1. What is the value of **foundElement** after the code runs?

const randomNums = [1, 123, 25, 90, 3543, 42];  
  
const foundElement = randomNums.findIndex(num => num > 200);

**3543**

**undefined**

**true**

**4**

In the following Javascript code snippet, the callback function **cb** is a function that sums the value of two numbers.

Fill in the blank so that when the function is passed into the higher order function **hof**, it is then invoked so that the value returned is 8.

let cb = (n1, n2) => {return n1 + n2};  
  
let hof = (func) => {  
  let value =  ;  
  return `this function returned ${value}`;  
}  
  
hof(cb) // 'this function returned 8'

cb(3, 5)

func(3, 5)

func(2, 4)

cb(2, 4)

1. The included code snippet includes a function assigned to the variable **sum**. Given the fact that Javascript functions are first-class objects, fill in the blanks so that a property of **sum** called **name** is assigned the String value, ‘summation’.

const sum = (x, y) => x + y;  
sum name ‘summation’;

===

..

,

==

=

.

1. Fill in the blanks to make **addItUp** log **100** to the console.

const nums = [1, 2, 3, 4];  
const initializer = 90;  
  
const addItUp = nums ((accumulator, currentValue) => {  
  return +  ;  
}, );  
  
console.log(addItUp);

initializer

currentValue

.reduce

accumulator

.map

1. Examine the below code. Which of the functions represents a higher-order function?

let multiplyTwoValues = (val1, val2) => {   
  return val1 \* val2;   
}  
  
let square = (val) => {   
  return val \* val;   
}  
  
let getFunctionString = (func) => {   
  return func.toString();   
}  
  
let result = multiplyTwoValues(2, 2);  
square(result);

**multiplyTwoValues()**

**square()**

**getFunctionString()**

**result**

1. Use the **.map()** method on the **oddNumbers** array to return each number multiplied by 2 into the **evenNumbers** array.

const oddNumbers = [1, 3, 5, 7];  
  
const evenNumbers =  (number => {  
  return \* 2    
});

1. Select the option that correctly assigns the **increment** function to the variable **incrementHolder**.

const increment = (salary) => {  
  const revisedSalary = salary + 5;  
  console.log(revisedSalary);  
  return revisedSalary;  
}

const incrementHolder = increment(5500);

const incrementHolder = increment;

const incrementHolder = incrementSalary;

increment(5500);

1. Which of the following iterator methods returns **undefined**?

**.map()**

**.filter()**

**.some()**

**.forEach()**

1. Which of the following methods returns an array with values that evaluate to truthy based on the condition in the method’s block?

**.forEach()**

**.map()**

**.some()**

**.filter()**