(1) Given a list of employees, find the names of employees who earn more than 50,000.

Input:

```
List<Employee> employees = Arrays.asList(
        new Employee("John", 60000),
        new Employee("Jane", 40000),
        new Employee("Jack", 70000)
);
```

Output: ["John", "Jack"]

(2) Given a list of integers, find the first even number greater than 10.

Input:

```
List<Integer> numbers = Arrays.asList(5, 12, 8, 21, 13);
```

Output: 12

(3) Given a list of strings, count how many strings start with the letter "A?

Input:

```
List<String> words = Arrays.asList("Apple", "Banana", "Avocado", "Cherry", "Apricot");
```

Output: 3

(4) Given a list of students, group them by their department.?

Input:

```
List<Student> students = Arrays.asList(
        new Student("Alice", "CS"),
        new Student("Bob", "Math"),
        new Student("Charlie", "CS")
);
```

Output:

```
{
  "CS": ["Alice", "Charlie"],
  "Math": ["Bob"]}
```

(5) Given a list of transactions, find the total amount of all transactions?

Input:

```
List<Transaction> transactions = Arrays.asList(
        new Transaction(100),
        new Transaction(200),
        new Transaction(300)
);
```

Output: 600

(6) Given a list of integers, find the maximum value.?

Input:

```
List<Integer> numbers = Arrays.asList(3, 7, 2, 9, 5);
```

Output: 9

(7) Given a list of strings, concatenate all the strings into a single string separated by commas.?

Input:

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

Output: apple,banana,cherry

(8) Given a list of strings, find the longest string.?

Input:

```
List<String> words = Arrays.asList("apple", "banana", "strawberry", "cherry");
```

Output: strawberry

(9) Given a list of integers, return a list of squares of the numbers.?

Input:

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

Output: [4, 9, 16]

(10) Given a list of employees, find the employee with the highest salary who works in the "IT" department?

Input:

```
List<Employee> employees = Arrays.asList(
        new Employee("John", "IT", 60000),
        new Employee("Jane", "HR", 40000),
        new Employee("Jack", "IT", 70000)
);
```

Output: Employee("Jack", "IT", 70000)

(11) Given a list of numbers, find the sum of all odd numbers after multiplying them by 2.?

Input:

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

Output: 18

(12) Given a list of employees, group them by department and find the average salary for each department. ?

Input:

```
List<Employee> employees = Arrays.asList(
        new Employee("John", "IT", 60000),
        new Employee("Jane", "HR", 50000),
        new Employee("Jack", "IT", 70000),
        new Employee("Jill", "HR", 40000)
);
```

Output:

```
{
  "IT": 65000.0,
  "HR": 45000.0
```

}

(13) Given a list of strings, find all distinct strings longer than 3 characters, sorted in descending order of length. ?

Input:

```
List<String> words = Arrays.asList("apple", "dog", "banana", "cat", "cherry");
```

Output: [banana, cherry, apple]

(14) Given a list of employees, find the employee with the second-highest salary.?

Input:

```
List<Employee> employees = Arrays.asList(
        new Employee("John", 60000),
        new Employee("Jane", 40000),
        new Employee("Jack", 70000),
        new Employee("Jill", 50000)
);
```

Output: Employee("John", 60000)

(15) Given a list of numbers, find the sum of the squares of all even numbers greater than 10.?

Input:

```
List<Integer> numbers = Arrays.asList(4, 12, 15, 20, 8, 18);
```

Output: 868

(16) Given a list of transactions, return a map where the keys are account types, and the values are the total amount of transactions for each type.?

Input:

```
List<Transaction> transactions = Arrays.asList(
        new Transaction("Savings", 200),
        new Transaction("Current", 500),
```

```
        new Transaction("Savings", 300),
        new Transaction("Current", 700)
    );
```

Output:

```
{
  "Savings": 500,
  "Current": 1200
}
```

(17) Given a list of numbers, find the product of the top 3 highest numbers.?

Input:

```
List<Integer> numbers = Arrays.asList(3, 7, 2, 9, 5, 6);
```

Output: 378

(18) Given a list of employees, group them by their department and find the employee with the highest salary in each department.?

Input:

```
List<Employee> employees = Arrays.asList(
        new Employee("John", "IT", 60000),
        new Employee("Jane", "HR", 50000),
        new Employee("Jack", "IT", 70000),
        new Employee("Jill", "HR", 40000)
    );
```

Output:

```
{
  "IT": Employee("Jack", "IT", 70000),
  "HR": Employee("Jane", "HR", 50000)
}
```

(19)

Input: from given list, print those fruits which containing exactly two vowels?

List<String> fruits = Arrays.asList("apple","banana","date","cherry","orange","mango");

Output: apple , date , mango

(20) Given a list of strings, find the three longest distinct strings and concatenate them.?

Input:

List<String> words = Arrays.asList("apple", "banana", "cherry", "dog", "elephant");

Output: elephantbananacherry

(21) Given a list of employees, find out if there is any employee with a salary greater than 100,000 and who works in the "Finance" department.?

Input:

```
List<Employee> employees = Arrays.asList(
        new Employee("John", "Finance", 120000),
        new Employee("Jane", "IT", 95000),
        new Employee("Jack", "Finance", 90000)
);
```

Output: true

(22) Given a list of students with their names and scores, find the top 3 students with the highest scores.?

Input:

```
List<Student> students = Arrays.asList(
        new Student("Alice", 85),
        new Student("Bob", 92),
        new Student("Charlie", 88),
        new Student("David", 95),
        new Student("Eva", 91)
```

);

Output: [Student("David", 95), Student("Bob", 92), Student("Eva", 91)]

30 FILTER QUESTION :

10 EASY:

(1) Question: Given a list of integers, filter out the even numbers.

