java

```
List<Integer> list = Arrays.asList(1, 2, 2, 3);
list.stream().map(i -> i * i).distinct().forEach(System.out::println);
```

19. Use streams to collect a list of even numbers into a new list.

java

```
List<Integer> list = Arrays.asList(1, 2, 3, 4);
List<Integer> evenNumbers = list.stream().filter(i -> i % 2 ==
0).collect(Collectors.toList());
```

20. Use `reduce()` to compute the product of a list of integers.

java

```
List<Integer> list = Arrays.asList(1, 2, 3, 4);
int product = list.stream().reduce(1, (a, b) -> a * b);
```

These examples cover a wide range of Java 8 concepts and are commonly asked during technical interviews.

You said:

I was asked some.question on string as well like return the character that occur most number of times in a string using Java 8
So si.ikar programs on array or string can you include and make the list again which is better
Remake the list

ChatGPT said:

Memory updated

Here’s an updated list of 20 frequently asked Java 8 coding interview questions, including string and array-based problems with Java 8 solutions:

1. Sort a list of strings using Java 8 streams.

java

```
List<String> list = Arrays.asList("apple", "banana", "orange");
```

```
list.stream().sorted().forEach(System.out::println);
```

2. Filter a list of integers greater than a given value using streams.

```java
List<Integer> list = Arrays.asList(1, 2, 3, 4, 5, 6);
list.stream().filter(i -> i > 3).forEach(System.out::println);
```

3. Find the character that occurs the most number of times in a string.

```java
String str = "javajava";
Optional<Map.Entry<Character, Long>> maxChar = str.chars().mapToObj(c -> (char) c)
    .collect(Collectors.groupingBy(c -> c, Collectors.counting()))
    .entrySet().stream()
    .max(Map.Entry.comparingByValue());

maxChar.ifPresent(entry -> System.out.println(entry.getKey()));
```

4. Remove duplicates from an array of integers using Java 8 streams.

```java
int[] numbers = {1, 2, 2, 3, 4, 4, 5};
int[] distinctNumbers = Arrays.stream(numbers).distinct().toArray();
```

5. Find the maximum and minimum numbers in an array using Java 8 streams.

```java
int[] numbers = {1, 2, 3, 4, 5};
int max = Arrays.stream(numbers).max().getAsInt();
int min = Arrays.stream(numbers).min().getAsInt();
```

6. Convert a list of strings to uppercase using Java 8 streams.

```java
List<String> list = Arrays.asList("apple", "banana", "orange");
list.stream().map(String::toUpperCase).forEach(System.out::println);
```

7. Reverse a string using Java 8.

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

8. Count the occurrences of a specific character in a string.

```java
String str = "java";
long count = str.chars().filter(ch -> ch == 'a').count();
```

9. Join a list of strings with a delimiter using Java 8.

```java
List<String> list = Arrays.asList("apple", "banana", "orange");
String result = String.join(", ", list);
```

10. Group a list of objects by a field using `Collectors.groupingBy()`.

```java
List<Employee> employees = Arrays.asList(new Employee("John", "HR"), new Employee("Jane",
"IT"));
Map<String, List<Employee>> grouped = employees.stream()
        .collect(Collectors.groupingBy(Employee::getDepartment));
```

11. Check if two strings are anagrams using Java 8.

```java
String str1 = "listen";
String str2 = "silent";
boolean isAnagram = Arrays.equals(
    str1.chars().sorted().toArray(),
    str2.chars().sorted().toArray());
```

12. Remove vowels from a string using Java 8.

```java
String str = "java";
String result = str.chars().filter(c -> "aeiouAEIOU".indexOf(c) == -1)
    .mapToObj(c -> String.valueOf((char) c))
    .collect(Collectors.joining());
```

13. Find the first non-repeated character in a string using Java 8.

```java
String str = "swiss";
Optional<Character> firstNonRepeated = str.chars().mapToObj(c -> (char) c)
    .collect(Collectors.groupingBy(c -> c, LinkedHashMap::new, Collectors.counting()))
    .entrySet().stream().filter(entry -> entry.getValue() == 1)
    .map(Map.Entry::getKey).findFirst();
```

14. Find the distinct elements in an array using Java 8.

```java
int[] numbers = {1, 2, 2, 3, 3, 4};
int[] distinct = Arrays.stream(numbers).distinct().toArray();
```

15. Check if a string contains only digits using Java 8.

```java
String str = "12345";
boolean isDigit = str.chars().allMatch(Character::isDigit);
```

16. Convert a list of strings to a list of their lengths using Java 8.

```java
List<String> list = Arrays.asList("apple", "banana", "orange");
List<Integer> lengths = list.stream().map(String::length).collect(Collectors.toList());
```

17. Sum the values of a list of integers using streams.

```java
List<Integer> list = Arrays.asList(1, 2, 3, 4);
int sum = list.stream().mapToInt(Integer::intValue).sum();
```

18. Count the number of even numbers in an array using streams.

```java
int[] numbers = {1, 2, 3, 4};
long count = Arrays.stream(numbers).filter(n -> n % 2 == 0).count();
```

19. Find the second highest number in an array using Java 8.

```java
int[] numbers = {1, 2, 3, 4, 5};
OptionalInt secondHighest = Arrays.stream(numbers)
    .boxed().sorted(Comparator.reverseOrder())
    .skip(1).findFirst().mapToInt(Integer::intValue);
```

20. Use `reduce()` to concatenate a list of strings.

```java
List<String> list = Arrays.asList("apple", "banana", "orange");
String concatenated = list.stream().reduce("", (a, b) -> a + b);
```

This list combines both string and array manipulation, covering a wide range of concepts that are essential for Java 8 technical interviews.

Is this conversation helpful so far?

ChatGPT can make mistakes. Check important info.