. What are generators?

Answer:

Generators are functions that yield items one at a time using yield, allowing lazy evaluation.

```
def gen():
    for i in range(3):
        yield i
```

You can iterate over it:

```
for i in gen():
    print(i)
```

✖️ ✅ 80. What is the difference between mutable and immutable objects?

Mutable can change (list, dict, set); immutable cannot (tuple, str, int)

Small Programs

✖️ ✅ 81. Reverse a String

```
s = "hello"
```

```
print(s[::-1])
```

👉 Output: olleh

Alternate:

```
rev = ""
```

```
for ch in s:
```

```
    rev = ch + rev
```

```
print(rev)
```

◆ ✖️ ✅ 82. Check Palindrome

```
s = "madam"
```

```
if s == s[::-1]:
```

```
    print("Palindrome")
```

```
else:
```

```
    print("Not Palindrome")
```

✖️ ✅ 83. Reverse an Integer

```
num = 12345
```

```
rev = 0  
while num > 0:  
    rev = rev * 10 + num % 10  
    num //= 10  
print(rev)
```

84. Count Vowels in a String

```
s = "interview"  
vowels = "aeiouAEIOU"  
count = sum(1 for ch in s if ch in vowels)  
print("Vowel count:", count)
```

85. Find Factorial of a Number

```
n = 5  
fact = 1  
for i in range(1, n + 1):  
    fact *= i  
print(fact)  
  
Or using recursion:  
  
def fact(n):  
    return 1 if n == 0 else n * fact(n - 1)
```

87. Fibonacci Series

```
a, b = 0, 1  
for _ in range(10):  
    print(a, end=" ")  
    a, b = b, a + b
```

88. Check Prime Number

```
n = 13  
if n > 1:  
    for i in range(2, int(n ** 0.5) + 1):  
        if n % i == 0:  
            print("Not Prime")  
            break  
    else:  
        print("Prime")  
else:  
    print("Not Prime")
```

89 .Find Largest Element in a List

```
nums = [10, 25, 5, 80, 50]  
print(max(nums))
```

89.Count Occurrences of Each Character

```
s = "banana"  
freq = {}  
for ch in s:  
    freq[ch] = freq.get(ch, 0) + 1  
print(freq)
```

90. Check Armstrong Number

(Armstrong = sum of cubes of digits = number itself)

```
n = 153  
sum_ = sum(int(d) ** 3 for d in str(n))  
print("Armstrong" if sum_ == n else "Not Armstrong")
```

91.Swap Two Numbers Without Temp

```
a, b = 5, 10
```

```
a, b = b, a
```

```
print(a, b)
```

92.Find Second Largest in a List

```
nums = [10, 40, 30, 20]
```

```
nums.sort()
```

```
print("Second largest:", nums[-2])
```

93.Check Anagram

```
s1 = "listen"
```

```
s2 = "silent"
```

```
print(sorted(s1) == sorted(s2))
```

94.Find Missing Number in a List

```
nums = [1, 2, 3, 5]
```

```
n = len(nums) + 1
```

```
missing = n * (n + 1) // 2 - sum(nums)
```

```
print("Missing number:", missing)
```

95. Count Words in a String

```
s = "Python interview practice"
```

```
print(len(s.split()))
```

96. Find Even and Odd Numbers in a List

```
nums = [1, 2, 3, 4, 5, 6]
```

```
even = [x for x in nums if x % 2 == 0]
```

```
odd = [x for x in nums if x % 2 != 0]
```

```
print("Even:", even)
```

```
print("Odd:", odd)
```

97.Sum of Digits of a Number

```
n = 1234
```

```
print(sum(int(d) for d in str(n)))
```

98.Check String is Numeric

```
s = "12345"
```

```
print(s.isdigit())
```

99.Print Pattern (Star Pyramid)

```
rows = 5
```

```
for i in range(1, rows + 1):
```

```
    print("*" * i)
```

100.Remove Duplicates from List

```
nums = [1, 2, 2, 3, 4, 4, 5]
```

```
print(list(set(nums)))
```

101.Find the Largest of Three Numbers

```
a, b, c = 10, 25, 15
```

```
print(max(a, b, c))
```

102.Sort a List Without Using sort()

```
nums = [5, 2, 9, 1, 3]
```

```
for i in range(len(nums)):
```

```
    for j in range(i + 1, len(nums)):
```

```
        if nums[i] > nums[j]:
```

```
            nums[i], nums[j] = nums[j], nums[i]
```

```
print(nums)
```

