

Crack Zoho IT Interviews: Complete Q&A Guide for Freshers & Experienced

1. Aptitude & Logical Reasoning

◆ Q1. Time & Work

Question:

If A can complete a work in 12 days and B can do it in 8 days, how long will it take if both work together?

Answer:

A's 1 day work = $1/12$

B's 1 day work = $1/8$

Together = $1/12 + 1/8 = (2 + 3) / 24 = 5/24$

▶ Total days = $24 / 5 = 4.8$ days

✦ **Tip:** Use the "1-day work method" for all Time & Work problems.

◆ Q2. Profit & Loss

Question:

A shopkeeper bought a pen for ₹60 and sold it for ₹75. Find the profit %.

Answer:

Profit = $75 - 60 = ₹15$

Profit% = $(15 / 60) \times 100 = 25\%$

✓ **Formula:**

Profit% = $(\text{Profit} / \text{Cost Price}) \times 100 =$

◆ Q3. Number Series (Reasoning)

Question:

Find the next number in the series: 2, 6, 12, 20, 30, ?

Answer:

Difference pattern:

+4, +6, +8, +10...

Last term = 30

Next difference = +12

Answer = $30 + 12 = 42$

◆ Q4. Percentage

Question:

What is 25% of 640?

Answer:

$25\% = 25/100 = 1/4$

➡ $640 \times 1/4 = 160$



◆ Q5. Time & Distance

Question:

A train 150 m long crosses a man in 15 seconds. Find the speed of the train in km/h.

Answer:

Speed = Distance / Time = $150 / 15 = 10$ m/s

Convert to km/h $\rightarrow 10 \times 18/5 = 36$ km/h

✦ Conversion Tip:

m/s $\rightarrow$ km/h: multiply by $18/5$

km/h $\rightarrow$ m/s: multiply by $5/18$

◆ Q6. Ratio & Proportion

Question:

If $A:B = 3:4$ and $B:C = 2:5$, then what is $A:C$?

Answer:

$$A:B = 3:4$$

$$B:C = 2:5 \rightarrow \text{Make B same in both} \rightarrow$$

$$\text{LCM of 4 and 2} = 4$$

$$A:B = 3:4$$

$$B:C = 4:10$$

$$\text{Now } A:C = 3:10$$

◆ Q7. Ages Problem

Question:

Father is 4 times the age of his son. After 5 years, the father will be 3 times the son's age. Find their present ages.

Answer:

$$\text{Let son's age} = x$$

$$\text{Then father} = 4x$$

After 5 years:

$$\text{Father} = 4x + 5, \text{ Son} = x + 5$$

$$\text{So, } 4x + 5 = 3(x + 5)$$

$$\Rightarrow 4x + 5 = 3x + 15$$

$$\Rightarrow x = 10$$

$$\text{Son} = 10, \text{ Father} = 40$$

◆ Q8. Logical Puzzle

Question:

If MONDAY = 123456, then what is the code for DAMN?

Answer:

Match letters:

M = 1, O = 2, N = 3, D = 4, A = 5, Y = 6

So DAMN = 4 (D), 5 (A), 1 (M), 3 (N) → 4513

◆ Q9. Calendar Question

Question:

If 1st Jan 2025 is a Wednesday, what day is 1st March 2025?

Answer:

Jan = 31 days → $31 \% 7 = 3$ days

Feb = 28 days (2025 is not a leap year) → $28 \% 7 = 0$ days

Total days from Jan 1 to March 1 = $3 + 0 = 3$

➡ 1st Jan is Wednesday → +3 days = Saturday

◆ Q10. Blood Relation

Question:

Pointing to a man, Rani said, "He is the son of my grandfather's only son." Who is the man to Rani?

Answer:

"Grandfather's only son" = Rani's father

So the man is Rani's father's son = Brother

✅ Answer: Brother

2.Programming Basics (C / C++ / Java / Python)

◆ Q1. What is the difference between C and C++?

