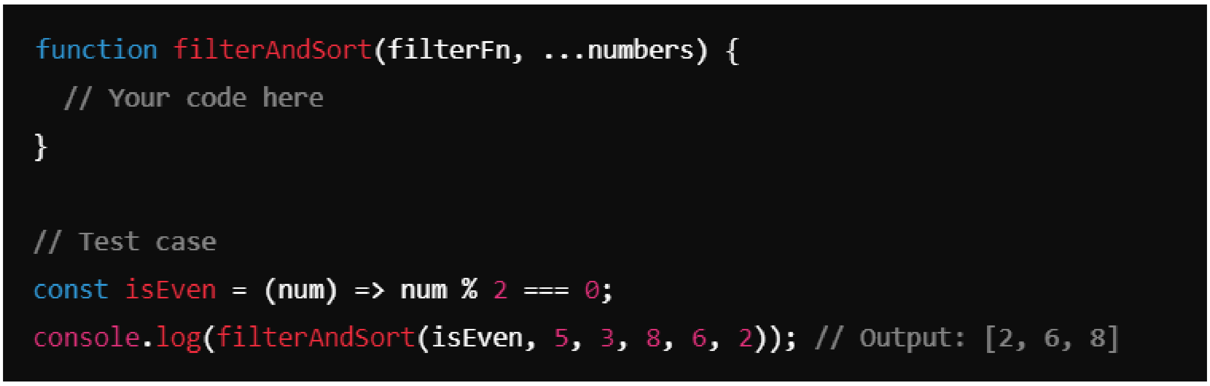
Exercise: Advanced Use of Rest and Spread Operators

Problem Statement

You are tasked with creating a utility library for handling various operations on arrays and objects. You need to write several functions that use both the rest and spread operators.

Instructions

1. **Function Definitions:**
    - **filterAndSort Function:** Create a function called filterAndSort that takes an indefinite number of arguments. The first argument should be a filter function, and the remaining arguments should be numbers. The function should:
        1. Use the rest operator to capture all arguments except the first one.
        2. Filter the numbers using the provided filter function.
        3. Sort the filtered numbers in ascending order.
        4. Return the sorted array of numbers.

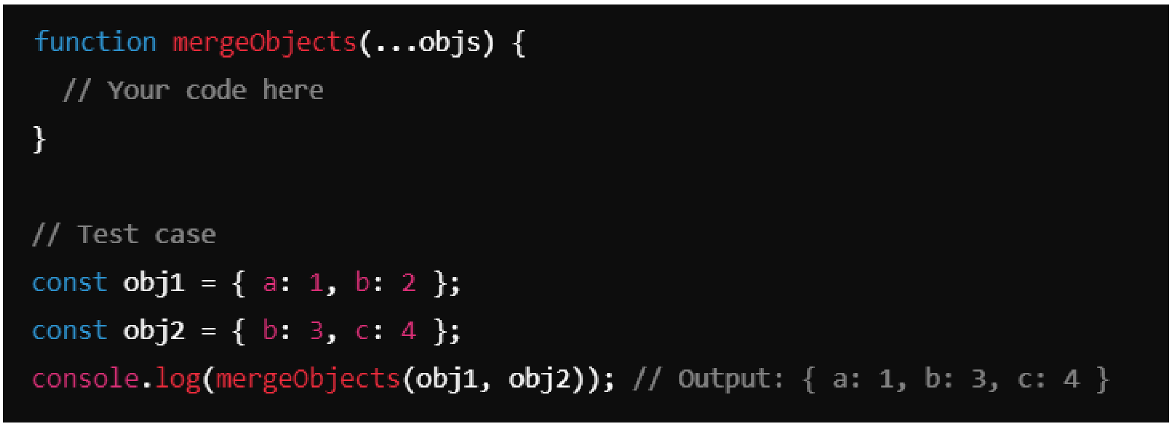


```
function filterAndSort(filterFn, ...numbers) {
  // Your code here
}

// Test case
const isEven = (num) => num % 2 === 0;
console.log(filterAndSort(isEven, 5, 3, 8, 6, 2)); // Output: [2, 6, 8]
```

2 : mergeObjects Function: Create a function called mergeObjects that accepts an indefinite number of objects. The function should:

1. Use the spread operator to merge all objects into one.
2. Ensure that if multiple objects contain the same key, the last object's value for that key should be used.
3. Return the merged object.