◆ ✅ 103.Count Frequency of Words in a Sentence

```
s = "this is a test this is only a test"
```

```
words = s.split()
```

```
freq = {}
```

```
for w in words:
```

```
    freq[w] = freq.get(w, 0) + 1
```

```
print(freq)
```

✖ ✅ 104.Check if String is Pangram (contains all alphabets)

```
import string
```

```
s = "The quick brown fox jumps over the lazy dog"
```

```
print(set(string.ascii_lowercase).issubset(s.lower()))
```

◆ 25. Find All Prime Numbers up to N

```
n = 20
```

```
for i in range(2, n + 1):
```

```
    for j in range(2, int(i ** 0.5) + 1):
```

```
        if i % j == 0:
```

```
            break
```

```
    else:
```

```
        print(i, end=" ")
```

◆ ✅ 105.Find Common Elements in Two Lists

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

```
b = [4, 5, 6, 7]
```

```
print(list(set(a) & set(b)))
```

  **106. Find the Sum of All Even Numbers in a List**

```
nums = [1, 2, 3, 4, 5, 6]  
print(sum(x for x in nums if x % 2 == 0))
```

  **107. Check if Two Strings are Rotation of Each Other**

```
s1 = "abcde"  
s2 = "deabc"  
print(len(s1) == len(s2) and s2 in s1 + s1)
```

  **108. Print Duplicate Elements from a List**

```
nums = [1, 2, 3, 2, 4, 1, 5]  
dups = [x for x in set(nums) if nums.count(x) > 1]  
print(dups)
```

  **109. Check if a Number is Perfect**

(A perfect number = sum of its divisors = number itself)

```
n = 28  
sum_ = sum(i for i in range(1, n) if n % i == 0)  
print("Perfect" if sum_ == n else "Not Perfect")
```

  **110. Find GCD (Greatest Common Divisor)**

```
import math  
print(math.gcd(12, 18))  
  
Or manually:  
  
a, b = 12, 18  
while b:  
    a, b = b, a % b  
print(a)
```

  **111.Find LCM (Least Common Multiple)**

```
import math  
a, b = 12, 18  
print(a * b // math.gcd(a, b))
```

  **112.Remove Special Characters from a String**

```
import re  
s = "Hello@World#2025!"  
print(re.sub(r'^A-Za-z0-9 ]+', "", s))
```

  **113.Check if Number is Power of 2**

```
n = 16  
print(n > 0 and (n & (n - 1)) == 0)
```

  **114. Find All Substrings of a String**

```
s = "abc"  
for i in range(len(s)):  
    for j in range(i + 1, len(s) + 1):  
        print(s[i:j])
```

  **115. Find Second Minimum Number in List**

```
nums = [10, 5, 8, 20, 3]  
nums = list(set(nums))  
nums.sort()  
print("Second smallest:", nums[1])
```

  **116. Merge Two Dictionaries**

```
d1 = {'a': 1, 'b': 2}  
d2 = {'c': 3, 'd': 4}
```

```
merged = {**d1, **d2}
```

```
print(merged)
```

117. Find First Non-Repeating Character

```
s = "aabbcde"
```

```
for ch in s:
```

```
    if s.count(ch) == 1:
```

```
        print(ch)
```

```
        break
```

118. Check Leap Year

```
year = 2024
```

```
if (year % 400 == 0) or (year % 100 != 0 and year % 4 == 0):
```

```
    print("Leap year")
```

```
else:
```

```
    print("Not leap year")
```

119. Flatten a Nested List

```
nested = [[1, 2], [3, 4], [5, 6]]
```

```
flat = [num for sublist in nested for num in sublist]
```

```
print(flat)
```

120 . Count uppercase, lowercase, digits, special characters

Input: "HeLLo123@"

Output: Upper=3 Lower=2 Digits=3 Special=1

```
s = "HeLLo123@"
```

```
upper = lower = digit = special = 0
```

```
for ch in s:
```

```
    if ch.isupper():
```

```
upper += 1  
elif ch.islower():  
    lower += 1  
elif ch.isdigit():  
    digit += 1  
else:  
    special += 1
```

✓ 121. Find second largest number

Input: [10, 20, 5, 30, 25]

