

Algoritmo III

☰ Tags	CSS & CSS3 Opcional Práctico
⚙ Status	Done

```
function a(x,y){
  return 5;
}
console.log(a(5,5))
```

var	value
x	5
y	5

Output esperado: 5

Output de la consola:

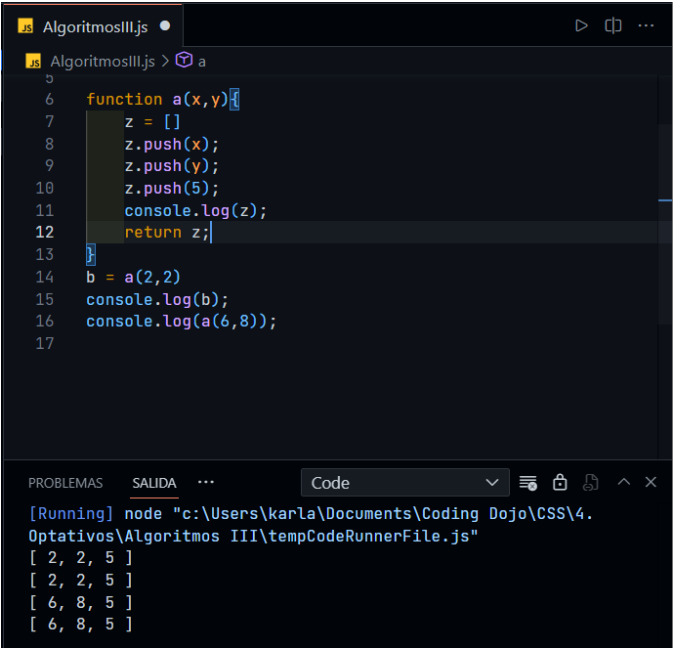


```
function a(x,y){
  z = []
  z.push(x);
  z.push(y);
  z.push(5);
  console.log(z);
  return z;
}
b = a(2,2)
console.log(b);
console.log(a(6,8));
```

var	value
x	2 ⇒ 6
y	2 ⇒ 8
z	[2, 2, 5] ⇒ [6, 8, 5]

Output esperado: [2, 2, 5] [2, 2, 5] [6, 8, 5] [6, 8, 5]

Output de la consola

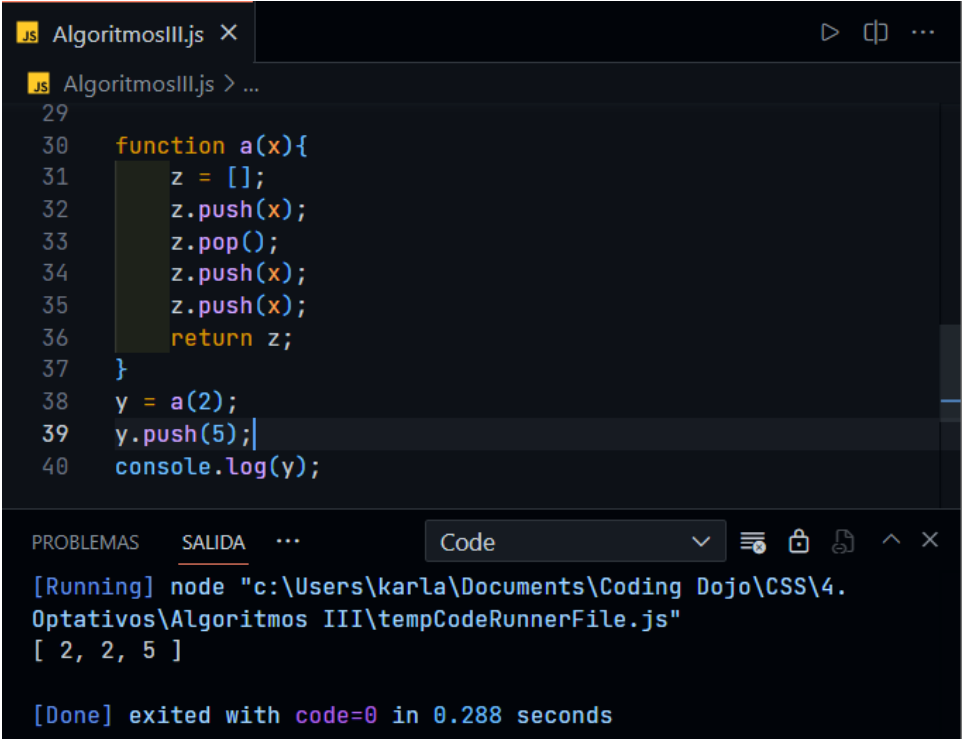


```
function a(x){
  z = [];
  z.push(x);
  z.pop();//quita el ultimo num de arr
  z.push(x);
  z.push(x);
  return z;
}
y = a(2);
y.push(5);
console.log(y);
```