Feature	C	C++
Paradigm	Procedural	Procedural + Object-Oriented
Encapsulation	✗ Not supported	✓ Supported via classes
Inheritance, OOP	✗ Not available	✓ Available
Function Overloading	✗ No	✓ Yes
Use Case	OS, embedded	Software, games, apps

✦ Tip: Zoho expects you to know the strengths and limitations of each language.

◆ Q2. Difference between == and .equals() in Java?

- == compares object references (memory location).
- .equals() compares object content/value.

```
String a = new String("zoho");
```

```
String b = new String("zoho");
```

```
System.out.println(a == b);    // false
```

```
System.out.println(a.equals(b)); // true
```

✓ Answer: == checks memory, .equals() checks value.

◆ Q3. What is a pointer in C?

A pointer is a variable that stores the address of another variable.

```
int a = 10;
```

```
int *p = &a;
```

```
printf("%d", *p); // Output: 10
```

✔ Use cases: dynamic memory allocation, function arguments, arrays.

◆ **Q4. Write a C program to check if a number is a palindrome**

```
#include <stdio.h>
```

```
int main() {
```

```
    int num, reversed = 0, temp, rem;
```

```
    printf("Enter number: ");
```

```
    scanf("%d", &num);
```

```
    temp = num;
```

```
    while (temp > 0) {
```

```
        rem = temp % 10;
```

```
        reversed = (reversed * 10) + rem;
```

```
        temp /= 10;
```

```
    }
```

```
    if (reversed == num)
```

```
        printf("Palindrome");
```

```
    else
```

```
        printf("Not Palindrome");
```

```
    return 0;
```

```
}
```

◆ **Q5. Explain OOPs concepts with real examples (Java / Python)**

OOPs 4 Pillars:

1. **Encapsulation – Wrapping data & code into one unit**
✦ Example: Class with private variables and public getters/setters
2. **Abstraction – Hiding implementation details**
✦ Example: Java Interface or Python abstract class
3. **Inheritance – One class derives from another**
✦ Example:
4. java
5. CopyEdit
6. class Animal { void sound() {} }
7. class Dog extends Animal { void sound() { System.out.println("Bark"); } }
8. **Polymorphism – One method, different behavior**
✦ Example: Function overloading or overriding

◆ Q6. Write a Python program to check if a string is a palindrome

```
def is_palindrome(s):
```

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

```
print(is_palindrome("level")) # Output: True
```

✓ Python's slicing makes this logic simple.

◆ Q7. What is the difference between List and Array in Python?

Feature	List	Array (from array module)
Data types	Mixed allowed	Only same type
Built-in	Yes ([])	Needs from array import array
Flexibility	High	Medium

✦ In interviews, prefer list unless you need performance + fixed-type arrays.

Q8. Write a C++ program to swap two numbers using a function

```
#include<iostream>

using namespace std;

void swap(int &a, int &b) {

    int temp = a;

    a = b;

    b = temp;

}

int main() {

    int x = 10, y = 20;

    swap(x, y);

    cout << x << " " << y; // Output: 20 10

}
```

✓ **Note:** Reference variable & allows swapping without returning.

◆ Q9. Difference between Stack and Heap memory

Feature	Stack	Heap
Usage	Function calls, local vars	Dynamic allocation memory
Size	Limited	Large
Allocation	Auto-managed	Manual (malloc, new)
Speed	Faster	Slower

✦ **C/C++** need manual free() or delete. **Java/Python** has garbage collection.

◆ **Q10. Write a program to print Fibonacci series (Python)**

```
def fibonacci(n):  
    a, b = 0, 1  
    for _ in range(n):  
        print(a, end=" ")  
        a, b = b, a + b  
fibonacci(5)
```

➡ **Output:** 0 1 1 2 3

3. Data Structures & Algorithms

◆ **Q1. What is an Array?**

An array is a collection of elements stored at contiguous memory locations.

- Fixed size, same data type
- Fast access using index ($O(1)$)

✚ In C:

```
int arr[5] = {1, 2, 3, 4, 5};  
printf("%d", arr[2]);
```

◆ **Q2. What is a Linked List? How is it different from an array?**

A Linked List is a linear data structure where each node points to the next.

