## Do the below programs in anonymous function & IIFE

## Print odd numbers in an array

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=[1,2,3,4,5,6]

var v= a.filter(function (val){

return (val%2)!= 0

})

console.log(v)

});

## Convert all the strings to title caps in a string array

## const readline = require('readline');

## const inp = readline.createInterface({

## input: process.stdin

## });

## const userInput = [];

## inp.on("line", (data) => {

## userInput.push(data);

## });

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

## var arr=["vedha","narashimma","malai","sumathi"]

## var z= arr.map(function (str) {

## 

## return str.charAt(0).toUpperCase() + str.substr(1).toLowerCase();

## 

## 

## })

## console.log(z)

## });

## Sum of all numbers in an array

const readline = require('readline');

const inp = readline.createInterface({

input: process.stdin

});

const userInput = [];

inp.on("line", (data) => {

userInput.push(data);

});

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

const array = [1, 2, 3, 4];

let sum = 0;

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

sum += array[i];

}

console.log(sum);

});

## Return all the prime numbers in an array

const readline = require('readline');

const inp = readline.createInterface({

input: process.stdin

});

const userInput = [];

inp.on("line", (data) => {

userInput.push(data);

});

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

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

const myPrimeArray = newArray.filter(function(num) {

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

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

}

return num !== 1;

});

console.log(myPrimeArray);

});

## Return all the palindromes in an array

## const readline = require('readline');

## const inp = readline.createInterface({

## input: process.stdin

## });

## const userInput = [];

## inp.on("line", (data) => {

## userInput.push(data);

## });

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

## var validatePalin=function (string) {

## 

## // get the total length of the words

## const len = string.length;

## 

## // Use for loop to divide the words into 2 half

## for (let i = 0; i < len / 2; i++) {

## 

## // validate the first and last characters are same

## if (string[i] !== string[len - 1 - i]) {

## console.log( 'It is not a palindrome');

## }

## }

## console.log( 'It is a palindrome');

## }

## validatePalin("malayalam")

## });

## Return median of two sorted arrays of same size

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=[1,5,3,6]

var b=[4,9,8,5]

function getMedian(ar1, ar2, n)

{

var i = 0;

var j = 0;

var count;

var m1 = -1, m2 = -1;

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

{

if (i == n)

{

m1 = m2;

m2 = ar2[0];

break;

}

else if (j == n)

{

m1 = m2;

m2 = ar1[0];

break;

}

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

{

m1 = m2;

m2 = ar1[i];

i++;

}

else

{

m1 = m2;

m2 = ar2[j];

j++;

}

}

return (m1 + m2)/2;

}

console.log(getMedian(a, b, 4))

});

## Remove duplicates from an array

const readline = require('readline');

const inp = readline.createInterface({

input: process.stdin

});

const userInput = [];

inp.on("line", (data) => {

userInput.push(data);

});

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

let chars = ['A', 'B', 'A', 'C', 'B'];

let uniqueChars = chars.filter(function(c, index) {

return chars.indexOf(c) === index;

});

console.log(uniqueChars);

});

**Problem**:

Write a function called “addFive”.  
Given a number, “addFive” returns 5 added to that number.  
  
Input:

addFive(5);  
addFive(0);  
addFive(-5);

Output:

10  
5  
0

## ANSWER: const readline = require('readline');

## const inp = readline.createInterface({

## input: process.stdin

## });

## const userInput = [];

## inp.on("line", (data) => {

## userInput.push(data);

## });

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

## 

## function addFive(num) {

## return num+5

## }

## console.log(addFive(5))

## console.log(addFive(0))

## console.log(addFive(-5))

## });

**Problem**:

Write a function called “getOpposite”.  
Given a number, return its opposite

Input:

getOpposite(5);  
getOpposite(0);  
getOpposite(-5);  
getOpposite(“5a”);  
getOpposite(5.5);

Output:

-5  
0  
5  
-1  
-1

const readline = require('readline');

const inp = readline.createInterface({

input: process.stdin

});

const userInput = [];

inp.on("line", (data) => {

userInput.push(data);

});

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

var num = 5;

function getOpposite(num) {

return num-(2\*num)

}

console.log(getOpposite(6))

});

**Problem**:

Fill in your code that takes an number minutes and converts it to seconds.