Output: 25

```
nums = [10, 20, 5, 30, 25]  
unique = list(set(nums))  
unique.sort()  
print(unique[-2])
```

✓ 123. Move all zeros to end

Input: [0,1,0,3,12]

Output: [1,3,12,0,0]

```
arr = [0,1,0,3,12]  
result = [x for x in arr if x != 0] + [0]*arr.count(0)  
print(result)
```

✓ 124. Print characters that appear more than once

Input: "programming"

Output: r, g, m

```
s = "programming"  
for ch in set(s):  
    if s.count(ch) > 1:  
        print(ch, end=" ")
```

✓ 125. Longest word in a sentence

Input: "Python is super powerful"

Output: powerful

```
s = "Python is super powerful"
```

```
print(max(s.split(), key=len))
```

✓ 126. Sum of digits in a number

Input: n = 987

Output: 24

```
n = 987
```

```
print(sum(int(d) for d in str(n)))
```

✓ 127. Remove all vowels from string

Input: "beautiful"

Output: "btfl"

```
s = "beautiful"
```

```
vowels = "aeiouAEIOU"
```

```
print("".join(ch for ch in s if ch not in vowels))
```

✓ 128. Find common elements in two lists

Input: [1,2,3,4] and [3,4,5,6]

Output: [3,4]

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

```
b = [3,4,5,6]
```

```
print(list(set(a) & set(b)))
```

✓ 129. Remove specific character

Input: "banana", remove "a"

Output: "bnn"

```
s = "banana"
```

```
remove = "a"
```

```
print(s.replace(remove, ""))
```

 **130. Check if substring exists**

Input: "hello", "ell"

Output: True

```
s = "hello"
```

```
sub = "ell"
```

```
print(sub in s)
```

 **131. Convert list of strings to integers**

Input: ["1","2","3"]

Output: [1,2,3]

```
lst = ["1","2","3"]
```

```
print(list(map(int, lst)))
```

 **132. Calculate factorial (loop)**

Input: 5

Output: 120

```
n = 5
```

```
fact = 1
```

```
for i in range(1, n+1):
```

```
    fact *= i
```

```
print(fact)
```

 **133. Check if list is sorted**

Input: [1,2,3,4]

Output: True

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

```
print(lst == sorted(lst))
```

 **134. Convert list to comma-separated string**

Input: ['a', 'b', 'c']

Output: "a,b,c"

```
lst = ['a','b','c']
```

```
print(",".join(lst))
```

✓ 135. Find index of element (manual search)

Input: [10,20,30], target=20

Output: 1

```
arr = [10,20,30]
```

```
target = 20
```

```
for i, v in enumerate(arr):
```

```
    if v == target:
```

```
        print(i)
```

```
        break
```

✓ 136. Reverse words but keep order

Input: "I love Python"

Output: "I evol nohtyP"

```
s = "I love Python"
```

```
print(" ".join(word[::-1] for word in s.split()))
```

✓ 137. Find character with highest frequency

Input: "mississippi"

Output: i

```
s = "mississippi"
```

```
freq = {}
```

```
for ch in s:
```

```
    freq[ch] = freq.get(ch, 0) + 1
```

```
print(max(freq, key=freq.get))
```

✓ 138. Remove all duplicates characters

Input: "engineering"

Output: "enginr"

```
s = "engineering"
```

```
res = ""
```

```
for ch in s:
```

```
    if ch not in res:
```

```
        res += ch
```

```
print(res)
```

139. Replace spaces with hyphens

Input: "a b c d"

Output: "a-b-c-d"

```
s = "a b c d"
```

```
print(s.replace(" ", "-"))
```

140. Print unique elements from list

Input: [1,2,2,3,4,4,5]

Output: [1,3,5]

```
lst = [1,2,2,3,4,4,5]
```

```
print([x for x in lst if lst.count(x) == 1])
```

141. Sum only even numbers

Input: [1,2,3,4,5]

Output: 6

```
lst = [1,2,3,4,5]
```

```
print(sum(x for x in lst if x % 2 == 0))
```

142. Count frequency of each word (case insensitive)

Input: "Hello hello HELLO world"

Output: {'hello': 3, 'world':1}

```
s = "Hello hello HELLO world".lower()
```

```
freq = {}
```

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

143. Remove punctuation from string

Input: "hi!!! hello???"