Feature	Array	Linked List
Memory	Contiguous	Dynamic (non-contiguous)
Insertion/Deletion	Costly	Easy (just change pointers)
Access	Fast ($O(1)$)	Slower ($O(n)$)
Size	Fixed	Dynamic

💡 Common in real-world use cases like implementing queues, stacks.

◆ Q3. Write a program to reverse a linked list in C++

```
Node* reverse(Node* head) {
```

```
    Node* prev = NULL;
```

```
    Node* curr = head;
```

```
    Node* next = NULL;
```

```
    while (curr != NULL) {
```

```
        next = curr->next;
```

```
        curr->next = prev;
```

```
        prev = curr;
```

```
        curr = next;
```

```
    }
```

```
    return prev;
```

```
}
```

✅ Asked frequently in Zoho coding rounds.

◆ **Q4. What is Stack? Give a real-time example.**

A stack follows LIFO (Last In First Out).

◆ Use cases:

- Undo operations
- Recursion
- Browser back button

```
stack = []  
  
stack.append(10)  
  
stack.append(20)  
  
print(stack.pop())
```

◆ **Q5. Difference between Stack and Queue**

Property	Stack (LIFO)	Queue (FIFO)
Insert/Remove	Top	Front/Rear
Use case	Undo, Backtracking	Scheduling, Queues

In Python:

```
from collections import deque  
  
queue = deque([1,2])  
  
queue.append(3)  
  
print(queue.popleft())
```

◆ **Q6. Write a program to check if a string is a palindrome using stack**

```
def is_palindrome(s):  
  
    return s == s[::-1]
```

```
print(is_palindrome("madam"))
```

◆ Q7. Explain Binary Search with Python code

Used to search in a sorted array. Reduces time complexity to $O(\log n)$.

```
def binary_search(arr, key):
```

```
    low, high = 0, len(arr) - 1
```

```
    while low <= high:
```

```
        mid = (low + high) // 2
```

```
        if arr[mid] == key:
```

```
            return mid
```

```
        elif arr[mid] < key:
```

```
            low = mid + 1
```

```
        else:
```

```
            high = mid - 1
```

```
    return -1
```

✓ Best case: $O(1)$

🕒 Worst case: $O(\log n)$



◆ Q8. What is Recursion? Example: Factorial

Recursion is when a function calls itself.

```
def fact(n):
```

```
    if n == 0: return 1
```

```
    return n * fact(n - 1)
```

```
print(fact(5)) # Output: 120
```

🔴 Use base case to stop infinite loop!

◆ **Q9. Write a program to find the largest element in an array**

```
arr = [5, 1, 9, 3]

max_val = arr[0]

for i in arr:

    if i > max_val:

        max_val = i

print(max_val)
```

◆ **Q10. Zoho-style Coding Question: Digit Sum**

Q: Given a number, keep summing its digits until you get a single digit.

Example: 987 → 9+8+7 = 24 → 2+4 = 6

```
def digit_sum(n):

    while n >= 10:

        n = sum(int(d) for d in str(n))

    return n

print(digit_sum(987))
```



4. SQL & Database (Zoho Interview Level)

◆ **Q1. What is a Primary Key?**

A Primary Key uniquely identifies each record in a table.

- Cannot be NULL
- Must be unique
- One primary key per table

✚ **Example:**

```
CREATE TABLE Employee (

    emp_id INT PRIMARY KEY,
```

```
name VARCHAR(50)
);
```

◆ **Q2. What is the difference between WHERE and HAVING?**

Clause	Used With	Works On
WHERE	SELECT, UPDATE, etc	Rows
HAVING	GROUP BY	Aggregated data

◆ **Example:**

-- WHERE filters rows before grouping

```
SELECT * FROM employees WHERE dept = 'IT';
```

-- HAVING filters groups after aggregation

```
SELECT dept, COUNT(*)
```

```
FROM employees
```

```
GROUP BY dept
```

```
HAVING COUNT(*) > 5;
```

◆ **Q3. Write an SQL query to find the second highest salary from a table**

```
SELECT MAX(salary)
```

```
FROM employee
```

```
WHERE salary < (
```

```
    SELECT MAX(salary) FROM employee
```

```
);
```

✓ Or use LIMIT (MySQL):

sql

CopyEdit

SELECT DISTINCT salary

FROM employee

ORDER BY salary DESC

LIMIT 1 OFFSET 1;

◆ Q4. What are Joins? Types?

