50 Java coding questions with solutions

🧠 SECTION 1: Basic Logic Building (Q1–10)

1. **Check Even or Odd**

```
int n = 7;
System.out.println(n % 2 == 0 ? "Even" : "Odd");
```

2. **Check Prime Number**

```
boolean isPrime = true;
for(int i=2; i<=n/2; i++) if(n%i==0) isPrime = false;
```

3. **Factorial of a Number**

```
int fact = 1;
for(int i = 1; i <= n; i++) fact *= i;
```

4. **Fibonacci Series (n terms)**

```
int a=0, b=1;
for(int i=0; i<n; i++) {
   System.out.print(a+" ");
   int temp = a + b;
   a = b;
   b = temp;
}
```

5. **Palindrome Number**

```
int temp=n, rev=0;
while(n!=0){ rev=rev*10+n%10; n/=10; }
System.out.println(temp == rev);
```

6. **Reverse a Number**

```
int rev = 0;
while(n > 0) {
   rev = rev*10 + n%10;
   n /= 10;
}
```

7. **Sum of Digits**

```
int sum = 0;
```

50 Java coding questions with solutions

```
while(n != 0){ sum += n % 10; n /= 10; }
```

8. **Swap Two Numbers (Without Temp)**

```
a = a + b;
b = a - b;
a = a - b;
```

9. **Greatest of Three Numbers**

```
int max = (a > b) ? (a > c ? a : c) : (b > c ? b : c);
```

10. **Armstrong Number**

```
int sum=0, temp=n;
while(n!=0){
   int d = n%10;
   sum += d*d*d;
   n /= 10;
}
System.out.println(temp == sum);
```

🔠 SECTION 2: String Problems (Q11–20)

11. **Reverse a String**

```
StringBuilder sb = new StringBuilder(str);
System.out.println(sb.reverse());
```

12. **Palindrome String**

```
String rev = new StringBuilder(str).reverse().toString();
System.out.println(str.equals(rev));
```

13. **Count Vowels and Consonants**

```
int v=0, c=0;
for(char ch : str.toLowerCase().toCharArray()){
   if("aeiou".indexOf(ch)!=-1) v++;
   else if(Character.isLetter(ch)) c++;
}
```

14. **Remove Duplicate Characters**

```
StringBuilder res = new StringBuilder();
```

```
for(char c : str.toCharArray()){
   if(res.indexOf(String.valueOf(c)) == -1)
      res.append(c);
}
```

15. **Count Each Character Frequency**

```
Map<Character,Integer> map = new HashMap<>();
for(char c : str.toCharArray()){
   map.put(c, map.getOrDefault(c, 0)+1);
}
```

16. **Anagram Check**

```
char[] a1 = s1.toCharArray();
char[] a2 = s2.toCharArray();
Arrays.sort(a1); Arrays.sort(a2);
System.out.println(Arrays.equals(a1, a2));
```

17. **Longest Word in Sentence**

```
String[] words = str.split(" ");
String longest = "";
for(String word : words){
   if(word.length() > longest.length()) longest = word;
}
```

18. **Capitalize First Letter of Each Word**

```
String[] words = str.split(" ");
for(String w : words){
   System.out.print(Character.toUpperCase(w.charAt(0)) + w.substring(1) + " ");
}
```

19. **Check Pangram**

```
boolean[] mark = new boolean[26];
for(char ch : str.toLowerCase().toCharArray()){
   if(ch >= 'a' && ch <= 'z') mark[ch - 'a'] = true;
}
System.out.println(Arrays.stream(mark).allMatch(b -> b));
```

50 Java coding questions with solutions

20. **Count Words in String**

```
int count = str.trim().split("\\s+").length;
```

📦 **SECTION 3: Array Problems (Q21–30)**

21. **Find Maximum Element**

```
int max = arr[0];
for(int i : arr) if(i > max) max = i;
```

22. **Find Minimum Element**

```
int min = arr[0];
for(int i : arr) if(i < min) min = i;
```

23. **Sum of All Elements**

```
int sum = 0;
for(int i : arr) sum += i;
```

24. **Reverse Array**

```
for(int i=0, j=arr.length-1; i<j; i++, j--){
   int temp = arr[i];
   arr[i] = arr[j];
   arr[j] = temp;
}
```

25. **Search Element in Array**

```
for(int i=0; i<arr.length; i++) {
   if(arr[i] == key) return i;
}
```

26. **Second Largest Element**

```
int max=Integer.MIN_VALUE, second=Integer.MIN_VALUE;
for(int n : arr){
   if(n > max){
      second = max;
      max = n;
   } else if(n > second && n != max){
      second = n;
```

```
    }
}
```

