|  |
| --- |
|  |
|  | **print odd numbers in an array using anonymous function**  **1)Print odd numbers in an array**  anonymous : function(array){ |
|  | for(var i = 0 ; i< array.length ; i++){ |
|  | if(array[i]%2!=0){ |
|  | console.log(array[i]) |
|  | } |
|  | } |
|  | } |
|  | IIFE : (function(array){ |
|  | for(var i = 0 ; i< array.length ; i++){ |
|  | if(array[i]%2!=0){ |
|  | console.log(array[i]) |
|  | } |
|  | } |
|  | })([1,2,3,4]) |
|  |  |
|  | Arrow Function oddNumbers = (array) => { |
|  | for(var i = 0 ; i< array.length ; i++){ |
|  | if(array[i]%2!=0){ |
|  | console.log(array[i]) |
|  | } |
|  | } |
|  | } |
|  |  |
|  | 2)Convert all the strings to title caps in a string array |
|  |  |
|  | anonymous : function (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(' '); |
|  | } |
|  | IIFE : (function (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(' '); |
|  | })("MUDRA IS MY NAME"); |
|  | Arrow 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(' '); |
|  | } |
|  | 3)Sum of all numbers in an array |
|  | anonymous : function(array){ |
|  | var sum = 0; |
|  | for(var i = 0 ; i< array.length ; i++){ |
|  | sum = sum + array[i]; |
|  | } |
|  | return sum; |
|  | } |
|  | IIFE : (function(array){ |
|  | var sum = 0; |
|  | for(var i = 0 ; i< array.length ; i++){ |
|  | sum = sum + array[i]; |
|  | } |
|  | return sum; |
|  | })([1,2,3,4]) |
|  | Arrow: sum = (array)=>{ |
|  | var sum = 0; |
|  | for(var i = 0 ; i< array.length ; i++){ |
|  | sum = sum + array[i]; |
|  | } |
|  | return sum; |
|  | } |
|  | 4)Return all the prime numbers in an array |
|  | Anonymous Function: |
|  | function(numArray){ |
|  | numArray = numArray.filter((number) => { |
|  | for (var i = 2; i <= Math.sqrt(number); i++) { |
|  | if (number % i === 0) return false; |
|  | } |
|  | return true; |
|  | }); |
|  | console.log(numArray); |
|  | } |
|  | IIFE |
|  | ( |
|  | function(numArray){ |
|  | numArray = numArray.filter((number) => { |
|  | for (var i = 2; i <= Math.sqrt(number); i++) { |
|  | if (number % i === 0) return false; |
|  | } |
|  | return true; |
|  | }); |
|  | console.log(numArray); |
|  | })([1,2,3,4]) |
|  | Arrow Function : |
|  |  |
|  | primeNumber = (numArray) => { |
|  | numArray = numArray.filter((number) => { |
|  | for (var i = 2; i <= Math.sqrt(number); i++) { |
|  | if (number % i === 0) return false; |
|  | } |
|  | return true; |
|  | }); |
|  | console.log(numArray); |
|  | } |
|  |  |
|  | 5) Return all the palindromes in an array |
|  |  |
|  | function isPalindrome(N) |
|  | { |
|  | let str = "" + N; |
|  | let len = str.length; |
|  | for (let i = 0; i < parseInt(len / 2, 10); i++) |
|  | { |
|  | if (str[i] != str[len - 1 - i ]) |
|  | return false; |
|  | } |
|  | return true; |
|  | } |
|  |  |
|  | Anonymous Function : function (arr, n) |
|  | { |
|  | // