Top 50 commonly asked **Java 8 coding logical interview** questions and answers:

Q: Write a program to find the sum of all numbers in a list using Java 8 streams.

A:

List<Integer> numbers = Arrays.asList(1, 2, 3, 4, 5);

int sum = numbers.stream().mapToInt(Integer::intValue).sum();

System.out.println("Sum: " + sum);

-------------------------------------------------------------------------------------------------------------

Q: Write a program to remove duplicates from an ArrayList using Java 8 streams.

A:

List<Integer> numbers = Arrays.asList(1, 2, 2, 3, 4, 4, 5);

List<Integer> uniqueNumbers = numbers.stream().distinct().collect(Collectors.toList());

System.out.println("Unique numbers: " + uniqueNumbers);

-------------------------------------------------------------------------------------------------------------

Q: Write a program to count the number of occurrences of each element in an ArrayList using Java 8 streams.

A:

List<Integer> numbers = Arrays.asList(1, 2, 2, 3, 3, 3, 4, 5);

Map<Integer, Long> occurrenceMap = numbers.stream().collect(Collectors.groupingBy(Function.identity(), Collectors.counting()));

System.out.println("Occurrence map: " + occurrenceMap);

-------------------------------------------------------------------------------------------------------------

Q: Write a program to find the maximum element in an ArrayList using Java 8 streams.

A:

List<Integer> numbers = Arrays.asList(1, 2, 3, 4, 5);

Optional<Integer> max = numbers.stream().max(Integer::compareTo);

if (max.isPresent()) {

System.out.println("Max: " + max.get());

} else {

System.out.println("List is empty.");

}

-------------------------------------------------------------------------------------------------------------

Q: Write a program to convert a list of strings to uppercase using Java 8 streams.

A:

List<String> strings = Arrays.asList("hello", "world");

List<String> uppercaseStrings = strings.stream().map(String::toUpperCase).collect(Collectors.toList());

System.out.println("Uppercase strings: " + uppercaseStrings);

-------------------------------------------------------------------------------------------------------------

Q: Write a program to filter even numbers from a list using Java 8 streams.

A:

List<Integer> numbers = Arrays.asList(1, 2, 3, 4, 5);

List<Integer> evenNumbers = numbers.stream().filter(n -> n % 2 == 0).collect(Collectors.toList());

System.out.println("Even numbers: " + evenNumbers);

-------------------------------------------------------------------------------------------------------------

Q: Write a program to find the average of all numbers in a list using Java 8 streams.

A:

List<Integer> numbers = Arrays.asList(1, 2, 3, 4, 5);

OptionalDouble average = numbers.stream().mapToDouble(Integer::doubleValue).average();

if (average.isPresent()) {

System.out.println("Average: " + average.getAsDouble());

} else {

System.out.println("List is empty.");

}

-------------------------------------------------------------------------------------------------------------

Q: Write a program to check if a list contains a specific element using Java 8 streams.

A:

List<Integer> numbers = Arrays.asList(1, 2, 3, 4, 5);

boolean containsElement = numbers.stream().anyMatch(n -> n == 3);

System.out.println("Contains element: " + containsElement);

-------------------------------------------------------------------------------------------------------------

Q: Write a program to sort a list of strings in alphabetical order using Java 8 streams.

A:

List<String> strings = Arrays.asList("world", "hello");

List<String> sortedStrings = strings.stream().sorted().collect(Collectors.toList());

System.out.println("Sorted strings: " + sortedStrings);

-------------------------------------------------------------------------------------------------------------

Q: Write a program to find the second smallest number in an array using Java 8 streams.

A:

Integer[] numbers = {1, 3, 2, 5, 4};

Optional<Integer> secondSmallest = Arrays.stream(numbers).distinct().sorted().skip(1).findFirst();

if (secondSmallest.isPresent()) {

System.out.println("Second smallest: " + secondSmallest.get());

} else {

System.out.println("Array is empty or doesn't have a second smallest element.");

}

-------------------------------------------------------------------------------------------------------------

Q: Write a program to check if a string is a palindrome using Java 8 streams.

A:

String str = "level";

boolean isPalindrome = IntStream.range(0, str.length() / 2)

.allMatch(i -> str.charAt(i) == str.charAt(str.length() - i - 1));

System.out.println("Is palindrome: " + isPalindrome);

-------------------------------------------------------------------------------------------------------------

Q: Write a program to reverse a string using Java 8 streams.

