Copy by value a composite data type.

What changes do we make to our code, so that changes in one array doesn't affect second array.

We can use the concept of **SPREAD OPERATOR (…)** to overcome this. To have a better understanding, let's look at below example.

```
let arr1 = [1,2,3];
let arr2 = [...arr1];

console.log(arr1);  //prints [ 1, 2, 3 ]

arr2.push(4);

console.log(arr1);  // prints [ 1, 2, 3 ]
console.log(arr2);  // prints [ 1, 2, 3, 4 ]
```

So, **arr2** *makes a copy of its own in the form of* **[…arr1]**. *What* **[…arr1]** *does here is, it takes in an array* **arr1** *and expands it into individual elements, and again it takes the form of array and gets assigned to* **arr2.**

So, the **arr1** *doesn't get altered even after we make changes to array* **arr2**, *after the usage of spread operator.*