2. Give a write-up on the Difference between copy by value and copy by reference.

Pass by Value:

In Pass by Value, Function is called by directly passing the value of the variable as the argument. Changing the argument inside the function doesn’t affect the variable passed from outside the function.

Javascript always pass by value so changing the value of the variable never changes the underlying primitive (String or number).

However, when a variable refers to an object which includes array, the value is the reference to the object.

Pass by Reference:

In Pass by Reference, Function is called by directly passing the reference/address of the variable as the argument. Changing the argument inside the function affect the variable passed from outside the function. In Javascript objects and arrays follows pass by reference.

so if we are passing object or array as an argument to the method, then there is a possibility that value of the object can change.

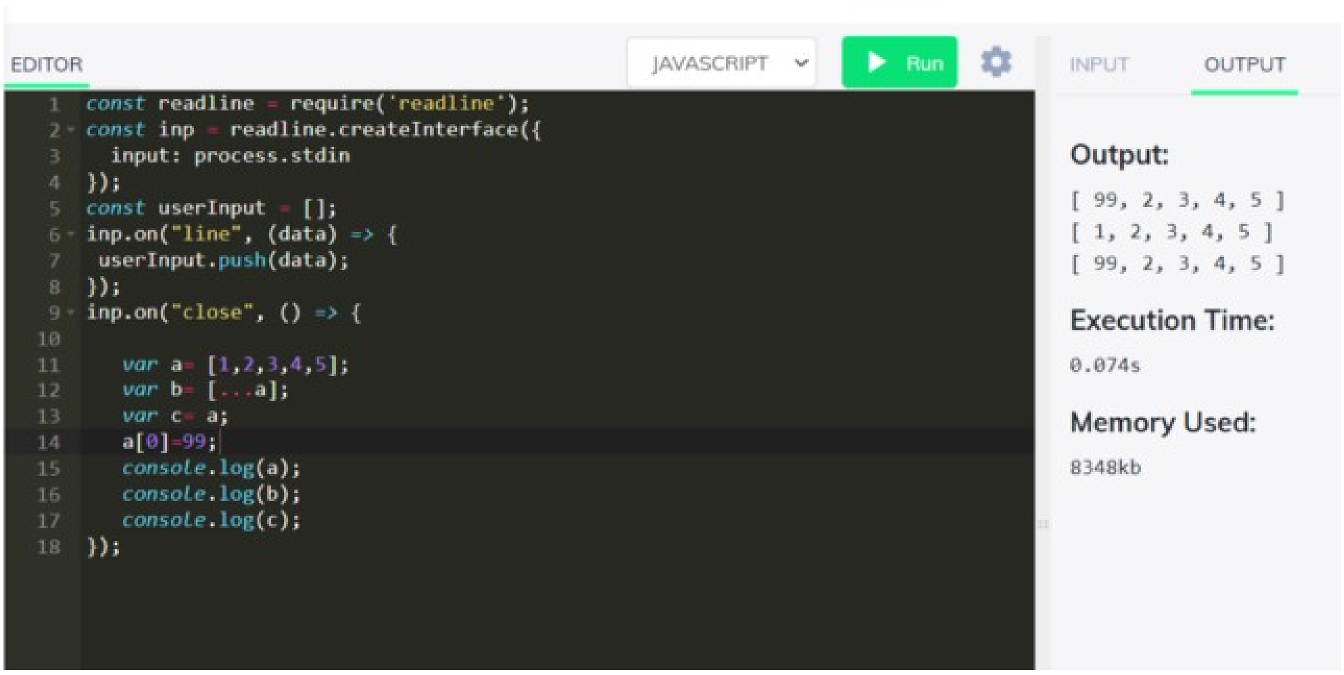
3. How to copy by value a composite datatype (array+objects).

Arrays, objects, functions are all of object type which comes under composite data types. As we know variable holds data in case of composite data type it holds reference that is address of that particular value in memory.

var a= 10; // here a holds the value 10.

var b= [10,20] // b holds some address like 8023 etc.. not 10 and 20.

Therefore we can’t clone data in composite data types. To do that spread operator is used, that is three dots (…), it spreads the elements of that particular array or object and its values can be used to assign to some other variable.



```javascript
const readline = require('readline');
const inp = readline.createInterface({
  input: process.stdin
});
const userInput = [];
inp.on("line", (data) => {
 userInput.push(data);
});
inp.on("close", () => {

   var a= [1,2,3,4,5];
   var b= [...a];
   var c= a;
   a[0]=99;
   console.log(a);
   console.log(b);
   console.log(c);
});
```

Output:

```
[ 99, 2, 3, 4, 5 ]
[ 1, 2, 3, 4, 5 ]
[ 99, 2, 3, 4, 5 ]
```

Execution Time:

0.074s

Memory Used:

8348kb