Joins combine data from multiple tables based on a related column.

Join Type	Description
INNER JOIN	Matches rows in both tables
LEFT JOIN	All from left + matching from right
RIGHT JOIN	All from right + matching from left
FULL JOIN	All rows from both sides (if supported)

✦ Example:

SELECT emp.name, dept.name

FROM emp

JOIN dept ON emp.dept_id = dept.id;

◆ Q5. What is Normalization?

Normalization is the process of organizing data to reduce redundancy.

Forms:

- **1NF: No repeating groups**
- **2NF: No partial dependency (for composite PK)**
- **3NF: No transitive dependency**

✔ Keeps the database clean and efficient.

◆ **Q6. Find the employee with the highest salary in each department**

SELECT dept_id, MAX(salary) AS max_salary

FROM employee

GROUP BY dept_id;

✦ For full employee details, use correlated subquery or JOIN with MAX

◆ **Q7. What is the difference between DELETE, TRUNCATE, and DROP?**

Command	Description	Rollback Possible?
DELETE	Removes rows (with condition)	✔ Yes
TRUNCATE	Removes all rows (fast)	✗ No
DROP	Deletes entire table structure	✗ No

◆ **Q8. Write a query to display the count of employees in each department**

SELECT dept_id, COUNT(*) AS emp_count

FROM employee

GROUP BY dept_id;

◆ **Q9. What is a Foreign Key?**

A Foreign Key links a column in one table to the Primary Key in another table.

✦ Example:

CREATE TABLE orders (

order_id INT PRIMARY KEY,

```
emp_id INT,  
FOREIGN KEY (emp_id) REFERENCES employee(emp_id)  
);
```

◆ Q10. Real-time Zoho Interview Task

"Write a query to find employees who joined in the last 30 days"

```
SELECT name, join_date  
FROM employee  
WHERE join_date >= CURDATE() - INTERVAL 30 DAY;
```

5. OS, OOPs, Networking

◆ Q1. What is a Process vs Thread?

Aspect	Process	Thread
Definition	Independent program in execution	Lightweight sub-part of a process
Memory	Separate memory space	Shares memory of parent process
Overhead	High	Low
Example	MS Word, Chrome tab	Single tab in browser

✦ Zoho often asks: "What happens when you open multiple tabs?"

◆ Q2. What is Deadlock?

A deadlock occurs when 2 or more processes wait forever for each other to release a resource.

✦ Example (classic):

- Process A holds Resource 1, needs Resource 2
- Process B holds Resource 2, needs Resource 1

Result: ⚠ Both are stuck = **Deadlock!**

✓ Prevent using resource hierarchy or timeout.

◆ Q3. What is Context Switching?

Switching CPU from one process/thread to another.

- Saves current state (PCB)
- Loads next process state

✦ Measured in nanoseconds; affects performance

◆ Q4. What is Scheduling? What is Round Robin?

Scheduling decides the order in which processes run on the CPU.

🕒 Round Robin:

- Each process gets fixed time slice (quantum)
- After time ends, goes to end of queue
- Fair for all, used in time-sharing OS

◆ Q5. Difference between Mutex and Semaphore

Feature	Mutex	Semaphore
Lock Type	Binary (locked/unlocked)	Counting (0,1,2...)
Ownership	Only the thread can unlock it	Anyone can signal
Use Case	Single resource access	Multiple resource limits

🧱 OBJECT-ORIENTED PROGRAMMING (OOP)

◆ Q6. What are the 4 Pillars of OOP?

