Real-Time Interview Questions – Structured Guide

Section 1: Programming & Logic (Core)

1. **String Output**
   Input: `String name = "tesDarj"`
   Output: `tesDarj`

2. **Missing Numbers in Array**
   Input: `{5,7,10}`
   Logic: Find missing numbers between min and max
   Output: `{6,8,9}`

3. **Remove Duplicate Characters**
   Input: `"hitesh darji"`
   Output: `tesdarj`

4. **Prime Number Check**

5. **Reverse String**

6. **Rotate Array**
   Example: `{1,2,3,4,5,6,7,8,9,10}`, rotate by 4
   Output: `{5,6,7,8,9,10,1,2,3,4}`

7. **Maximum Repeating Character**

8. **Pattern Program**

   ```
   3 2 1
   2 1
   1
   ```

9. **Reverse Words in String**
   Input: `"abc def"`
   Output: `"cba fed"`

Section 2: Java Concepts

1. Count characters in a string
2. Count duplicate characters

3. Capitalize first letter of each word
4. Palindrome check (with and without reverse)
5. Swap strings (with & without third variable)
6. Missing number (1–10)
7. HashMap duplicate key behavior
8. Meaning of `System.out.println`
9. Email validation program
10. Check if string contains only digits

Section 3: Selenium / Cypress / Playwright – Core Concepts

1. Why Cypress came? Why Playwright came?
2. Selenium vs Cypress vs Playwright
3. Sync vs Async execution
4. Parallel execution (TestNG / Playwright)
5. TestNG annotations
6. Page Object Model – framework or design pattern?
7. POM vs Framework difference
8. Handling alerts, mouse events, windows in Cypress
9. All mouse events in Cypress
10. var vs let
11. Global variables
12. HashMap usage in Cypress

Section 4: Cypress Framework & API

1. API automation in same project – yes or no?
2. Advantages of Cypress over Selenium
3. Fixtures – other uses
4. PUT request – record not found status code
5. Cucumber framework in Cypress
6. Combine UI + API framework in Cypress

Section 5: Playwright Core Interview Questions

1. Why Playwright is better than Selenium
2. Parallel execution in Playwright
3. Test execution flow
4. Actions: right click, drag-drop, multiple windows
5. Locator chaining (skip for-loop)
6. page.locator vs page.$
7. Shadow DOM & SVG handling
8. hasText usage
9. Sorting validation
10. HTML table – get data based on condition
11. Reports used in Playwright

12. Browser vs Context vs Page fixture
13. Auto-wait vs explicit wait
14. Request API library in Playwright

Section 6: Login Bypass & Optimization

1. API login and token injection in localStorage
2. One-time login using storageState
3. Setup activity using API
4. Cleanup activity using API
5. Reduce execution time & flakiness

Section 7: CI/CD & Debugging

1. Local pass but CI fail – how to debug
2. Parallel execution issues in CI
3. What to take care in CI parallel runs
4. Mock API usage

Section 8: Special & Thinking Questions

1. Variable lifecycle & memory storage
2. Internal working of Selenium / Cypress / Playwright
3. Improve debugging skills
4. Understand API automation internals
5. Always ask **WHY** in design decisions

Section 9: Practice Strategy

1. Write 100 programs
2. First simple logic, then optimize
3. Focus on time & space complexity
4. Daily practice + revision

Section 10: Naveen Style Interview Questions

1. Hybrid Web + API framework design
2. E2E framework with CI + reports + email
3. Avoid repeated login using session/token
4. Promise types & handling
5. Dynamic locator handling
6. Custom fixtures – beforeAll vs beforeEach

Final Tip

Strong interviews focus on **logic + WHY + real project experience + optimization mindset**