A:

String str = "hello";

String reversed = new StringBuilder(str).reverse().toString();

System.out.println("Reversed string: " + reversed);

-------------------------------------------------------------------------------------------------------------

Q: Write a program to find the second highest element in an array using Java 8 streams.

A:

Integer[] numbers = {1, 3, 2, 5, 4};

Optional<Integer> secondHighest = Arrays.stream(numbers).distinct().sorted(Comparator.reverseOrder()).skip(1).findFirst();

if (secondHighest.isPresent()) {

System.out.println("Second highest: " + secondHighest.get());

} else {

System.out.println("Array is empty or doesn't have a second highest element.");

}

-------------------------------------------------------------------------------------------------------------

Q: Write a program to find the factorial of a number using Java 8 streams.

A:

int number = 5;

int factorial = IntStream.rangeClosed(1, number).reduce(1, (a, b) -> a \* b);

System.out.println("Factorial: " + factorial);

-------------------------------------------------------------------------------------------------------------

Q: Write a program to check if a number is prime using Java 8 streams.

A:

int number = 7;

boolean isPrime = IntStream.rangeClosed(2, (int) Math.sqrt(number)).noneMatch(n -> number % n == 0);

System.out.println("Is prime: " + isPrime);

-------------------------------------------------------------------------------------------------------------

Q: Write a program to find the third largest element in an array using Java 8 streams.

A:

Integer[] numbers = {1, 3, 2, 5, 4};

Optional<Integer> thirdLargest = Arrays.stream(numbers)

.distinct()

.sorted(Comparator.reverseOrder())

.skip(2)

.findFirst();

if (thirdLargest.isPresent()) {

System.out.println("Third largest: " + thirdLargest.get());

} else {

System.out.println("Array is empty or doesn't have a third largest element.");

}

-------------------------------------------------------------------------------------------------------------

Q: Write a program to find the maximum and minimum element in an array using Java 8 streams.

A:

Integer[] numbers = {1, 3, 2, 5, 4};

Optional<Integer> max = Arrays.stream(numbers).max(Integer::compareTo);

Optional<Integer> min = Arrays.stream(numbers).min(Integer::compareTo);

if (max.isPresent() && min.isPresent()) {

System.out.println("Max: " + max.get());

System.out.println("Min: " + min.get());

} else {

System.out.println("Array is empty.");

}

-------------------------------------------------------------------------------------------------------------

Q: Write a program to find the length of the longest string in an array using Java 8 streams.

A:

String[] strings = {"hello", "world", "java", "programming"};

OptionalInt maxLength = Arrays.stream(strings).mapToInt(String::length).max();

if (maxLength.isPresent()) {

System.out.println("Max length: " + maxLength.getAsInt());

} else {

System.out.println("Array is empty.");

}

-------------------------------------------------------------------------------------------------------------

Q: Write a program to find the common elements between two arrays using Java 8 streams.

A:

Integer[] arr1 = {1, 2, 3, 4, 5};

Integer[] arr2 = {4, 5, 6, 7, 8};

Set<Integer> commonElements = Arrays.stream(arr1)

.filter(Arrays.stream(arr2).collect(Collectors.toSet())::contains)

.collect(Collectors.toSet());

System.out.println("Common elements: " + commonElements);

-------------------------------------------------------------------------------------------------------------

Q: Write a program to find the frequency of each character in a string using Java 8 streams.

A:

String str = "hello world";

Map<Character, Long> frequencyMap = str.chars()

.mapToObj(c -> (char) c)

.collect(Collectors.groupingBy(Function.identity(), Collectors.counting()));

System.out.println("Frequency map: " + frequencyMap);

-------------------------------------------------------------------------------------------------------------

Q: Write a program to find the largest and smallest palindrome number in an array using Java 8 streams.