- **Input**: `[1, 2, 3, 4, 5]`
- **Output**: `[1, 3, 5]`

(2) Question: Given a list of strings, filter out strings with length greater than 3.

- **Input**: `["apple", "bat", "car", "dog"]`
- **Output**: `["bat", "car", "dog"]`

(3) Question: Given a list of names, filter out names that start with 'A'.

- **Input**: `["Alice", "Bob", "Charlie"]`
- **Output**: `["Bob", "Charlie"]`

(4) Question: Given a list of integers, filter numbers greater than 10.

- **Input**: `[5, 10, 15, 20]`
- **Output**: `[15, 20]`

(5) Question: Given a list of products, filter those that are in stock (quantity > 0).

- **Input**: `[{name: "Laptop", quantity: 0}, {name: "Phone", quantity: 5}]`
- **Output**: `[{name: "Phone", quantity: 5}]`

(6) Question: Given a list of ages, filter those that are legal adults (age ≥ 18).

- **Input**: `[15, 20, 17, 22]`
- **Output**: `[20, 22]`

(7) Question: Given a list of employees, filter those who work in the IT department.

- **Input**: `[{name: "Alice", department: "HR"}, {name: "Bob", department: "IT"}]`
- **Output**: `[{name: "Bob", department: "IT"}]`

(8) Question: Given a list of strings, filter out null or empty strings.

- **Input**: `["apple", "", null, "banana"]`
- **Output**: `["apple", "banana"]`

(9) Question: Given a list of books, filter out those with a rating above 4.0.

- **Input**: `[{title: "Book1", rating: 4.5}, {title: "Book2", rating: 3.9}]`
- **Output**: `[{title: "Book1", rating: 4.5}]`

(10) Question: Given a list of temperatures, filter out those above freezing (temp > 0).

- **Input**: `[10, -5, 20, -15]`
- **Output**: `[10, 20]`

MEDIUM

(1) Print the multiplication of alternate number in an array.

Input: int arr []= {1,43,2,25,1,53,6};

Output: 12

(2) WAF to multiply 1st and last element and 2nd element to 2nd last element in a given array?

Input: int arr[]={4,5,7,2,1,9};

Output: 36,5

(3) Question: Given a list of students, filter those who scored more than 70 in Math.?

Input:

```
List<E_02> list = Arrays.asList(
        new E_02("Alice", 85),
        new E_02("Bob", 65)
);
```

Output: E_02{name='Alice', score=85}

(4) Question: Given a list of employees, filter out those with a salary greater than 50,000.

Input:

```
List<E_02> list = Arrays.asList(
        new E_02("Alice", 40000),
        new E_02("Bob", 55000)
);
```

Output: E_02{name='Bob', score=55000}

(5) Question: Given a list of strings, filter out those that contain the letter 'e'.?

Input:

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

Output: bat , dog

(6) Question: Given a list of people, filter out those whose names contain more than 5 characters.?

Input: List<String> list = Arrays.*asList*("Alice", "Bob", "Charlie");

Output: Alice, Bob

(7) Question: Given a list of countries, filter out countries that are located in Europe.?

Input:

List<Country> countries = Arrays.asList(

new Country("Germany", "Europe"),

new Country("USA", "North America"));

Output: E_02{name='Germany', continent=Europe}

(8) Question: Given a list of students with their grades in multiple subjects, filter those who have passed all subjects (grade ≥ 40).?

Input:

```
List<E_02> students = Arrays.asList(
        new E_02("Alice", Arrays.asList(45, 50, 35)),
        new E_02("Bob", Arrays.asList(50, 60, 45))
);
```

Output: [E_02{name='Bob', grades=[50, 60, 45]}]

(9) Question: Given a list of books with multiple authors, filter those books that have more than 1 author.?

Input:

```
List<E_02> authors = Arrays.asList(
        new E_02("Alice", Arrays.asList("Author1")),
        new E_02("Bob", Arrays.asList("Author1", "Author2"))
);
```

Output: [E_02{name='Bob', authors=[Author1, Author2]}]

HARD

(1) Question: Given a list of customers with their orders, filter out customers who have ordered more than 3 times?

Input:

```
List<E_03> list = Arrays.asList(
        new E_03("Alice", 2),
        new E_03("Bob", 5)
);
```

Output: [E_03{name='Bob', orders=5}]

(2) Question: Given a list of events with their start and end times, filter those that last more than 3 hours.?

Input:

```
List<E_03> list = Arrays.asList(
        new E_03("Event1", 2, 4),
        new E_03("Event2", 2, 6)
);
```

Output: [E_03{name='Event2', start=2, end=6}]

(3) Question: Given a list of employees with their attendance data, filter out employees who have less than 90% attendance.?

Input:

```
List<E_03> list = Arrays.asList(
        new E_03("Alice", 95),
        new E_03("Bob", 85)
);
```

Output: [E_03{name='Alice', attendance=95}]

(4) Question: Given a list of projects with their start and end dates, filter projects that lasted more than 1 year.?

Input:

```
List<E_03> list = Arrays.asList(
        new E_03("Project1", "2020-01-01", "2020-12-31"),
        new E_03("Project2", "2020-01-01", "2021-06-30")
);
```

Output: [E_03{name='Project2', start='2020-01-01', end='2021-06-30'}]

(5) Question: Given a list of students with multiple test scores, filter out those whose average score is above 70.?

Input:

```
List<E_03> students = Arrays.asList(
        new E_03("Alice", Arrays.asList(60, 70, 80)),
        new E_03("Bob", Arrays.asList(50, 60, 65))
);
```

Output: [E_03{name='Alice', scores=[60, 70, 80]}]