**1.What will the following code output?**

**const arr = [1, 2, 3, 4];**

**const result = arr.map((num, index, array) => array[index] \* 2);**

**console.log(result);**

**a) `[1, 2, 3, 4]`**

**b) `[2, 4, 6, 8]`**

**c) `[4, 8, 12, 16]`**

**d) `undefined**

**2. What will the result of this code be?**

**const arr = [10, 20, 30];**

**const result = arr.map((num) => num.toString());**

**console.log(result);**

**a) `['10', '20', '30']`**

**b) `[10, 20, 30]`**

**c) `['10', 20, '30']`**

**d) `undefined`**

**3. \*\*What happens if `map` callback does not return a value?\*\***

**const arr = [1, 2, 3];**

**const result = arr.map((num) => {**

**console.log(num);**

**});**

**console.log(result);**

**a) `undefined`**

**b) `[undefined, undefined, undefined]`**

**c) `[null, null, null]`**

**d) `[]**

**4. \*\*What is logged by the following code?\*\***

**const arr = [10, 20, 30];**

**arr.forEach(num => console.log(num \* 2));**

1. **Logs `10, 20, 30`**

**b) Logs `20, 40, 60`**

**c) Logs `undefined`**

**d) Logs `[20, 40, 60]`**

**5. \*\*What happens if you try to break out of a `forEach` loop using `break`?\*\***

**a) The loop breaks immediately.**

**b) The loop throws an error.**

**c) The loop continues without interruption.**

**d) The loop skips the current iteration.**

**6. \*\*What will the following code output?\*\***

**const arr = [1, 2, 3];**

**const result = arr.forEach(num => num \* 2);**

**console.log(result);**

**a) `[2, 4, 6]`**

**b) `[1, 2, 3]`**

1. **`undefined`**

**7. \*\*Can `forEach` be used to transform an array?\*\***

**a) Yes, directly.**

**b) No, because `forEach` does not return a value.**

**c) Yes, but only with `return` inside the callback.**

1. **No, because it only works on strings.**

**8. \*\*How many times does the callback execute in the following code?\*\***

**const arr = [1, 2, 3];**

**arr.forEach((num, index) => {**

**if (index === 1) return;**

**console.log(num);**

**});**

**a) 1**

**b) 2**

**c) 3**

1. **None**

**9. \*\*Which of the following code chains `map` and `forEach` correctly?\*\***

**const arr = [1, 2, 3];**

**a) `arr.map(num => num \* 2).forEach(num => num \* 3);`**

**b) `arr.map(num => num \* 2).forEach(num => console.log(num));`**

**c) `arr.forEach(num => num \* 2).map(num => console.log(num));`**

**d) Both b) and c)**

**10. \*\*What happens if the callback function passed to `map` throws an error?\*\***

**a) The `map` loop stops immediately.**

**b) The error is caught silently.**

**c) The error propagates.**

1. **The corresponding entry in the result array is `undefined`.**

**11. \*\*What does this code output?\*\***

**const arr = ['a', 'b', 'c'];**

**const result = arr.map((char, index) => index + char);**

**console.log(result);**

**a) `['0a', '1b', '2c']`**

**b) `['a0', 'b1', 'c2']`**

**c) `['a1', 'b2', 'c3']`**

**d) `undefined`**

**12.What will this code output?**

**const arr = [[1, 2], [3, 4]];**

**const result = arr.map(subArr => subArr.map(num => num \* 2));**

**console.log(result)**

**a) `[2, 4, 6, 8]`**

**b) `[[1, 2], [3, 4]]`**

**c) `[[2, 4], [6, 8]]`**

**d) `[2, 4]`**

**13. What is the output of this code?**

**const arr = [**

**{ id: 1, value: 10 },**

**{ id: 2, value: 20 },**

**];**

**const result = arr.map(obj => ({ ...obj, value: obj.value \* 2 }));**

**console.log(result);**

**a)  [{ id: 1, value: 20 }, { id: 2, value: 40 }]**

**b)   [{ id: 1, value: 10 }, { id: 2, value: 20 }]**

**c)  [{ id: 1, value: 2 }, { id: 2, value: 4 }]     d) `undefined`**

**Question 1: Extract the First Character of Each Word**

Write a JavaScript function that uses charAt to extract the first character of each word in the array ["apple", "banana", "cherry"]. What is the resulting array?

o/p: ['a', 'b', 'c']

let array=["apple", "banana", "cherry"]

let a=array.map(function(a){

    return a.charAt(0)

})

console.log(a)

output:-

[ 'a', 'b', 'c' ]

### Question 2: Get Unicode Values of Last Characters

Use charCodeAt to find the Unicode value of the last character in each word of the array ["dog", "cat", "bird"]. What does the resulting array look like?

o/p: [103, 116, 100]

let array=["dog","cat","bird"]

let a=array.map(function(arr){

return arr.charCodeAt()

})

console.log(a)

output:-

[ 100, 99, 98 ]

### Question 3: Shift the First Character of Each Word

Use charCodeAt and String.fromCharCode to replace the first character of each word in ["frog", "duck", "goose"] with the next letter in the alphabet. What is the modified array?

o/p: ['grog', 'euck', 'hoose']

let arr=["frog", "duck", "goose"]

let a=arr.map(function(x){

    return String.fromCharCode(x.charCodeAt(0) + 1) + x.slice(1);

})

console.log(a)

output:-

[ 'grog', 'euck', 'hoose' ]

### Question 4: Capitalize Even Unicode Characters

Write a JavaScript program that iterates through each character of words in the array ["zebra", "lion", "tiger"]. Use charCodeAt to check if the Unicode value of a character is even, and convert such characters to uppercase. What does the transformed array look like?

o/p: ['ZeBRa', 'LIoN', 'TIGeR']

let arr = ["zebra", "lion", "tiger"];

let result = arr.map(function (word) {

  return word

    .split("")

    .map(function (char) {

      return char.charCodeAt(0) % 2 === 0 ? char.toUpperCase() : char;

    })

    .join("");

});

console.log(result);

output:-

[ 'grog', 'euck', 'hoose' ]

Using charAt, reverse the characters in each word of the array ["bat", "cat", "hat"]. What is the final array after the reversal?

o/p: ['tab', 'tac', 'tah']

let a=["bat", "cat", "hat"]

let b=a.map(function(num){

    return num.split("").reverse().join("");

})

console.log(b)

output:-

[ 'tab', 'tac', 'tah' ]