1. Write a simple JavaScript program to join all elements of the following array into a string.
Sample array : myColor = ["Red", "Green", "White", "Black"];
Expected Output :
"Red,Green,White,Black"
"Red,Green,White,Black"
"Red+Green+White+Black"

Answer:

```
myColor = ["Red", "Green", "White", "Black"];
c = myColor.toString();
console.log("" + c + "");
console.log("" + c + "");
console.log("" + c.replaceAll(',' , '+') + "");
```

---

2. Write a JavaScript function that returns a passed string with letters in alphabetical order.
Example string : 'webmaster'
Expected Output : 'abeemrstw'
Assume punctuation and numbers symbols are not included in the passed string.

Answer:

```
function sortWord(word) {
   a = Array.from(word);
   c = a.sort();
   d = c.toString();
   return d.replaceAll(',' , '');
}
```

---

4. Write a JavaScript function which will take an array of numbers stored and find the second lowest and second greatest numbers, respectively.
Sample array : [1,2,3,4,5]
Expected Output : 2,4

Answer:

```
function secondLowGreat(...number){
   a = number.sort(function(a, b){return a-b});
   return (a[1] + ',' + a[a.length-2])
}
```

7. Write a JavaScript function that accepts two arguments, a string and a letter and the function will count the number of occurrences of the specified letter within the string.
Sample arguments : 'muchas gracias', 'a'
Expected output : 3

Answer:

```
function findIndex(word,letter) {
   a = Array.from(word);
   i = 0;
   c = 0;
   while (i != a.length) {
      if (letter === a[i]){
      c++;
      }
      i++;
   }
return c
}
```

---

11. Write a JavaScript program to compute the least common multiple (GCD) of two positive integers.

Answer:

```
function LCM(number1, number2) {
   i = 1;
   o = 1;
   if (number1 > number2) {
      c = number1;
      while (i != c) {
      x = number1 * i;
         while (o != c +1) {
         y = number2 * o;
         if (x == y) break;
         o++;
         }
      if (x == y) break;
      i++;
      o = 1;
      }
   }
   else {
      c = number2;
      while (i != c) {
      x = number2 * i;
         while (o != c +1) {
```

```
        y = number1 * o;
        if (x == y) break;
        o++;
        }
      if (x == y) break;
      i++;
      o = 1;
      }
   }
return x
}
```

---

13. Write a JavaScript function to convert a binary number to a decimal number.
Test Data :
console.log(bin_to_dec('110011'));
console.log(bin_to_dec('100'));
51
4

Answer:

```
function binToDec(number) {
   j = '' + number;
   k = Array.from(j);
   sum = 0;
   i = k.length - 1;
   o = 0;
   while (i != -1) {
      x = Math.pow(2, o) * k[i];
      sum = sum + x;
      i--;
      o++;
   }
return sum
}
```

---

16. Write a JavaScript function to get the nth largest element from an unsorted array.
console.log(nthlargest([43, 56, 23, 89, 88, 90, 99, 652], 4));
89

Answer:

```
function nthLargest(digit, ...number){
   a = number.sort(function(a, b){return a-b});
   c = a.reverse()
   return c[digit-1]
}
```

17. Write a JavaScript function to generate an array of specified length, filled with integer numbers, increase by one from starting position.

Test Data :
console.log(array_range(1, 4));
[1, 2, 3, 4]
console.log(array_range(-6, 4));
[-6, -5, -4, -3]

Answer:

```
function increaseNumber(start, limit) {
   arr = [];
   i = 0;
   while (i != limit) {
      arr.push(start)
      start++;
      i++;
   }
return arr
}
```

---

19. Write a JavaScript function to check if an array contains a specific element.

Test data :
arr = [2, 5, 9, 6];
console.log(contains(arr, 5));
[True]

Answer:

```
function checkSameValue(element) {
   var a = [2, 5, 9, 6];
   i = 0;
   while (a[i] != element) {
      i++
   }
   return a[i] == element
}
```

21. Write a JavaScript function to cast a square root of a number to an integer.
Test Data :
console.log(sqrt_to_int(17));
4

Answer:

```
function squareRoot(number) {
   x = Math.sqrt(number);
   return c = Math.floor(x)
}
```

---

22. Write a JavaScript function to convert an angle from degrees to radians
Test Data :
console.log(degrees_to_radians(45));
0.7853981633974483

Answer:

```
function degToRad(number) {
   return radian = number * (Math.PI/180)
}
```

---

23. Write a JavaScript function which will return true if the values are powers of two and false if they are not.
Test Data :
console.log(isPower_of_two(64));
true
console.log(isPower_of_two(94));
false

Answer:

```
function ifPow(number) {
   i = number;
   while (i != 0) {
      a = Math.pow(2,i);
      if (a == number) break;
      i--;
   }
return (a == number)
}
```

26. Write a JavaScript function to concatenate a given string n times (default is 1).
Test Data :
console.log(repeat('Ha!'));
console.log(repeat('Ha!',2));
console.log(repeat('Ha!',3));
"Ha!"
"Ha!Ha!"
"Ha!Ha!Ha!"

Answer:

```
function repeatWord(string, number) {
   c = '';
   i = 0;
   while (i != number) {
      c += string;
      i++
   }
return c
}
```

---

28. There are two arrays with individual values, write a JavaScript program to compute the sum of each individual index value from the given arrays.
Sample array :
array1 = [1,0,2,3,4];
array2 = [3,5,6,7,8,13];
Expected Output :
[4, 5, 8, 10, 12, 13]

Answer:

```
sumArray = [];
i = 0
if (array1.length < array2.length) {
   while (i != array1.length) {
      c = array1[i] + array2[i];
      sumArray.push(c);
      i++;
   }
   while (i != array2.length) {
      c = array2[i];
      sumArray.push(c);
      i++;
   }
```

```
}
else if (array2.length < array1.length) {
   while (i != array2.length) {
      a = array1.length;
      c = array1[i] + array2[i];
      sumArray.push(c);
      i++;
   }
   while (i != array1.length) {
      c = array1[i];
      sumArray.push(c);
      i++;
   }
}
console.log(sumArray);
```

---

30. Write a JavaScript function to check whether an `input` is an array or not.
Test Data :
console.log(is_array('Homelander'));
console.log(is_array([1, 2, 4, 0]));
false
true

Answer:

```
function checkIfArray(input) {
   c = Array.isArray(input)
   return c
}
```

---

32. Write a JavaScript function to compute the factors of a positive integer.

Answer:

```
function findFactors(number) {
   i = 0
   factors = []
   while (i != number + 1) {
      a = number % i;
      if (a == 0) {
         factors.push(i)
      }
      i++;
   }
return ('The factors of ' + number + ' are ' + factors)
}
```