Java Technical Interview Programming Questions

This Programming Questions were Asked in **Tech Mahindra**, **TCS**, **Wipro**, **Infosys**, **Accenture**, **Capgemini** and Many Other Service and Product Based Companies

---

1. Swap Two Numbers

Problem Statement

Write a program to swap two numbers without using a third variable.

Input

```
a = 10
b = 20
```

Output

```
a = 20
b = 10
```


Explanation

To swap two numbers without using a third variable, you can use arithmetic operations to interchange the values.

Program

```
public class learn2code {
    public static void main(String[] args) {
        int a = 10;
        int b = 20;

        // Swapping without a third variable
```

```
        a = a + b; // a = 30
        b = a - b; // b = 10
        a = a - b; // a = 20

        System.out.println("a = " + a);
        System.out.println("b = " + b);
    }
}
```

---

2. Reverse a Number

Problem Statement

Write a program to reverse a number.

Input

```
1234
```

Output

```
4321
```


Explanation

To reverse a number, repeatedly extract the last digit and build the reversed number.

Program

```
public class learn2code {
    public static void main(String[] args) {
        int num = 1234;
        int reversed = 0;

        while (num != 0) {
            int digit = num % 10;
            reversed = reversed * 10 + digit;
```

```
            num /= 10;
        }

        System.out.println("Reversed Number: " + reversed);
    }
}
```

---

3. Reverse a String

Problem Statement

Write a program to reverse a string.

Input

```
"ABCD"
```

Output

```
"DCBA"
```

Explanation

To reverse a string, convert it to a character array and swap characters from the ends towards the center.

Program

```
public class learn2code {
    public static void main(String[] args) {
        String str = "ABCD";
        char[] chars = str.toCharArray();
        int left = 0, right = chars.length - 1;

        while (left < right) {
            char temp = chars[left];
            chars[left] = chars[right];
            chars[right] = temp;
```

```
            left++;
            right--;
        }

        String reversedStr = new String(chars);
        System.out.println("Reversed String: " + reversedSt
r);
    }
}
```

---

4. Fibonacci Series

Problem Statement

Write a program to print the Fibonacci series up to 10 terms.

Output

```
0, 1, 1, 2, 3, 5, 8, 13, 21, 34
```

Explanation

The Fibonacci series is a sequence where each number is the sum of the two preceding ones.

Program

```
public class learn2code {
    public static void main(String[] args) {
        int n = 10;
        int a = 0, b = 1;

        System.out.print(a + ", " + b);

        for (int i = 2; i < n; i++) {
            int next = a + b;
            System.out.print(", " + next);
            a = b;
```

```
            b = next;
        }
    }
}
```

---

5. Check if a Number is a Palindrome

Problem Statement

Write a program to check if a number is a palindrome.

Input

```
1221
```

Output

```
1221 is a Palindrome
```

Explanation

A palindrome is a number that reads the same backward as forward. Reverse the number and compare it to the original.

Program

```
public class learn2code {
    public static void main(String[] args) {
        int num = 1221;
        int original = num;
        int reversed = 0;

        while (num != 0) {
            int digit = num % 10;
            reversed = reversed * 10 + digit;
            num /= 10;
        }
```

```
        if (original == reversed) {
            System.out.println(original + " is a Palindrom
e");
        } else {
            System.out.println(original + " is not a Palind
rome");
        }
    }
}
```

---

6. Check if a String is a Palindrome

Problem Statement

Write a program to check if a string is a palindrome.

Input

```
"DAD"
```

Output

```
DAD is a Palindrome
```


Explanation

A palindrome string reads the same backward as forward. Reverse the string and compare it to the original.

Program

```
public class learn2code {
    public static void main(String[] args) {
        String str = "DAD";
        String reversed = new StringBuilder(str).reverse().
toString();

        if (str.equals(reversed)) {
```

```
            System.out.println(str + " is a Palindrome");
        } else {
            System.out.println(str + " is not a Palindrom
e");
        }
    }
}
```

7. Find Prime Numbers in a Given Range

Problem Statement

Write a program to find prime numbers between 2 and 10.

Input

```
2, 10
```

Output

```
Prime numbers between 2 and 10 are: 2, 3, 5, 7
```

Explanation

A prime number is a number greater than 1 with no divisors other than 1 and itself.