1. Encapsulation – Wrapping data + methods into one unit (class)
2. Abstraction – Hiding internal logic, showing only essentials
3. Inheritance – One class derives from another
4. Polymorphism – One name, many forms (method overloading/overriding)

◆ Q7. Real-world examples of OOPs

OOP Concept	Real Life Example
Encapsulation	ATM → data (balance) + function (withdraw)
Abstraction	Car → press start, engine logic hidden
Inheritance	Dog → Animal → inherits speak()
Polymorphism	print() → different behaviors for int, str, float

◆ Q8. What is Constructor Overloading in C++?

Multiple constructors in a class with different parameters.

```
class Box {  
  
    public:  
  
        Box() { cout << "Default\n"; }  
  
        Box(int l) { cout << "Length: " << l; }  
  
};
```

◆ **Q9. What is the difference between Abstract Class and Interface?**

Feature	Abstract Class	Interface
Methods	Can have both defined/undefined	All methods undefined (Java 7)
Variables	Can have variables	Only constants
Inheritance	Supports single inheritance	Supports multiple inheritance

 NETWORKING

◆ **Q10. What is the difference between TCP and UDP?**

Protocol	TCP	UDP
Type	Connection-oriented	Connection-less
Reliability	Reliable (acknowledgment)	Unreliable (no ack)
Speed	Slower	Faster
Use Case	File transfer, emails	Streaming, gaming

◆ **Q11. What is DNS?**

DNS = Domain Name System

Converts domain names into IP addresses.

◆ **Example:**

www.zoho.com → 103.116.220.55

◆ **Q12. Difference between HTTP and HTTPS?**

Protocol	HTTP	HTTPS
Full Form	HyperText Transfer Protocol	HTTP Secure
Security	✗ No encryption	✓ SSL encryption
Port	80	443

◆ Q13. What happens when you type a URL?

Flow:

1. Browser checks cache → DNS lookup
2. TCP handshake
3. HTTP request sent to server
4. Server processes and sends response
5. Browser renders HTML

✓ Be ready to explain this with simple steps!

6. Zoho Coding Round – Real Questions (with Solutions)

◆ Q1. Pattern Problem

Print the following pattern for N = 4:

```

1
2 3
4 5 6
7 8 9 10

```

Logic:

- Use a counter starting from 1.
- For each row, print i elements.

n = 4

num = 1

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

for j in range(i):

print(num, end=' ')

num += 1

print()

✓ Pattern questions are compulsory in Zoho's basic rounds.

◆ Q2. String Compression

Input: "aaabbcccaaa"

Output: "a3b2c3a3"

Explanation: Consecutive characters are counted and compressed.

def compress(s):

result = ""

count = 1

for i in range(1, len(s)):

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

count += 1

else:

result += s[i - 1] + str(count)

count = 1

result += s[-1] + str(count)

return result

print(compress("aaabbcccaaa"))

◆ Q3. Digit Sum Until One Digit

Input: 987 → 9 + 8 + 7 = 24 → 2 + 4 = 6

```
def digit_sum(n):
```

```
    while n >= 10:
```

```
        n = sum(int(i) for i in str(n))
```

```
    return n
```

```
print(digit_sum(987)) # Output: 6
```

✓ This is a common round 1 filter question.

◆ Q4. Find the Most Frequent Character in a String

Input: "zohozoho" → Output: {'z': 2, 'o': 2, 'h': 2}

```
from collections import Counter
```

```
s = "zohozoho"
```

```
freq = Counter(s)
```

```
max_freq = max(freq.values())
```

```
for char in freq:
```

```
    if freq[char] == max_freq:
```

```
        print(f"{char}: {freq[char]}")
```

◆ Q5. Check if Two Strings are Anagrams

Input: "listen", "silent" → ✓ Yes

Logic: Sort both strings and compare.

```
def is_anagram(a, b):
```

```
    return sorted(a) == sorted(b)
```

```
print(is_anagram("listen", "silent")) # True
```

