Assignment Day 1:

Q1.)

1. find() method:

The `find()` method returns the value of the first element in an array that satisfies a provided testing function. If no elements satisfy the testing function, `undefined` is returned.

- Parameters:

- element: The current element being processed in the array.

- index (optional): The index of the current element being processed.

- array (optional): The array `find` was called upon.

- thisArg (optional): An object to which the `this` keyword can refer in the callback.

- Example:

const array1 = [5, 12, 8, 130, 44];

const found = array1.find(element => element > 10);

console.log(found); // 12

2. findIndex():

The findIndex() method returns the index of the first element in an array that satisfies the provided testing function. If no elements satisfy the testing function, `-1` is returned.

- Parameters:

element: The current element being processed in the array.

index (optional): The index of the current element being processed.

array (optional): The array `findIndex` was called upon.

thisArg (optional): An object to which the `this` keyword can refer in the callback.

- Example:

const array1 = [5, 12, 8, 130, 44];

const isLargeNumber = (element) => element > 13;

console.log(array1.findIndex(isLargeNumber)); // 3

3. filter():

- The filter() method creates a new array with all elements that pass the test implemented by the provided function. Essentially, if the function returns `true`, the element is included in the new array.

- Here's the syntax:

let newArray = array.filter(function(element, index, array) {

// Return true to keep the element, false otherwise

}, thisArg);

- Parameters:

element: The current element being processed in the array.

index (optional): The index of the current element being processed.

array (optional): The array `filter` was called upon.

thisArg (optional): An object to which the `this` keyword can refer in the callback.

- Example:

const words = ['spray', 'limit', 'elite', 'exuberant', 'destruction', 'present'];

const result = words.filter(word => word.length > 6);

console.log(result); // ["exuberant", "destruction", "present"]

Q 2.)

Each of these methods provides a different way to search through arrays and return information based on custom criteria defined in a callback function.

The map() method in JavaScript is used with arrays to create a new array by calling a function on every element of the original array.

This method is very useful for applying a specific operation to each item of an array and creating a new array with the results

element: The current element being processed in the array.

index (optional): The index of the current element being processed in the array.

array (optional): The array map was called upon.

thisArg (optional): Value to use as this when executing function.