var	value
z	[2, 2, 5]

Output esperado: [2,2,5]

Output de la consola:



```
function a(x){
```

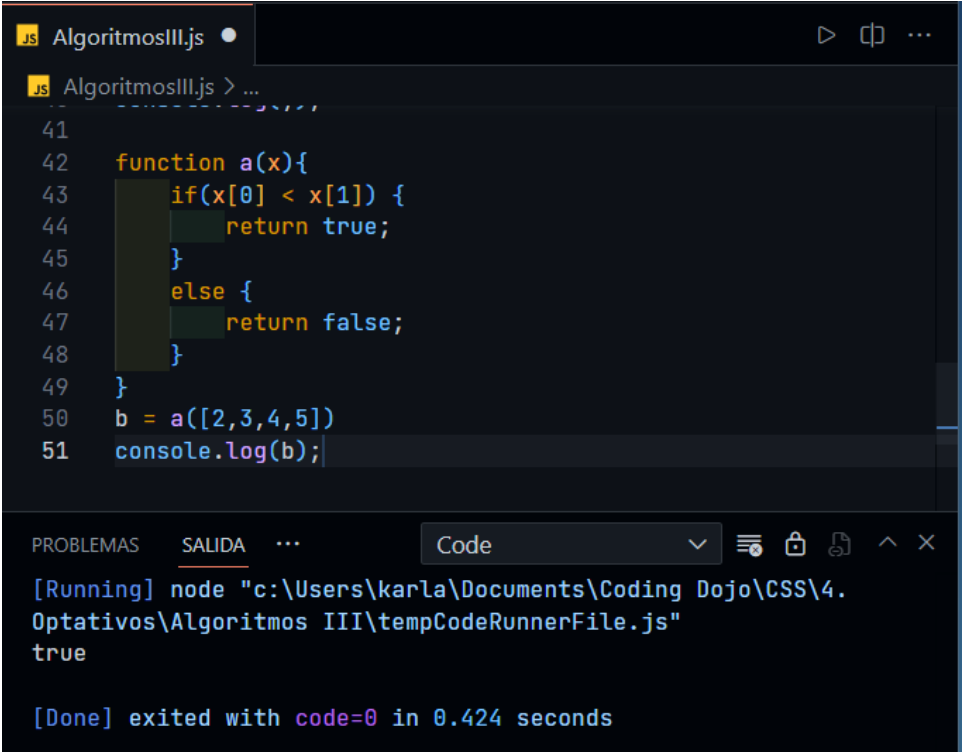
var	value

```
if(x[0] < x[1]) {
    return true;
}
else {
    return false;
}
}
b = a([2,3,4,5])
console.log(b);
```

a	[2, 3, 4, 5]
---	--------------

Output esperado: true

Output de la consola



```
function a(x){
    for(var i=0; i<x.length; i++){
        if(x[i] > 0){
            x[i] = "Coding";
        }
    }
    return x;
}
console.log(a([1,2,3,4]))
```

var	value
a	[1, 2, 3, 4]⇒ [Coding, 2, 3, 4]⇒[Codign, Coding, 3, 4] ⇒ [Codign, Coding, Codign, Coding]

Se remplazan los valores del array por coding y el array no es devuelto hasta que termina elForLoop

Output esperado: [Codign, Coding, Coding, Codign]

Output de la consola

JSAlgoritmosIII.js

JSAlgoritmosIII.js > ...

```
53 function a(x) {
54   for(var i, i=0; i< x.length; i++){
55     if (x[i] > 0) {
56       x[i] = "Coding";
57     }
58   }
59   return x;
60 }
61 console.log(a([1,2,3,4]))
```

PROBLEMAS

SALIDA

...

