1. Do the below programs in anonymous function & IIFE
   1. Print odd numbers in an array

inp.on("close", () => {

var N=userInput[0];

for(var i=1; i<=N; i += 2)

{

console.log(i);

}

});

Input

10

**Output:**

1

3

5

7

9

* 1. Convert all the strings to title caps in a string array

inp.on("close", () => {

function titleCase(str) {

str = str.toLowerCase().split(' ');

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

str[i] = str[i].charAt(0).toUpperCase() + str[i].slice(1);

}

return str.join(' ');

}

console.log(titleCase("I'm a little tea pot"));

});

##### Output:

I'm A Little Tea Pot

* 1. Sum of all numbers in an array

inp.on("close", () => {

var arr = [1,2,3,4]

var sum = 0;

for (let i = 0; i < arr.length; i++) {

sum += arr[i];

}

console.log(sum);

});

##### Output:

10

* 1. Return all the prime numbers in an array

inp.on("close", () => {

var numArray = [2, 3, 4, 5, 6, 7, 8, 9, 10]

numArray = numArray.filter((number) => {

for (var i = 2; i <= Math.sqrt(number); i++) {

if (number % i === 0) return false;

}

return true;

});

console.log(numArray);

});

##### Output:

[ 2, 3, 5, 7 ]

* 1. Return all the palindromes in an array

inp.on("close", () => {

const arr = ['carecar', 1344, 12321, 'did', 'cannot'];

const isPalindrome = el => {

const str = String(el);

let i = 0;

let j = str.length - 1;

while(i < j) {

if(str[i] === str[j]) {

i++;

j--;

}

else {

return false;

}

}

return true;

};

const findPalindrome = arr => {

return arr.filter(el => isPalindrome(el));

};

console.log(findPalindrome(arr));

});

##### Output:

[ 12321, 'did' ]

* 1. Return median of two sorted arrays of same size

inp.on("close", () => {

function getMedian(ar1, ar2, n)

{

var i = 0; /\* Current index of i/p array ar1[] \*/

var j = 0; /\* Current index of i/p array ar2[] \*/

var count;

var m1 = -1, m2 = -1;

/\* Since there are 2n elements, median will be average

of elements at index n-1 and n in the array obtained after

merging ar1 and ar2 \*/

for (count = 0; count <= n; count++)

{

/\*Below is to handle case where all elements of ar1[] are

smaller than smallest(or first) element of ar2[]\*/

if (i == n)

{

m1 = m2;

m2 = ar2[0];

break;

}

/\*Below is to handle case where all elements of ar2[] are

smaller than smallest(or first) element of ar1[]\*/

else if (j == n)

{

m1 = m2;

m2 = ar1[0];

break;

}

/\* equals sign because if two

arrays have some common elements \*/

if (ar1[i] <= ar2[j])

{

m1 = m2; /\* Store the prev median \*/

m2 = ar1[i];

i++;

}

else

{

m1 = m2; /\* Store the prev median \*/

m2 = ar2[j];

j++;

}

}

return (m1 + m2)/2;

}

/\* Driver program to test above function \*/

var ar1 = [1, 12, 15, 26, 38];

var ar2 = [2, 13, 17, 30, 45];

var n1 = ar1.length;

var n2 = ar2.length;

if (n1 == n2)

console.log("Median is "+ getMedian(ar1, ar2, n1));

else

console.log("Doesn't work for arrays of unequal size");

});

Input

[1 2 3 4]

[2 3 4 5]

##### Output:

Median is 16

* 1. Remove duplicates from an array

inp.on("close", () => {

var [fname,,,lname] = ["John","doe","working","in guvi"];

console.log(fname);

console.log(lname);

});

##### Output:

John

in guvi

* 1. Rotate an array by k times

inp.on("close", () => {

const arr = [1, 3, 6, 5, 4];

const rotatedArr = arr.map((e, i) => arr[arr.length-i-1])

console.log(rotatedArr)

});

##### Output:

[ 4, 5, 6, 3, 1 ]

1. Do the below programs in arrow functions
2. Print odd numbers in an array

inp.on("close", () => {

let arr = [1,2,3,4,5,6,7,8,9,10,11,12]

let odds = arr.filter(n => n%2)

console.log(odds)

});

##### Output:

[ 1, 3, 5, 7, 9, 11 ]

1. Sum of all numbers in an array

inp.on("close", () => {

sum = (a, b) => {

return(a+b)

}

x= sum(34,87)

console.log(x)

});

##### Output:

121

1. Return all the prime numbers in an array

inp.on("close", () => {

const newArray = [1, 3, 2, 5, 10];

const isPrime = num => {

for (let i = 2; i < num; i++) {

if (num % i === 0) return false;

}

return num !== 1;

};

const myPrimeArray = newArray.filter(element => isPrime(element));

console.log(myPrimeArray);

});

##### Output:

[ 3, 2, 5 ]