A:

Integer[] numbers = {121, 123, 232, 345, 454};

Optional<Integer> largestPalindrome = Arrays.stream(numbers)

.filter(n -> {

String str = String.valueOf(n);

return IntStream.range(0, str.length() / 2)

.allMatch(i -> str.charAt(i) == str.charAt(str.length() - i - 1));

})

.max(Integer::compareTo);

Optional<Integer> smallestPalindrome = Arrays.stream(numbers)

.filter(n -> {

String str = String.valueOf(n);

return IntStream.range(0, str.length() / 2)

.allMatch(i -> str.charAt(i) == str.charAt(str.length() - i - 1));

})

.min(Integer::compareTo);

if (largestPalindrome.isPresent() && smallestPalindrome.isPresent()) {

System.out.println("Largest palindrome: " + largestPalindrome.get());

System.out.println("Smallest palindrome: " + smallestPalindrome.get());

} else {

System.out.println("Array is empty or doesn't have any palindrome numbers.");

}

-------------------------------------------------------------------------------------------------------------

Q: Write a program to find the missing number in an array of consecutive numbers using Java 8 streams.

A:

int[] numbers = {1, 2, 3, 5, 6, 7, 8};

int missingNumber = IntStream.rangeClosed(1, numbers.length + 1)

.sum() - Arrays.stream(numbers).sum();

System.out.println("Missing number: " + missingNumber);

-------------------------------------------------------------------------------------------------------------

Q: Write a program to find the first non-repeated character in a string using Java 8 streams.

A:

String str = "hello world";

Optional<Character> firstNonRepeated = str.chars()

.mapToObj(c -> (char) c)

.filter(c -> str.indexOf(c) == str.lastIndexOf(c))

.findFirst();

if (firstNonRepeated.isPresent()) {

System.out.println("First non-repeated character: " + firstNonRepeated.get());

} else {

System.out.println("String doesn't have any non-repeated characters.");

}

-------------------------------------------------------------------------------------------------------------

Q: Write a program to find the power of each element in an array using Java 8 streams.

A:

double[] numbers = {2.0, 3.0, 4.0};

double[] powers = Arrays.stream(numbers).map(n -> Math.pow(n, 2)).toArray();

System.out.println("Powers: " + Arrays.toString(powers));

-------------------------------------------------------------------------------------------------------------

Q: Write a program to check if a string contains only digits using Java 8 streams.

A:

String str = "12345";

boolean containsOnlyDigits = str.chars().allMatch(Character::isDigit);

System.out.println("Contains only digits: " + containsOnlyDigits);

-------------------------------------------------------------------------------------------------------------

Q: Write a program to find the difference between the largest and smallest elements in an array using Java 8 streams.

A:

Integer[] numbers = {1, 3, 2, 5, 4};

Optional<Integer> max = Arrays.stream(numbers).max(Integer::compareTo);

Optional<Integer> min = Arrays.stream(numbers).min(Integer::compareTo);

if (max.isPresent() && min.isPresent()) {

int difference = max.get() - min.get();

System.out.println("Difference: " + difference);

} else {

System.out.println("Array is empty.");

}

-------------------------------------------------------------------------------------------------------------

Q: Write a program to concatenate all strings in an array using Java 8 streams.

A:

String[] strings = {"hello", "world", "java", "programming"};

String concatenated = Arrays.stream(strings).reduce("", String::concat);

System.out.println("Concatenated string: " + concatenated);

-------------------------------------------------------------------------------------------------------------

Q: Write a program to remove all whitespace from a string using Java 8 streams.

A:

String str = " hello world ";

String withoutWhitespace = str.chars()

.filter(c -> !Character.isWhitespace(c))

.mapToObj(c -> String.valueOf((char) c))

.collect(Collectors.joining());

System.out.println("String without whitespace: " + withoutWhitespace);

-------------------------------------------------------------------------------------------------------------

Q: Write a program to find the average of the squares of all odd numbers in an array using Java 8 streams.

A:

Integer[] numbers = {1, 2, 3, 4, 5};

OptionalDouble average = Arrays.stream(numbers)

.filter(n -> n % 2 != 0)

.mapToDouble(n -> Math.pow(n, 2))

.average();

if (average.isPresent()) {

System.out.println("Average of squares of odd numbers: " + average.getAsDouble());

} else {

System.out.println("Array doesn't have any odd numbers.");

}

-------------------------------------------------------------------------------------------------------------

Q: Write a program to find the sum of digits in a number using Java 8 streams.

A:

int number = 12345;

int sumOfDigits = String.valueOf(number)

.chars()

.map(Character::getNumericValue)

.sum();

System.out.println("Sum of digits: " + sumOfDigits);

-------------------------------------------------------------------------------------------------------------

Q: Write a program to remove all duplicate characters from a string using Java 8 streams.