Code

⌵

🔍

🔒

📄

⬆

✕

[Running] node "c:\Users\karla\Documents\Coding Dojo\CSS\4. Optativos\Algoritmos III\tempCodeRunnerFile.js"

['Coding', 'Coding', 'Coding', 'Coding']

[Done] exited with code=0 in 0.351 seconds

```
function a(x){
  for(var i=0; i<x.length; i++){
    if(x[i] > 5){
      x[i] = "Coding";
    }
    else if(x[i] < 0){
      x[i] = "Dojo";
    }
  }
  return x;
}
console.log(a([5,7,-1,4]))
```

var	value
a	[4, 7, -1, 4]⇒ [5, Codign, -1, 4]⇒ [5, Coding, Dojo, 4]

Output esperado: [5, Coding, Dojo, 4]

Output de la consola

```
JS AlgoritmosIII.js
JS AlgoritmosIII.js > ...
63 |
64 | function a(x){
65 |     for(var i=0; i<x.length; i++){
66 |         if(x[i] > 5){
67 |             x[i]="Codign";
68 |         } else if (x[i] < 0){
69 |             x[i] ="Dojo";
70 |         }
71 |     }
72 |     return x;
73 | }
74 | console.log(a([5, 7, -1, 4]))
```

PROBLEMAS SALIDA ... Code

[Running] node "c:\Users\karla\Documents\Coding Dojo\CSS\4. Optativos\Algoritmos III\tempCodeRunnerFile.js"

[5, 'Codign', 'Dojo', 4]

[Done] exited with code=0 in 0.394 seconds

```
function a(x){
    if(x[0] > x[1]) {
        return x[1];
    }
    return 10;
}
b = a([5,10])
console.log(b);
```

var	value
a	[5, 10]

La condición nunca se cumple, por lo cual solo se retorna 10.

Output esperado: 10

Output de la consola

JS

AlgoritmosIII.js

▶

📄

...

JS

AlgoritmosIII.js

>

📦

a

75

76

77

78

79

80

81

82

83

function a(x){

if(x[0]> x[1]){

return x[1];

}

return 10;

}

b = a([5, 10])

console.log(b)

PROBLEMAS

SALIDA

...

Code

⌵

🔧

🔒

📄

⬆

✕

[Running]

node "c:\Users\karla\Documents\Coding Dojo\CSS\4.

Optativos\Algoritmos III\tempCodeRunnerFile.js"

10

[Done]

exited with code=0

in 0.627 seconds

```
function sum(x){
  sum = 0;
  for(var i=0; i<x.length; i++){
    sum = sum + x[i];
    console.log(sum);
  }
  return sum;
}
```

var	value
sum	0

No retorna nada porque sum no vale nada

Output esperado: nada

Output de la consola

JS

AlgoritmosIII.js

▶

📄

...

JS

AlgoritmosIII.js

>

📦

sum

85

86

87

88

89

90

91

92

93

function sum(x){

sum = 0;

for(var i=0; i<x.length; i++){

sum = sum + x[i];

console.log(sum);

}

return sum;

}

PROBLEMAS

SALIDA

...

Code

⌵

🔧

🔒

📄

⬆

✕

[Running]

node "c:\Users\karla\Documents\Coding Dojo\CSS\4.

