TASK-JSmethods

```
let a=[1,2,3[1,2[1,2]],4,5]
a.splice(6)
console.log(a)
let x=a.splice(2,3)
console.log(x);
x=a.flat()
console.log(x)
```

****

```
let b=[1,2,3,4,5]
let x=b.slice(2)
console.log(x)
```

The slice() method slices out a piece of an array.

****

```
let x=[1,2,3,4,5,6]
a=x.splice(2,3,"hi","rishi")
console.log(a);//[3,4,5]
console.log(x)//[1,2,"hi","rishi"]
```

The splice() method adds new items to an array.

The first parameter (2) defines the position where new elements should be added (spliced in).

The second parameter (3) defines how many elements should be removed.

The rest of the parameters ("hi" , "rishi") define the new elements to be added.

The splice() method returns an array with the deleted items:

****

```
const myArr = [1,2,[3,4,5],[7,[8,9],10],11];

const newArr = myArr.flat(7);
console.log(newArr); // [1, 2, 3, 4, 5,6,7,8,9,10,11]
```

Flattening an array is the process of reducing the dimensionality of an array.

Flattening is useful when you want to convert a multi-dimensional array into a one-dimensional array.

****

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

delete a[2]

console.log(a)[1,2,3,<1 empty space>,5]

it deletes the index value which we have passed in the delete function and returns the array with the index value deleted the space will be created in the array it will be empty

****

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

x=a.indexOf(8)

console.log(x);//-1

The indexOf() method returns its index of the element specified in the array.if not found it returns -1

****

lastindex()

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

x=a.lastIndexOf(4)

console.log(x);//4

The lastIndexOf() method returns the index of the last occurrence of the specified value, or -1 if it is not present.

****

include()

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

x=a.includes(4)

console.log(x);//true

The includes() method returns true if a sequence of characters (length 2 or more) within a string can be found, starting the search at the specified location.if not found it returns false

****

map()

```
let arr=[1,2,3,4,5]

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

let x=arr.map(a=>a)

let x=arr.map(a=>a+"hi")

console.log(x);//["hi","hi","hi","hi","hi"]
```

The map() method creates a new array populated with the results of calling a provided function on every element in the calling array.

****

1)

```
 let arr=[1,2,3,4,5]

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

{

   return Math.pow(a,2);

   return a * a;

})

console.log(x);//[1,4,9,16,25]
```

****

2)

```
 let arr=["a","b","c"]

 let x=arr.map(a=>a.touppercase)

 console.log(x);
```

The map() method creates a new array populated with the results of calling a provided function on every element in the calling array.

****