Question 1: Reverse an Array
Problem: Write a function that takes an array and returns a new array with the elements in reverse order.
Input: [1, 2, 3, 4, 5]
Output: [5, 4, 3, 2, 1]
Use Case: This function can be used in a web application where user reviews need to be displayed in reverse chronological order.

```
function revarray(damon) {

return damon.slice().reverse()

}

const a = [1,2,3,4,5]

const b = revarray(a)

console.log(b)
```

Question 2: Flatten an Array
Problem: Write a function that takes a nested array and flattens it to a single-level array.
Input: [1, [2, 3], [4, [5]]]
Output: [1, 2, 3, 4, 5]
Use Case: Useful for aggregating user-selected items from multiple categories into a single list for checkout.

```
const a = [1,[2,3,[4,5],6],7]

const b = a.flat(Infinity)

console.log(b)
```

Question 3: Check for Duplicates

Problem: Write a function that checks if an array contains duplicates.
Input: [1, 2, 3, 4, 5, 1]
Output: true
Input: [1, 2, 3, 4, 5]
Output: false
Use Case: Can be used to validate user inputs in forms, such as ensuring usernames are unique during registration.

```
function findDuplicate(arr)
{
const seen = new Set()
for( const item of arr) {
if(seen.has(item)) {
return true }
seen.add(item) }
return false
}
console.log(findDuplicate([1,2,2,3,4,5,5]))
console.log(findDuplicate([1,2,3,4,5]))
```

Question 4: Merge Two Objects
Problem: Write a function that merges two objects into one.
Input: { a: 1, b: 2 }, { b: 2, c: 4 }
Output: { a: 1, b: 2, c: 4 }
Use Case: This can be used in a web application to combine user profile settings from different sources.

```
function merger(cas1,cas2) {
 return Object.assign({}, cas1, cas2)
}
 const cas1 = { a: 1, b: 2 }
 const cas2 = { b: 2, c: 4 }
 const mergedObject = merger(cas1, cas2)
console.log(mergedObject)
```

Question 5: Find the Maximum Number in an Array
Problem: Write a function that finds the maximum number in an array.
Input: [1, 3, 2, 8, 5]
Output: 8
Use Case: This function can help in analytics dashboards to find the highest sales figure or user activity.

```
function getBig(ant) {
  return Math.max(...ant)
}
const numbers = [1, 3, 2, 8, 5]
const maxNumber = getBig (numbers)
console.log(maxNumber)
```

Question 6: Group Array of Objects by Property
Problem: Write a function that groups an array of objects by a specific property.
Input: [{ id: 1, category: 'fruit' }, { id: 2, category: 'vegetable' }, { id: 3, category: 'fruit' }]
Output: {
fruit: [{ id: 1, category: 'fruit' }, { id: 3, category: 'fruit' }],
vegetable: [{ id: 2, category: 'vegetable' }]
}
Use Case: Useful for organizing products by category in an e-commerce application.

```
function groupByProperty(arr, property) {
  return arr.reduce((grouped, item) => {
    const key = item[property];      if (!grouped[key]) {
      grouped[key] = [];      }
    grouped[key].push(item);      return grouped;
  }, {});  }

const data = [
  { id: 1, category: 'fruit' },
  { id: 2, category: 'vegetable' },
  { id: 3, category: 'fruit' }
];

const grouped = groupByProperty(data, 'category');
console.log(grouped);
```

Question 7: Find the Intersection of Two Arrays
Problem: Write a function that returns the intersection of two arrays.
Input: [1, 2, 3], [2, 3, 4]
Output: [2, 3]
Use Case: This can be used in social media applications to find mutual friends between users.

```
function findIntersection(arr1, arr2) {
  const set1 = new Set(arr1);
  const set2 = new Set(arr2);

  return [...set1].filter(item => set2.has(item))
}

const array1 = [1, 2, 3]
const array2 = [2, 3, 4]

const intersection = findIntersection(array1, array2)
console.log(intersection)
```

Question 8: Calculate the Sum of Array Elements
Problem: Write a function that calculates the sum of all numbers in an array.
Input: [1, 2, 3, 4, 5]
Output: 15
Use Case: Useful in financial applications to calculate the total expenses or revenue.

```
function sumArray(arr) {
  return arr.reduce((sum, num) => sum + num, 0)
}
const numbers = [1, 2, 3, 4, 5]
const totalSum = sumArray(numbers)
 console.log(totalSum)
```

Question 9: Remove Falsy Values from an Array
Problem: Write a function that removes all falsy values from an array.
Input: [0, 1, false, 2, '', 3]
Output: [1, 2, 3]
Use Case: This function can be used to clean up user inputs or configuration arrays.

```
function removeFalsyValues(arr) {
  return arr.filter(Boolean)
}
const array = [0, 1, false, 2, '', 3]
const cleanedArray = removeFalsyValues(array)
console.log(cleanedArray)
```

Question 10: Calculate Average of an Array
Problem: Write a function that calculates the average of all numbers in an array.
Input: [1, 2, 3, 4, 5]
Output: 3
Use Case: This function is useful in educational applications where you need to compute the average score of students from an array of their grades.

```
function calculateAverage(arr) {
  const sum = arr.reduce((acc, num) => acc + num, 0)
  return sum / arr.length
}

const numbers = [1, 2, 3, 4, 5]
const average = calculateAverage(numbers)
console.log(average)
```