Optativos\Algoritmos III\tempCodeRunnerFile.js"

[Done]

exited with code=0

in 0.395 seconds

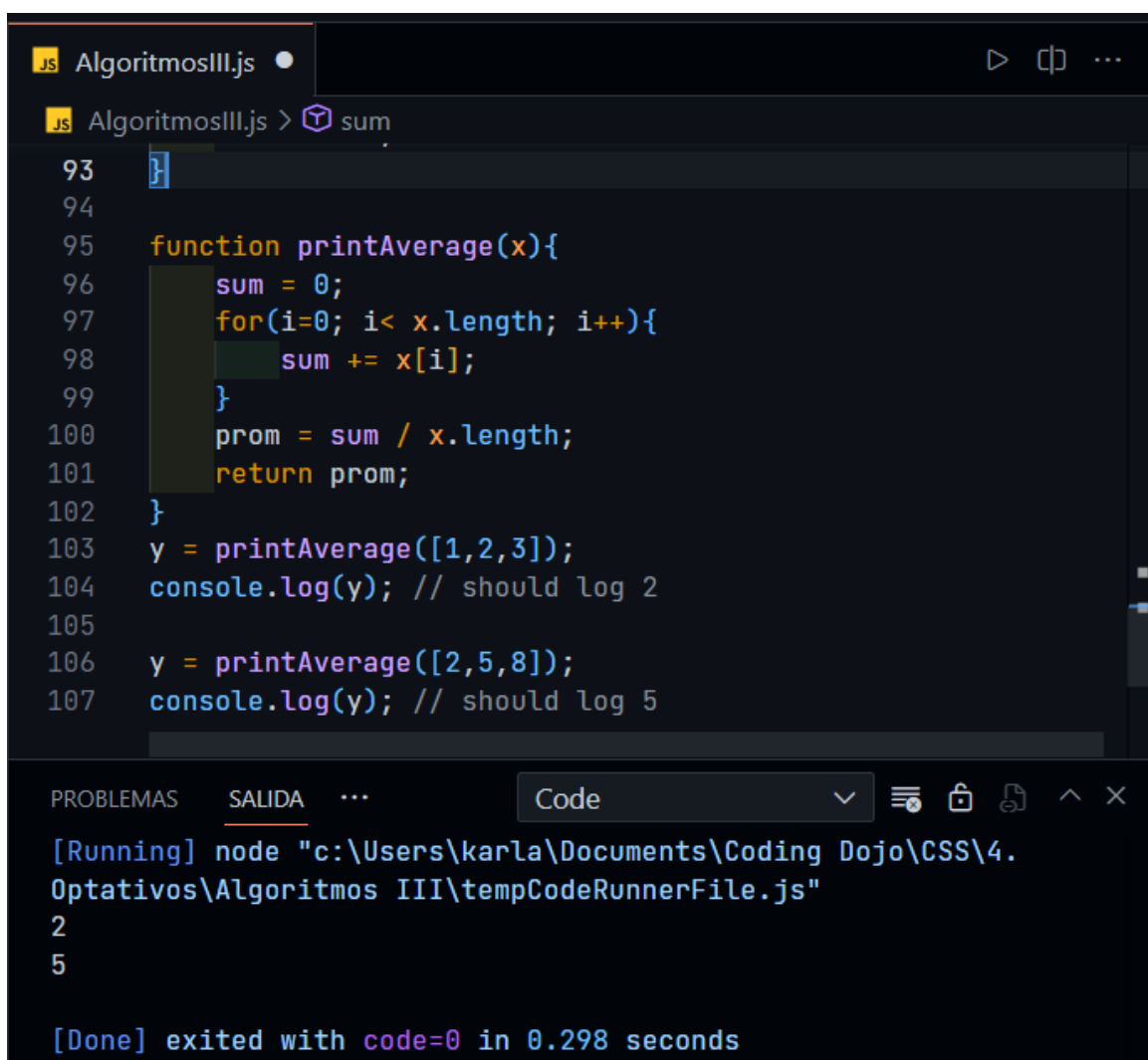
Parte 2

1) Analiza los valores de un array y obtén el promedio (average) de esos valores.

```
function printAverage(x){
  sum = 0;
  for(i=0; i< x.length; i++){
    sum += x[i];
  }
  prom = sum / x.length;
  return prom;
}
y = printAverage([1,2,3]);
console.log(y); // should log 2

y = printAverage([2,5,8]);
console.log(y); // should log 5
```