Examples  
toSeconds(5) ➞ 300

toSeconds(3) ➞ 180

toSeconds(2) ➞ 120

ANSWER: const readline = require('readline');

const inp = readline.createInterface({

input: process.stdin

});

const userInput = [];

inp.on("line", (data) => {

userInput.push(data);

});

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

var num = 5;

function tosecond(num) {

return num\*60

}

console.log(tosecond(6))

});

**Problem**  
Create a function that takes a string and returns it as an integer.

Examples  
toInteger(“6”) ➞ 6

toInteger(“1000”) ➞ 1000

toInteger(“12”) ➞ 12

ANSWER: const readline = require('readline');

const inp = readline.createInterface({

input: process.stdin

});

const userInput = [];

inp.on("line", (data) => {

userInput.push(data);

});

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

var num = 5;

function toINTEGER(num) {

return parseInt(num)

}

console.log(typeof(toINTEGER("6")))

console.log(typeof("6"))

});

**Problem**

Create a function that takes a number as an argument, increments the number by +1 and returns the result.

Examples  
nextNumber(0) ➞ 1

nextNumber(9) ➞ 10

nextNumber(-3) ➞ -2

ANSWER: const readline = require('readline');

const inp = readline.createInterface({

input: process.stdin

});

const userInput = [];

inp.on("line", (data) => {

userInput.push(data);

});

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

var num = 5;

function nxtnum(num) {

return (num+1)

}

console.log(nxtnum(5))

});

**Problem**

Create a function that takes an array and returns the first element.

Examples  
getFirstElement([1, 2, 3]) ➞ 1

getFirstElement([80, 5, 100]) ➞ 80

getFirstElement([-500, 0, 50]) ➞ -500

ANSWER: const readline = require('readline');

const inp = readline.createInterface({

input: process.stdin

});

const userInput = [];

inp.on("line", (data) => {

userInput.push(data);

});

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

var n = [1,2,3,4];

function FIRSTele(num) {

return num[0]

}

console.log(FIRSTele(n) )

});

**Problem**

Convert Hours into Seconds

Write a function that converts hours into seconds.

Examples  
hourToSeconds(2) ➞ 7200

hourToSeconds(10) ➞ 36000

hourToSeconds(24) ➞ 86400

ANSWER: const readline = require('readline');

const inp = readline.createInterface({

input: process.stdin

});

const userInput = [];

inp.on("line", (data) => {

userInput.push(data);

});

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

function TOsecond(num) {

return num\*60

}

console.log(TOsecond(1) )

});

**Problem**

Find the Perimeter of a Rectangle  
Create a function that takes height and width and finds the perimeter of a rectangle.

Examples  
findPerimeter(6, 7) ➞ 26

findPerimeter(20, 10) ➞ 60

findPerimeter(2, 9) ➞ 22

const readline = require('readline');

const inp = readline.createInterface({

input: process.stdin

});

const userInput = [];

inp.on("line", (data) => {

userInput.push(data);

});

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

function Perimeter(a,b) {

return 2\*(a+b)

}

console.log(Perimeter(5,5) )

});

**Problem**

Less Than 100?  
Given two numbers, return true if the sum of both numbers is less than 100. Otherwise return false.

const readline = require('readline');

const inp = readline.createInterface({

input: process.stdin

});

const userInput = [];

inp.on("line", (data) => {

userInput.push(data);

});

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

function lesshund(a,b) {

if((a+b)<100){

return true}

return false

}

console.log(lesshund(6,5) )

});

**Problem**

There is a single operator in JavaScript, capable of providing the remainder of a division operation. Two numbers are passed as parameters. The first parameter divided by the second parameter will have a remainder, possibly zero. Return that value.

const readline = require('readline');

const inp = readline.createInterface({

input: process.stdin

});

const userInput = [];

inp.on("line", (data) => {

userInput.push(data);

});

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

function Reminder(a,b) {

return a%b

}

console.log(Reminder(5,5) )

});