A:

String str = "hello world";

String withoutDuplicates = str.chars()

.distinct()

.mapToObj(c -> String.valueOf((char) c))

.collect(Collectors.joining());

System.out.println("String without duplicates: " + withoutDuplicates);

-------------------------------------------------------------------------------------------------------------

Q: Write a program to find the frequency of each word in a string using Java 8 streams.

A:

String sentence = "Java is a programming language. Java is widely used.";

Map<String, Long> wordFrequency = Arrays.stream(sentence.split(" "))

.collect(Collectors.groupingBy(Function.identity(), Collectors.counting()));

System.out.println("Word frequency: " + wordFrequency);

-------------------------------------------------------------------------------------------------------------

Q: Write a program to find the sum of squares of all even numbers in an array using Java 8 streams.

A:

Integer[] numbers = {1, 2, 3, 4, 5};

int sumOfSquares = Arrays.stream(numbers)

.filter(n -> n % 2 == 0)

.mapToInt(n -> (int) Math.pow(n, 2))

.sum();

System.out.println("Sum of squares of even numbers: " + sumOfSquares);

-------------------------------------------------------------------------------------------------------------

Q: Write a program to reverse the order of words in a string using Java 8 streams.

A:

String sentence = "Java is a programming language";

String reversedSentence = Arrays.stream(sentence.split(" "))

.reduce((word1, word2) -> word2 + " " + word1)

.orElse("");

System.out.println("Reversed sentence: " + reversedSentence);

-------------------------------------------------------------------------------------------------------------

Q: Write a program to find the second most frequent character in a string using Java 8 streams.

A:

String str = "hello world";

Optional<Map.Entry<Character, Long>> secondMostFrequent = str.chars()

.mapToObj(c -> (char) c)

.collect(Collectors.groupingBy(Function.identity(), Collectors.counting()))

.entrySet()

.stream()

.sorted(Map.Entry.comparingByValue(Comparator.reverseOrder()))

.skip(1)

.findFirst();

if (secondMostFrequent.isPresent()) {

System.out.println("Second most frequent character: " + secondMostFrequent.get().getKey());

} else {

System.out.println("String doesn't have a second most frequent character.");

}

-------------------------------------------------------------------------------------------------------------

Q: Write a program to find the longest substring without repeating characters in a string using Java 8 streams.

A:

String str = "abcabcbb";

String longestSubstring = IntStream.range(0, str.length())

.mapToObj(i -> IntStream.range(i, str.length())

.mapToObj(j -> str.substring(i, j + 1))

.filter(sub -> sub.chars().distinct().count() == sub.length())

.max(Comparator.comparingInt(String::length))

.orElse(""))

.max(Comparator.comparingInt(String::length))

.orElse("");

System.out.println("Longest substring without repeating characters: " + longestSubstring);

-------------------------------------------------------------------------------------------------------------

Q: Write a program to find the median of an array using Java 8 streams.

A:

Integer[] numbers = {1, 3, 2, 5, 4};

OptionalDouble median;

int length = numbers.length;

if (length % 2 == 0) {

median = Arrays.stream(numbers)

.sorted()

.skip(length / 2 - 1)

.limit(2)

.mapToDouble(Integer::doubleValue)

.average();

} else {

median = Arrays.stream(numbers)

.sorted()

.skip(length / 2)

.findFirst()

.mapToDouble(Integer::doubleValue);

}

if (median.isPresent()) {

System.out.println("Median: " + median.getAsDouble());

} else {

System.out.println("Array is empty.");

}

-------------------------------------------------------------------------------------------------------------

Q: Write a program to find the common characters between two strings using Java 8 streams.

A:

String str1 = "hello";

String str2 = "world";

Set<Character> commonCharacters = str1.chars()

.mapToObj(c -> (char) c)

.filter(str2::contains)

.collect(Collectors.toSet());

System.out.println("Common characters: " + commonCharacters);

-------------------------------------------------------------------------------------------------------------

Q: Write a program to count the number of words in a string using Java 8 streams.

A:

String sentence = "Java is a programming language";

long wordCount = Arrays.stream(sentence.split(" ")).count();

System.out.println("Word count: " + wordCount);

-------------------------------------------------------------------------------------------------------------

Q: Write a program to check if a number is a palindrome using Java 8 streams.