Output de la consola:



```
93
94
95 function printAverage(x){
96     sum = 0;
97     for(i=0; i< x.length; i++){
98         sum += x[i];
99     }
100     prom = sum / x.length;
101     return prom;
102 }
103 y = printAverage([1,2,3]);
104 console.log(y); // should log 2
105
106 y = printAverage([2,5,8]);
107 console.log(y); // should log 5
```

PROBLEMAS SALIDA ... Code

[Running] node "c:\Users\karla\Documents\Coding Dojo\CSS\4. Optativos\Algoritmos III\tempCodeRunnerFile.js"

2

5

[Done] exited with code=0 in 0.298 seconds

2) Crea un array con todos los enteros impares (odd integers) entre 1 y 255 (inclusive)

```
function returnOddArray(){
  arr = [];
  for(i=0; i<= 255; i++){
    if(i % 2 == 1){
      arr.push(i);
    }
  }
}
```

```

    }
    return arr;
}
y = returnOddArray();
console.log(y); // should log [1,3,5,...,253,255]

```

Output de la consola:

The screenshot shows a code editor with a file named 'AlgoritmosIII.js'. The code defines a function 'returnOddArray()' that creates an empty array 'arr', iterates from 0 to 255, and pushes odd numbers into the array. Below the function, the function is called and the result is logged to the console. The console output shows a long array of odd numbers from 1 to 255, with a truncated view showing the first 28 items.

```

108
109 function returnOddArray(){
110     arr = [];
111     for(i=0; i<= 255; i++){
112         if(i % 2 == 1){
113             arr.push(i);
114         }
115     }
116     return arr;
117 }
118 y = returnOddArray();
119 console.log(y); // should log [1,3,5,...,253,255]

```

PROBLEMAS SALIDA ... Code

[Running] node "c:\Users\karla\Documents\Coding Dojo\CSS\4. Optativos\Algoritmos III\tempCodeRunnerFile.js"

```

[
  1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23,
  25, 27, 29, 31, 33, 35, 37, 39, 41, 43, 45, 47,
  49, 51, 53, 55, 57, 59, 61, 63, 65, 67, 69, 71,
  73, 75, 77, 79, 81, 83, 85, 87, 89, 91, 93, 95,
  97, 99, 101, 103, 105, 107, 109, 111, 113, 115, 117, 119,
  121, 123, 125, 127, 129, 131, 133, 135, 137, 139, 141, 143,
  145, 147, 149, 151, 153, 155, 157, 159, 161, 163, 165, 167,
  169, 171, 173, 175, 177, 179, 181, 183, 185, 187, 189, 191,
  193, 195, 197, 199,
  ... 28 more items
]

```

[Done] exited with code=0 in 0.307 seconds

3) Cuadra cada valor con un array dado, obteniendo el mismo array con valores cambiados..

```

function squareValue(x){
    // your code here
    var arr = [];
    for(var i = 0; i< x.length; i++){
        arr.push(x[i] * x[i]);
    }
    return arr;
}
y = squareValue([1,2,3]);
console.log(y); // should log [1,4,9]

y = squareValue([2,5,8]);
console.log(y); // should log [4,25,64]

```


output de la consola:



The screenshot shows a code editor with a file named 'AlgoritmosIII.js'. The code defines a function 'squareValue(x)' that takes an array and returns a new array with the squares of its elements. The function is called twice: first with [1,2,3] and then with [2,5,8]. The console output shows the results of these calls: [1, 4, 9] and [4, 25, 64].

```
121
122 function squareValue(x){
123     // your code here
124     var arr = [];
125     for(var i = 0; i< x.length; i++){
126         arr.push(x[i] * x[i]);
127     }
128     return arr;
129 }
130 y = squareValue([1,2,3]);
131 console.log(y); // should log [1,4,9]
132
133 y = squareValue([2,5,8]);
134 console.log(y); // should log [4,25,64]
```

PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL Code

[Running] node "c:\Users\karla\Documents\Coding Dojo\CSS\4.0ptativos\Algoritmos III\tempCodeRunnerFile.js"

[1, 4, 9]

[4, 25, 64]

[Done] exited with code=0 in 0.463 seconds