Program

```
public class learn2code {
    public static void main(String[] args) {
        int start = 2, end = 10;

        System.out.print("Prime numbers between " + start +
" and " + end + " are: ");

        for (int i = start; i <= end; i++) {
            if (isPrime(i)) {
```

```
                System.out.print(i + " ");
            }
        }
    }

    static boolean isPrime(int n) {
        if (n <= 1) {
            return false;
        }
        for (int i = 2; i <= Math.sqrt(n); i++) {
            if (n % i == 0) {
                return false;
            }
        }
        return true;
    }
}
```

8. Check if a Number is Prime

Problem Statement

Write a program to check if a given number is a prime number.

Input

```
2
```

Output

```
2 is a Prime Number
```

Explanation

Check if a number has no divisors other than 1 and itself.

Program

```
public class learn2code {
    public static void main(String[] args) {
        int num = 2;

        if (isPrime(num)) {
            System.out.println(num + " is a Prime Number");
        } else {
            System.out.println(num + " is not a Prime Numbe
r");
        }
    }

    static boolean isPrime(int n) {
        if (n <= 1) {
            return false;
        }
        for (int i = 2; i <= Math.sqrt(n); i++) {
            if (n % i == 0) {
                return false;
            }
        }
        return true;
    }
}
```


9. Find Factorial of a Number

Problem Statement

Write a program to find the factorial of a number.

Input

```
5
```

Output

```
120
```

Explanation

The factorial of a number is the product of all positive integers less than or equal to that number.

Program

```
public class learn2code {
    public static void main(String[] args) {
        int num = 5;
        int factorial = 1;

        for (int i = 1; i <= num; i++) {
            factorial *= i;
        }

        System.out.println("Factorial of " + num + " is " +
factorial);
    }
}
```


10. Check if a Number is an Armstrong Number

Problem Statement

Write a program to check if a number is an Armstrong number.

Input

```
153
```

Output

```
153 is an Armstrong Number
```

Explanation

An Armstrong number is a number that is equal to the sum of its own digits raised to the power of the number of digits.

Program

```
public class learn2code {
    public static void main(String[] args) {
        int num = 153;
        int original = num;
        int sum = 0;

        while (num != 0) {
            int digit = num % 10;
            sum += Math.pow(digit, 3);
            num /= 10;
        }

        if (sum == original) {
            System.out.println(original + " is an Armstrong
Number");
        } else {
            System.out.println(original + " is not an Armst
rong Number");
        }
    }
}
```


11. Count Number of Digits in a Number

Problem Statement

Write a program to count the number of digits in a number.

Input

```
1234
```

Output

```
Number of digits are 4
```

Explanation

To count the number of digits, repeatedly divide the number by 10 until it becomes 0, counting the iterations.

Program

```
public class learn2code {
    public static void main(String[] args) {
        int num = 1234;
        int count = 0;

        while (num != 0) {
            num /= 10;
            count++;
        }

        System.out.println("Number of digits are " + coun
t);
    }
}
```


12. Count Even and Odd Digits in a Number

Problem Statement

Write a program to count the number of even and odd digits in a number.

Input

```
1234
```

Output

```
Even Numbers: 2
Odd Numbers: 2
```

Explanation

To count even and odd digits, extract each digit and check if it's divisible by 2.

Program

```
public class learn2code {
    public static void main(String[] args) {
        int num = 1234;
        int evenCount = 0;
        int oddCount = 0;

        while (num != 0) {
            int digit = num % 10;
            if (digit % 2 == 0) {
                evenCount++;
            } else {
                oddCount++;
            }
            num /= 10;
        }

        System.out.println("Even Numbers: " + evenCount);
        System.out.println("Odd Numbers: " + oddCount);
    }
}
```


13. Check if a Number is Even or Odd

Problem Statement

Write a program to check if a number is even or odd.

Input

```
2
```

Output

```
Even Number
```

Explanation

A number is even if it is divisible by 2, otherwise it is odd.

Program

```
public class learn2code {
    public static void main(String[] args) {
        int num = 2;

        if (num % 2 == 0) {
            System.out.println(num + " is an Even Number");
        } else {
            System.out.println(num + " is an Odd Number");
        }
    }
}
```


---

14. Find Sum of Digits in a Number

Problem Statement

Write a program to find the sum of digits in a number.

Input

```
1234
```

Output

```
10
```

Explanation

To find the sum of digits, extract each digit and add them together.

Program

```
public class learn2code {
    public static void main(String[] args) {
        int num = 1234;
        int sum = 0;

        while (num != 0) {
            int digit = num % 10;
            sum += digit;
            num /= 10;
        }

        System.out.println("Sum of digits: " + sum);
    }
}
```


15. Find the Largest Number

Problem Statement

Write a program to find the largest number among three numbers.

