1. **Convert a List of Integers to Their Squares**

Solution:

```
List<Integer> numbers = List.of(1, 2, 3, 4, 5);
List<Integer> squares = numbers.stream()
    .map(n -> n * n)
    .collect(Collectors.toList());
System.out.println("Squares: " + squares);
```

2. **Check if Any Element in a List Satisfies a Condition**

Solution:

```
List<Integer> numbers = List.of(1, 2, 3, 4, 5);
boolean anyEven = numbers.stream().anyMatch(n -
> n % 2 == 0);
System.out.println("Any Even: " + anyEven);
```

3. **Sort a List of Integers in Ascending Order**

Solution:

```
List<Integer> numbers = List.of(5, 3, 1, 4, 2);
List<Integer> sortedNumbers = numbers.stream()
    .sorted()
    .collect(Collectors.toList());
```

```
System.out.println("Sorted Numbers: " +
sortedNumbers);
```

4. **Sort a List of Integers in Descending Order**

Solution:

```
List<Integer> numbers = List.of(5, 3, 1, 4, 2);
List<Integer> sortedNumbers = numbers.stream()
    .sorted(Comparator.reverseOrder())
    .collect(Collectors.toList());
System.out.println("Sorted Numbers
(Descending): " + sortedNumbers);
```

5. **Check if a String is a Palindrome**

Solution:

```
String input = "madam";
boolean isPalindrome = IntStream.range(0,
input.length() / 2)
    .allMatch(i -> input.charAt(i) ==
input.charAt(input.length() - 1 - i));
System.out.println("Is Palindrome: " +
isPalindrome);
```

6. **Group Employees by Department and Calculate Average Salary**

Solution:

```
List<Employee> employees = List.of(
    new Employee("Alice", "HR", 50000),
    new Employee("Bob", "IT", 60000),
    new Employee("Charlie", "HR", 55000)
);
Map<String, Double> avgSalaryByDept =
employees.stream()
    .collect(Collectors.groupingBy(
        Employee::getDepartment,

Collectors.averagingDouble(Employee::getSalary)
    ));
System.out.println("Average Salary by
Department: " + avgSalaryByDept);
```