A:

int number = 121;

boolean isPalindrome = IntStream.range(0, String.valueOf(number).length() / 2)

.allMatch(i -> String.valueOf(number).charAt(i) == String.valueOf(number).charAt(String.valueOf(number).length() - i - 1));

System.out.println("Is palindrome: " + isPalindrome);

-------------------------------------------------------------------------------------------------------------

Q: Write a program to find the first repeated character in a string using Java 8 streams.

A:

String str = "hello world";

Optional<Character> firstRepeated = str.chars()

.mapToObj(c -> (char) c)

.filter(c -> str.indexOf(c) != str.lastIndexOf(c))

.findFirst();

if (firstRepeated.isPresent()) {

System.out.println("First repeated character: " + firstRepeated.get());

} else {

System.out.println("String doesn't have any repeated characters.");

}

-------------------------------------------------------------------------------------------------------------

Q: Write a program to find the sum of all even digits in a number using Java 8 streams.

A:

int number = 123456;

int sumOfEvenDigits = String.valueOf(number)

.chars()

.map(Character::getNumericValue)

.filter(n -> n % 2 == 0)

.sum();

System.out.println("Sum of even digits: " + sumOfEvenDigits);

-------------------------------------------------------------------------------------------------------------

Q: Write a program to check if two strings are anagrams using Java 8 streams.

A:

String str1 = "listen";

String str2 = "silent";

boolean areAnagrams = str1.chars()

.sorted()

.mapToObj(c -> (char) c)

.collect(Collectors.toList())

.equals(str2.chars().sorted().mapToObj(c -> (char) c).collect(Collectors.toList()));

System.out.println("Are anagrams: " + areAnagrams);

-------------------------------------------------------------------------------------------------------------

Q: Write a program to find the factorial of a number using Java 8 streams.

A:

int number = 5;

int factorial = IntStream.rangeClosed(1, number)

.reduce(1, (a, b) -> a \* b);

System.out.println("Factorial: " + factorial);

-------------------------------------------------------------------------------------------------------------

Q: Write a program to remove all non-alphabetic characters from a string using Java 8 streams.

A:

String str = "hello123world";

String alphabeticOnly = str.chars()

.filter(Character::isLetter)

.mapToObj(c -> String.valueOf((char) c))

.collect(Collectors.joining());

System.out.println("Alphabetic only: " + alphabeticOnly);

-------------------------------------------------------------------------------------------------------------

Q: Write a program to find the third largest element in an array using Java 8 streams.

A:

Integer[] numbers = {1, 3, 2, 5, 4};

Optional<Integer> thirdLargest = Arrays.stream(numbers)

.distinct()

.sorted(Comparator.reverseOrder())

.skip(2)

.findFirst();

if (thirdLargest.isPresent()) {

System.out.println("Third largest element: " + thirdLargest.get());

} else {

System.out.println("Array doesn't have a third largest element.");

}

-------------------------------------------------------------------------------------------------------------

Q: Write a program to remove all vowels from a string using Java 8 streams.

A:

String str = "hello world";

String withoutVowels = str.chars()

.filter(c -> !"aeiou".contains(String.valueOf((char) c).toLowerCase()))

.mapToObj(c -> String.valueOf((char) c))

.collect(Collectors.joining());

System.out.println("String without vowels: " + withoutVowels);

-------------------------------------------------------------------------------------------------------------

Q: Write a program to check if an array is sorted in ascending order using Java 8 streams.

A:

Integer[] numbers = {1, 2, 3, 4, 5};

boolean isSortedAscending = IntStream.range(0, numbers.length - 1)

.allMatch(i -> numbers[i] <= numbers[i + 1]);

System.out.println("Is sorted ascending: " + isSortedAscending);

-------------------------------------------------------------------------------------------------------------

Q: Write a program to find the product of all elements in an array using Java 8 streams.

A:

Integer[] numbers = {1, 2, 3, 4, 5};

int product = Arrays.stream(numbers).reduce(1, (a, b) -> a \* b);

System.out.println("Product: " + product);

-------------------------------------------------------------------------------------------------------------

Q: Write a program to check if a string is a palindrome using Java 8 streams.

A:

String str = "madam";

boolean isPalindrome = IntStream.range(0, str.length() / 2)

.allMatch(i -> str.charAt(i) == str.charAt(str.length() - i - 1));

System.out.println("Is palindrome: " + isPalindrome);

-------------------------------------------------------------------------------------------------------------