Input

```
a = 10, b = 20, c = 30
```

Output

```
Largest Number is 30
```

Explanation

To find the largest number, compare each number with the others and determine the maximum.

Program

```
public class learn2code {
    public static void main(String[] args) {
        int a = 10;
        int b = 20;
        int c = 30;

        int largest = (a > b) ? (a > c ? a : c) : (b > c ?
b : c);

        System.out.println("Largest Number is " + largest);
    }
}
```


16. Generate Random Number and String

Problem Statement

Write a program to generate a random number and a random string.

Explanation

Use `Math.random()` for generating random numbers and a loop for random strings.

Program

```
import java.util.Random;

public class learn2code {
    public static void main(String[] args) {
        // Generate random number
        int randomNumber = (int) (Math.random() * 100);
        System.out.println("Random Number: " + randomNumbe
```

```
r);

        // Generate random string
        String characters = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcd
efghijklmnopqrstuvwxyz0123456789";
        StringBuilder randomString = new StringBuilder();
        Random rand = new Random();
        int length = 10;

        for (int i = 0; i < length; i++) {
            randomString.append(characters.charAt(rand.next
Int(characters.length())));
        }

        System.out.println("Random String: " + randomStrin
g.toString());
    }
}
```

17. Find Sum of Elements in an Array

Problem Statement

Write a program to find the sum of elements in an array.

Input

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

Output

```
10
```

Explanation

To find the sum of elements, iterate through the array and add each element.

Program

```
public class learn2code {
    public static void main(String[] args) {
        int[] a = {1, 2, 3, 4};
        int sum = 0;

        for (int i = 0; i < a.length; i++) {
            sum += a[i];
        }

        System.out.println("Sum of elements: " + sum);
    }
}
```

18. Print Even and Odd Numbers in an Array

Problem Statement

Write a program to print even and odd numbers in an array.

Input

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


Output

```
Even Numbers: 2, 4
Odd Numbers: 1, 3
```

Explanation

To print even and odd numbers, iterate through the array and check each number's divisibility by 2.

Program

```
public class learn2code {
    public static void main(String[] args) {
```

```
        int[] a = {1, 2, 3, 4};

        System.out.print("Even Numbers: ");
        for (int i = 0; i < a.length; i++) {
            if (a[i] % 2 == 0) {
                System.out.print(a[i] + " ");
            }
        }

        System.out.print("\\nOdd Numbers: ");
        for (int i = 0; i < a.length; i++) {
            if (a[i] % 2 != 0) {
                System.out.print(a[i] + " ");
            }
        }
    }
}
```

19. Check if Two Arrays are Equal

Problem Statement

Write a program to check if two arrays are equal.

Input

```
a[] = {1, 2, 3, 4}
b[] = {1, 2, 3, 4}
```

Output

```
a and b are equal
```

Explanation

Two arrays are equal if they have the same length and corresponding elements are equal.

Program

```
import java.util.Arrays;

public class learn2code {
    public static void main(String[] args) {
        int[] a = {1, 2, 3, 4};
        int[] b = {1, 2, 3, 4};

        if (Arrays.equals(a, b)) {
            System.out.println("a and b are equal");
        } else {
            System.out.println("a and b are not equal");
        }
    }
}
```

20. Find Missing Number in an Array

Problem Statement

Write a program to find the missing number in an array.

Input

```
a[] = {1, 2, 3, 5}
```

Output

```
Missing number is 4
```

Explanation

To find the missing number, calculate the expected sum and subtract the actual sum from it.

Program

```
public class learn2code {
    public static void main(String[] args) {
        int[] a = {1, 2, 3, 5};
        int n = a.length + 1;
        int totalSum = n * (n + 1) / 2;
        int actualSum = 0;

        for (int i = 0; i < a.length; i++) {
            actualSum += a[i];
        }

        int missingNumber = totalSum - actualSum;
        System.out.println("Missing number is " + missingNu
mber);
    }
}
```

21. Find the Max and Min Element in an Array

Problem Statement

Write a program to find the maximum and minimum elements in an array.

Input

```
a[] = {3, 4, 5, 6}
```

Output

```
Max is 6
Min is 3
```

Explanation

To find the maximum and minimum elements, iterate through the array and keep track of the largest and smallest values.