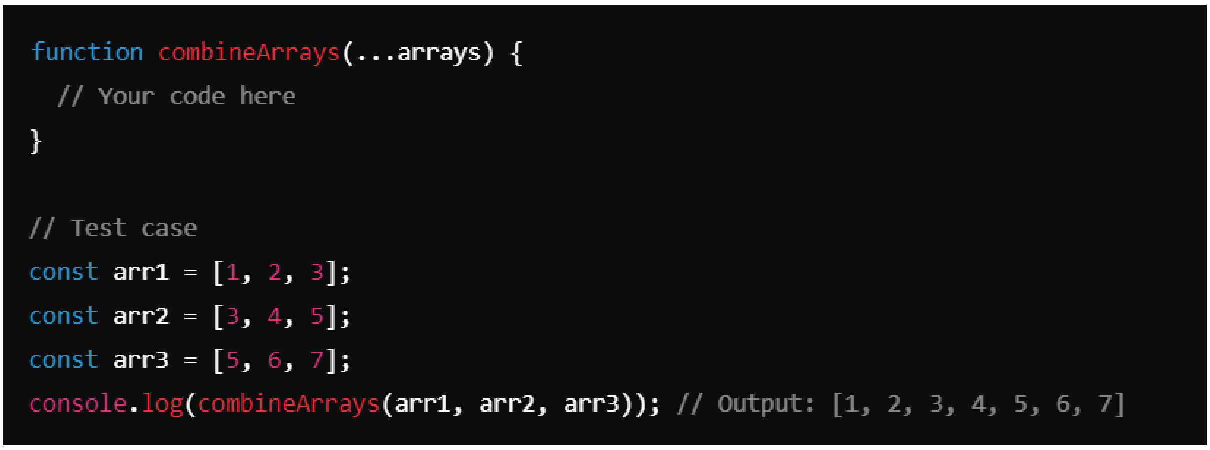


```
function mergeObjects(...objs) {
  // Your code here
}

// Test case
const obj1 = { a: 1, b: 2 };
const obj2 = { b: 3, c: 4 };
console.log(mergeObjects(obj1, obj2)); // Output: { a: 1, b: 3, c: 4 }
```

3: combineArrays Function: Create a function called combineArrays that takes multiple arrays as arguments. The function should:

1. Use the spread operator to combine all arrays into one.
2. Remove duplicate elements from the combined array.
3. Return the new array with unique elements.



```
function combineArrays(...arrays) {
  // Your code here
}

// Test case
const arr1 = [1, 2, 3];
const arr2 = [3, 4, 5];
const arr3 = [5, 6, 7];
console.log(combineArrays(arr1, arr2, arr3)); // Output: [1, 2, 3, 4, 5, 6, 7]
```

4. extractProperties Function

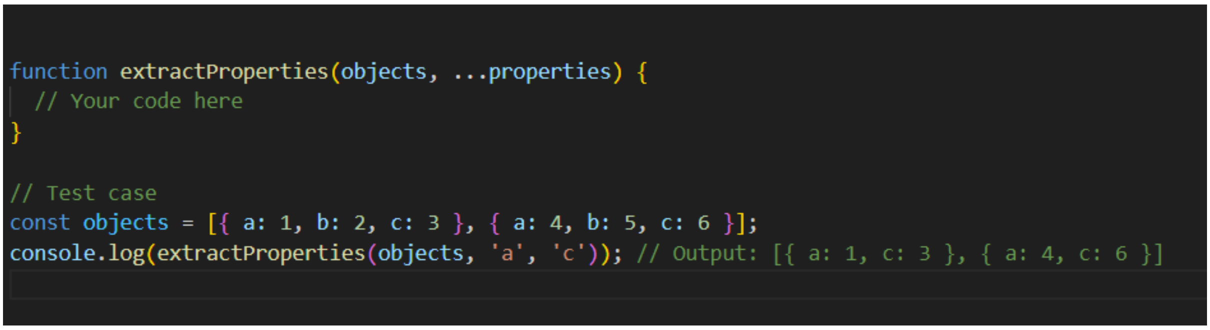
Create a function named extractProperties that accepts the following:

- An array of objects.
- A list of property names.

The function should:

- Use the spread operator to create a new array of objects with only the specified properties.

- Return the new array of objects.



```
function extractProperties(objects, ...properties) {
  // Your code here
}

// Test case
const objects = [{ a: 1, b: 2, c: 3 }, { a: 4, b: 5, c: 6 }];
console.log(extractProperties(objects, 'a', 'c')); // Output: [{ a: 1, c: 3 }, { a: 4, c: 6 }]
```