◆ Q6. Print Numbers in Words

Input: 512 → Output: "Five One Two"

```
digits = ["Zero", "One", "Two", "Three", "Four", "Five",  
          "Six", "Seven", "Eight", "Nine"]
```

```
def number_to_words(n):
```

```
    return " ".join(digits[int(i)] for i in str(n))
```

```
print(number_to_words(512))
```

◆ Q7. Remove Duplicates from a String

Input: "engineering" → "engir"

```
def remove_duplicates(s):
```

```
    result = ""
```

```
    for c in s:
```

```
        if c not in result:
```

```
            result += c
```

```
    return result
```

```
print(remove_duplicates("engineering"))
```

◆ Q8. Check for Balanced Parentheses

Input: {[()]} →  Balanced

Logic: Use a stack

```
def is_balanced(s):
```

```
    stack = []
```

```
    pairs = {'(': ')', '[': ']', '{': '}'}
```

```

for c in s:
    if c in "([{":
        stack.append(c)
    elif c in ")]}":
        if not stack or stack[-1] != pairs[c]:
            return False
        stack.pop()
return not stack

print(is_balanced("[[()]]")) # True

```

◆ Q9. Sort an Array Without Using Built-in Sort

```

def bubble_sort(arr):
    n = len(arr)
    for i in range(n):
        for j in range(n - i - 1):
            if arr[j] > arr[j + 1]:
                arr[j], arr[j + 1] = arr[j + 1], arr[j]
    return arr

print(bubble_sort([5, 2, 1, 8, 3]))

```

◆ Q10. Zoho Round 2 – Project-Based Coding Scenario

Task Example:

"Design a simple Student Management System that stores name, id, and mark. Print students with mark > 90."

class Student:

```

def __init__(self, name, id, mark):
    self.name = name

```

```
self.id = id

self.mark = mark

students = [

    Student("A", 101, 95),

    Student("B", 102, 87),

    Student("C", 103, 91)

]

for s in students:

    if s.mark > 90:

        print(s.name, s.mark)
```

✦ They may modify the question mid-way like:

- “Add a function to update mark”
- “Sort students by mark descending”

7. Resume + HR Round Prep

RESUME TIPS FOR ZOHO INTERVIEWS

✓ Must-Have Sections:

- Name, contact, LinkedIn/GitHub (if available)
- Career Objective (2 lines MAX)
- Skills (languages, tools, OS)
- Projects (1 major + 1 mini)
- Education + Certifications (with year and percentage)
- Achievements / Internships (if any)

✦ Guidance:

- Don't lie. HR/Tech will ask project-based Qs.
- Avoid "hardworking, fast learner" cliché words.
- Use action verbs: Developed, Designed, Led, Built...

COMMON HR QUESTIONS + BEST SAMPLE ANSWERS

◆ Q1. Tell me about yourself

Format:

- **Start with name + education**
- **Add strengths + project/tools**
- **End with why you're excited for the opportunity**

✓ Example:

I'm Divya, a Computer Science graduate from Anna University. I've worked on a student project that uses Python and SQL to manage hostel records. I enjoy solving logic-based problems and am constantly improving my coding skills. I'm looking forward to starting my career with a company like Zoho where I can apply my skills and grow.

◆ Q2. Why should we hire you?

Sample Answer:

I am confident in my problem-solving ability and I adapt quickly to new technologies. I've practiced Zoho-style coding problems, and I enjoy working in challenging environments. I'm also consistent and reliable in team settings.

◆ Q3. Why Zoho?

Tips:

- **Mention: Product-based company, learning culture, Indian tech pride**

✓ Example:

Zoho is one of the very few companies that builds world-class software fully in India. I've used Zoho Mail and Zoho CRM before and admire the design quality. I'm excited about working with teams that build global products right here.

◆ Q4. What are your strengths & weaknesses?

✓ Strength:

Logical thinking, team collaboration, and consistent focus on tasks.