Program

```
public class learn2code {
    public static void main(String[] args) {
        int[] a = {3, 4, 5, 6};
        int max = a[0];
        int min = a[0];

        for (int i = 1; i < a.length; i++) {
            if (a[i] > max) {
                max = a[i];
            }
            if (a[i] < min

) {
                min = a[i];
            }
        }

        System.out.println("Max is " + max);
        System.out.println("Min is " + min);
    }
}
```


22. Separate 0s and 1s in an Array

Problem Statement

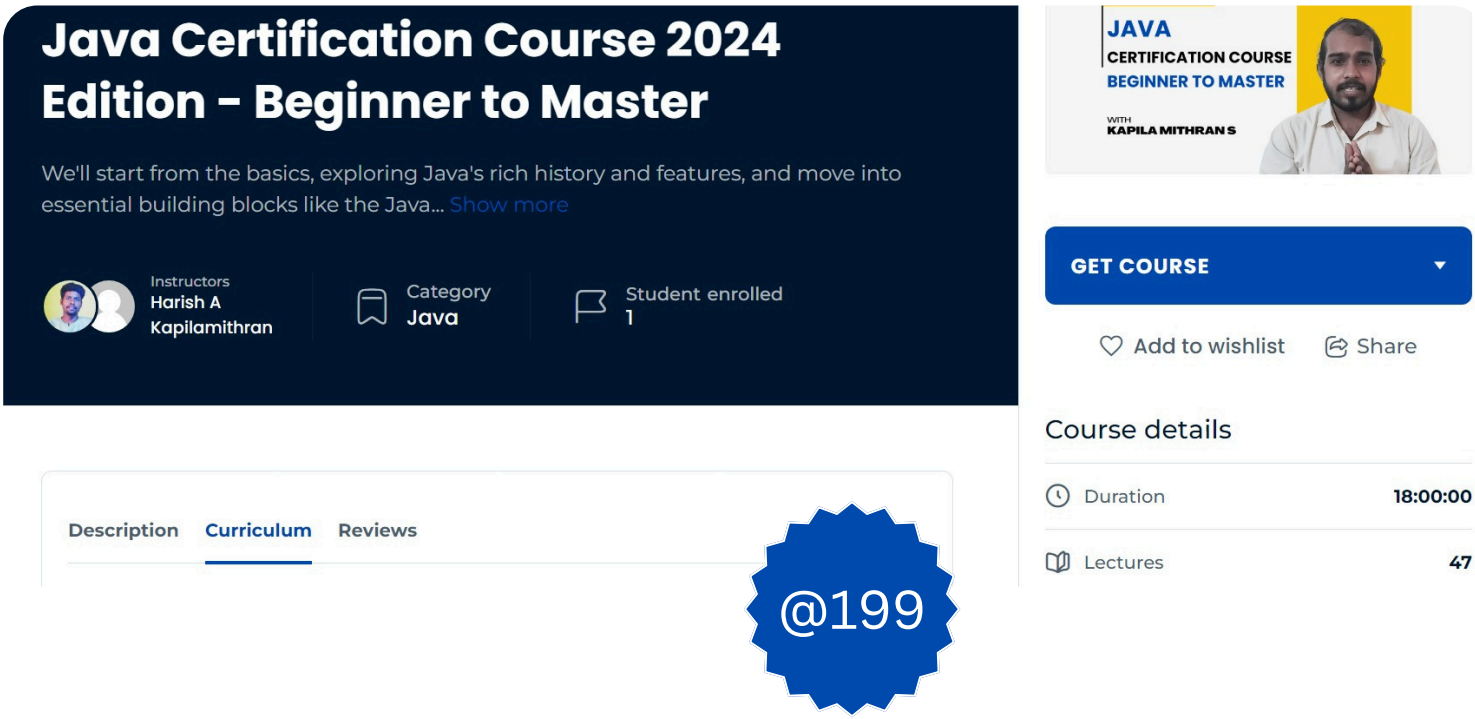
Write a program to separate 0s and 1s in an array.

Input

```
a[] = {0, 1, 0, 1, 0, 1}
```

Output

```
0, 0, 0, 1, 1, 1
```



Becoming a Java Developer

JAVA CERTIFICATION COURSE 2024 EDITION - BEGINNER TO MASTER

Visit: https://learn2code.tech/

Explanation

To separate 0s and 1s, count the number of 0s and 1s and then place them accordingly.