Output: "hi hello"

```
import string
s = "hi!!! hello???"
for p in string.punctuation:
    s = s.replace(p, "")
print(s)
```

144. Get common characters between two strings

Input: "apple", "grapes"

Output: ['a','p','e']

```
s1 = "apple"
s2 = "grapes"
print(list(set(s1) & set(s2)))
```

145. Check if number is Armstrong

Input: 153

Output: True

n = 153

```
digits = list(map(int, str(n)))
print(sum(d**len(digits) for d in digits) == n)
```

146. Merge dictionaries (add values if keys repeat)

Input: {"a":5,"b":2}, {"a":3,"c":4}

Output: {"a":8,"b":2,"c":4}

```
d1 = {"a":5,"b":2}
```

```
d2 = {"a":3,"c":4}
```

```
out = d1.copy()
```

```
for k,v in d2.items():
```

```
    out[k] = out.get(k, 0) + v
```

```
print(out)
```

146. Print only digits from a string

Input: "ab12cd34"

Output: 1234

```
s = "ab12cd34"
```

```
print("".join(ch for ch in s if ch.isdigit()))
```

147. Sort dictionary by values

Input: {"a":3,"b":1,"c":2}

Output: [('b',1), ('c',2), ('a',3)]

```
d = {"a":3,"b":1,"c":2}
```

```
print(sorted(d.items(), key=lambda x: x[1]))
```

148. Find missing number in list (1 to n)

Input: [1,2,4,5]

Output: 3

```
lst = [1,2,4,5]
```

```
n = len(lst) + 1
```

```
print(n*(n+1)//2 - sum(lst))
```

149. Remove None values from list

Input: [1,None,2,None,3]

Output: [1,2,3]

```
lst = [1,None,2,None,3]
```

```
print([x for x in lst if x is not None])
```

150. Reverse integer

Input: 1234

Output: 4321

n = 1234

```
print(int(str(n)[::-1]))
```

151. Check if two lists are identical (order matters)

Input: [1,2,3] and [1,2,3]

Output: True

a = [1,2,3]

b = [1,2,3]

```
print(a == b)
```

152. Print duplicate numbers

Input: [4,5,6,4,7,5]

Output: 4 5

lst = [4,5,6,4,7,5]

```
print([x for x in set(lst) if lst.count(x) > 1])
```

153.. Get max & min without built-in functions

Input: [3,1,4,2]

Output: Max=4 Min=1

lst = [3,1,4,2]

mx = mn = lst[0]

for x in lst:

 if x > mx: mx = x

 if x < mn: mn = x

```
print(mx, mn)
```

154. Count words that start with vowel

Input: "apple is an awesome fruit"

Output: 3s

```
s = "apple is an awesome fruit"
```

```
vowels = "aeiou"
```

```
print(sum(1 for w in s.split() if w[0].lower() in vowels))
```

Big Query Interview Questions and Answers

1: rank each order by amount with in the same country using rankdency rank, row number

Customers

customer_id	first_name	last_name	age	country
1	John	Doe	31	USA
2	Robert	Luna	22	USA
3	David	Robinson	22	UK
4	John	Reinhardt	25	UK
5	Betty	Doe	28	UAE

Orders

order_id	item	amount	customer_id
1	Keyboard	400	4
2	Mouse	300	4
3	Monitor	12000	3
4	Keyboard	400	1
5	Mousepad	250	2

Ans . select c.country,
 o.order_id,
 o_items,
 o.amount,
 c.first_name ,
 rank() over(partition by c.country order by o.amount desc) as amount_rank from
 order O
 join Customersc
 ON o.customer_id = c.customer_id 4order by c.country
 amount_rank

✓✓ 2: each order show the previous and the next order amount

select
 c.country,
 o.order_id,
 o.item,
 o.amount,
 lead(o.amount) over(partition by c.country order by o.amount) as next_order_amount
 from Orders o
 join Customers c
 on o.customer_id = c.customer_id
 order by c.country,
 o.amount

✓✓ 3: Questions:-

ID	NAME	GENDE R
1	Alice	Female
2	Bob	Male
3	Carol	Female
4	David	Male
5	Eve	Female

Output :-

Female	60.00%
Male	40.00%

Query

```
select gender,round((count(*) * 100.0 /  
select count(*) from emp )),2) as percentage  
from emp  
group by gender
```