27. **Sort Array (Bubble Sort)**

```
for(int i=0;i<n-1;i++)
    for(int j=0;j<n-i-1;j++)
        if(arr[j]>arr[j+1]){
            int t=arr[j]; arr[j]=arr[j+1]; arr[j+1]=t;
        }
```

28. **Remove Duplicates from Sorted Array**

```
int j=0;
for(int i=1;i<arr.length;i++){
    if(arr[i] != arr[j]) arr[++j] = arr[i];
}
```

29. **Left Rotate Array by 1**

```
int first = arr[0];
for(int i=0; i<arr.length-1; i++) arr[i] = arr[i+1];
arr[arr.length-1] = first;
```

30. **Find Frequency of Each Element**

```
Map<Integer, Integer> map = new HashMap<>();
for(int i : arr) map.put(i, map.getOrDefault(i, 0)+1);
```

💡 **SECTION 4: Intermediate DSA Problems (Q31–40)**

31. **Binary Search**

```
while(l <= r){
    int mid = (l + r) / 2;
    if(arr[mid] == key) return mid;
    else if(arr[mid] < key) l = mid + 1;
    else r = mid - 1;
}
```

32. **Linear Search**

```
for(int i=0; i<arr.length; i++) if(arr[i]==key) return i;
```

50 Java coding questions with solutions

33. **Check if Array is Sorted**

```
boolean sorted = true;
for(int i=1; i<arr.length; i++)
   if(arr[i] < arr[i-1]) sorted = false;
```

34. **Two Sum Problem**

```
Map<Integer,Integer> map = new HashMap<>();
for(int i=0;i<arr.length;i++){
   int diff = target - arr[i];
   if(map.containsKey(diff)) return new int[]{map.get(diff), i};
   map.put(arr[i], i);
}
```

35. **Merge Two Arrays**

```
int[] merged = new int[a.length + b.length];
System.arraycopy(a, 0, merged, 0, a.length);
System.arraycopy(b, 0, merged, a.length, b.length);
```

36. **Intersection of Two Arrays**

```
Set<Integer> set = new HashSet<>();
for(int i : arr1) set.add(i);
for(int i : arr2) if(set.contains(i)) System.out.println(i);
```

37. **Matrix Addition**

```
for(int i=0; i<m; i++)
   for(int j=0; j<n; j++)
      result[i][j] = a[i][j] + b[i][j];
```

38. **Transpose of Matrix**

```
for(int i=0; i<rows; i++)
   for(int j=0; j<cols; j++)
      transpose[j][i] = matrix[i][j];
```

39. **Spiral Order Matrix Print**

```
// Top, Bottom, Left, Right pointers approach
```

40. **Kadane's Algorithm – Max Subarray Sum**

```
int maxSum = arr[0], curr = arr[0];
```

```
for(int i=1; i<arr.length; i++){
   curr = Math.max(arr[i], curr + arr[i]);
   maxSum = Math.max(maxSum, curr);
}
```

🧩 **SECTION 5: Advanced & Logical (Q41–50)**

41. **Balanced Parentheses**

```
Stack<Character> s = new Stack<>();
for(char c : str.toCharArray()){
   if(c=='(') s.push(c);
   else if(c==')'){
      if(s.isEmpty()) return false;
      s.pop();
   }
}
return s.isEmpty();
```

42. **Valid Palindrome Ignoring Non-Alphanumerics**

```
String clean = s.replaceAll("[^a-zA-Z0-9]", "").toLowerCase();
return clean.equals(new StringBuilder(clean).reverse().toString());
```

43. **Check Power of 2**

```
boolean isPowerOfTwo = (n & (n - 1)) == 0;
```

44. **Count Set Bits**

```
int count=0;
while(n>0){
   n = n & (n - 1);
   count++;
}
```

45. **Find Missing Number in Array 1 to N**

```
int sum = (n+1)*(n+2)/2;
for(int i : arr) sum -= i;
```

50 Java coding questions with solutions

46. **LCM & GCD**

```
int gcd = 1;
for(int i=1; i<=a && i<=b; i++)
   if(a%i==0 && b%i==0) gcd=i;
int lcm = (a*b)/gcd;
```

47. **Binary to Decimal**

```
int dec = Integer.parseInt(binaryStr, 2);
```

48. **Decimal to Binary**

```
String bin = Integer.toBinaryString(n);
```

49. **Sort String Alphabetically**

```
char[] ch = str.toCharArray();
Arrays.sort(ch);
System.out.println(new String(ch));
```

50. **Check Armstrong in Any Digits**

```
int temp=n, sum=0, len=String.valueOf(n).length();
while(n!=0){
   sum += Math.pow(n%10, len);
   n /= 10;
}
System.out.println(temp == sum);
```