Program

```
public class learn2code {
    public static void main(String[] args) {
        int[] a = {0, 1, 0, 1, 0, 1};
        int count = 0;

        for (int i = 0; i < a.length; i++) {
            if (a[i] == 0) {
                count++;
            }
        }

        for (int i = 0; i < count; i++) {
            a[i] = 0;
        }

        for (int i = count; i < a.length; i++) {
            a[i] = 1;
        }

        System.out.print("Array after separating 0s and 1s:
");
        for (int i = 0; i < a.length; i++) {
            System.out.print(a[i] + " ");
        }
    }
}
```


23. Find Duplicates in an Array

Problem Statement

Write a program to find duplicates in an array.

Input

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

Output

```
1, 2, 3
```

Explanation

To find duplicates, use a nested loop to compare each element with the others.

Program

```
import java.util.HashSet;

public class learn2code {
    public static void main(String[] args) {
        int[] a = {1, 2, 3, 1, 2, 3, 4};
        HashSet<Integer> seen = new HashSet<>();
        HashSet<Integer> duplicates = new HashSet<>();

        for (int i = 0; i < a.length; i++) {
            if (!seen.add(a[i])) {
                duplicates.add(a[i]);
            }
        }

        System.out.print("Duplicates: ");
        for (int num : duplicates) {
            System.out.print(num + " ");
        }
    }
}
```


24. Find the Second Largest Element in an Array

Problem Statement

Write a program to find the second largest element in an array.

Input

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

Output

```
3
```

Explanation

To find the second largest element, maintain two variables for the largest and second largest values.

Program

```
public class learn2code {
    public static void main(String[] args) {
        int[] a = {1, 2, 3, 4};
        int largest = Integer.MIN_VALUE;
        int secondLargest = Integer.MIN_VALUE;

        for (int i = 0; i < a.length; i++) {
            if (a[i] > largest) {
                secondLargest = largest;
                largest = a[i];
            } else if (a[i] > secondLargest && a[i] != larg
est) {
                secondLargest = a[i];
            }
        }

        System.out.println("Second largest element is " + s
econdLargest);
```

```
    }
}
```

25. Move all Zeros to End of Array

Problem Statement

Write a program to move all zeros to the end of an array.

Input

```
a[] = {0, 1, 0, 3, 12}
```

Output

```
1, 3, 12, 0, 0
```

Explanation

To move all zeros to the end, iterate through the array and place non-zero elements at the front, then fill the rest with zeros.

Program

```
public class learn2code {
    public static void main(String[] args) {
        int[] a = {0, 1, 0, 3, 12};
        int count = 0;

        for (int i = 0; i < a.length; i++) {
            if (a[i] != 0) {
                a[count++] = a[i];
            }
        }

        while (count < a.length) {
            a[count++] = 0;
        }
```

```
        System.out.print("Array after moving zeros to end:
");
        for (int i = 0; i < a.length; i++) {
            System.out.print(a[i] + " ");
        }
    }
}
```

26. Find Intersection of Two Arrays

Problem Statement

Write a program to find the intersection of two arrays.

Input

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

Output

```
3, 4
```


Explanation

The intersection of two arrays is the set of elements that are present in both arrays.

Program

```
import java.util.HashSet;

public class learn2code {
    public static void main(String[] args) {
        int[] a = {1, 2, 3, 4};
        int[] b = {3, 4, 5, 6};
        HashSet<Integer> set = new HashSet<>();
```

```
        for (int i = 0; i < a.length; i++) {
            set.add(a[i]);
        }

        System.out.print("Intersection: ");
        for (int i = 0; i < b.length; i++) {
            if (set.contains(b[i])) {
                System.out.print(b[i] + " ");
            }
        }
    }
}
```

27. Find Union of Two Arrays

Problem Statement

Write a program to find the union of two arrays.

Input

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


Output

```
1, 2, 3, 4, 5, 6
```

Explanation

The union of two arrays is the set of all distinct elements present in both arrays.

Program

```
import java.util.HashSet;

public class learn2code {
```

```
    public static void main(String[] args) {
        int[] a = {1, 2, 3, 4};
        int[] b = {3, 4, 5, 6};
        HashSet<Integer> set = new HashSet<>();

        for (int i = 0; i < a.length; i++) {
            set.add(a[i]);
        }

        for (int i = 0; i < b.length; i++) {
            set.add(b[i]);
        }

        System.out.print("Union: ");
        for (int num : set) {
            System.out.print(num + " ");
        }
    }
}
```

By practicing these problems, you'll not only prepare for interviews but also build a solid foundation in programming logic.