⊘ Weakness (with recovery):

I used to struggle with speaking up in meetings, but I've been improving by participating in team presentations during college.

◆ Q5. Describe your final year project

Checklist:

- Problem statement
- Tools/language used
- Your contribution (most important)
- Output/result

✓ **Sample:**

My final year project was “Leave Management System” using PHP and MySQL. I handled the login, dashboard logic, and status flow. We used HTML/CSS for frontend and hosted it on XAMPP locally. The goal was to automate the leave process for faculty in my college.

◆ **Q6. Are you open to relocation or working in Madurai/Salem?**

✓ **Best Answer:**

Yes, I’m open to relocate. I understand that Zoho has development centers in multiple cities, and I’m happy to work wherever I’m placed.

◆ **Q7. How do you handle pressure or tight deadlines?**

✓ **Sample Answer:**

I plan tasks into smaller milestones, so even when deadlines are short, I know how to stay focused. I’ve handled multiple submissions during college and managed time well by prioritizing key work.

◆ **Q8. Do you have any questions for us?**

⊘ **Don’t say: “No, all good.”**

✓ **Ask one polite question:**

“Yes, I’d love to know more about the type of projects freshers get to work on initially.”

8. MCQs + Final Cheat Sheet (Quick Revision)

◆ SECTION A: TECHNICAL MCQs (Mixed)

Q1. Which of the following is not a valid OOP concept?

- A. Inheritance
- B. Encapsulation
- C. Compilation
- D. Polymorphism

✓ **Answer: C**

Q2. What is the default return type of main() in C?

- A. void
- B. int
- C. float
- D. char

✓ Answer: B

Q3. Which SQL clause is used to filter aggregated data?

- A. WHERE
- B. GROUP BY
- C. HAVING
- D. ORDER BY

✓ Answer: C

Q4. In C, what is the size of int on a 32-bit compiler?

- A. 8 bytes
- B. 2 bytes
- C. 4 bytes
- D. Depends on OS

✓ Answer: C

Q5. Which sorting algorithm has the best average-case time complexity?

- A. Bubble Sort
- B. Merge Sort
- C. Selection Sort
- D. Insertion Sort

✓ Answer: B

Q6. What will strcpy() do in C?

- A. Compare two strings
- B. Copy one string into another
- C. Concatenate strings
- D. None of the above

✓ Answer: B

Q7. In Python, what does list[::-1] do?

- A. Deletes list
- B. Reverses list
- C. Returns only even values
- D. Error

✓ Answer: B

Q8. Which layer in OSI model is responsible for routing?

- A. Data Link Layer
- B. Session Layer

- C. Network Layer
- D. Transport Layer

✓ Answer: C

◆ SECTION B: QUICK FORMULAS & SHORTCUTS

🧠 Aptitude:

- $\% \rightarrow (\text{Part} / \text{Whole}) \times 100$
- $\text{Speed} = \text{Distance} / \text{Time}$
- $\text{Compound Interest} = P(1 + r/100)^n$
- $\text{Ratio } A : B \rightarrow A / (A + B)$

🧠 C Programming:

- `int arr[5];` → Valid index: 0 to 4
- `scanf("%d", &var);` → Don't forget &
- `void main()` ✗ → Use `int main()` ✓

🧠 SQL:

- `SELECT COUNT(*) FROM table;`
- `GROUP BY` → Always used with aggregation
- `HAVING` → Used after `GROUP BY`

🧠 OOPs:

- **Encapsulation:** class + private variables
- **Abstraction:** Hide complexity, show usage
- **Inheritance:** `class B : public A`
- **Polymorphism:** Function overloading, overriding

◆ SECTION C: LAST-MINUTE CHECKLIST ✓

 Final Touchup	Status
Practiced 10+ pattern programs	
Learned 2 DSA concepts daily	
Revised DB joins + group by	
Prepared self-intro	
Final year project – revised	
HR questions practiced	
Resume is clean & real	
Know basic Linux/OS terms	
Done 5+ mock interviews (or self)	

Made with ❤ by careers.tn_2.0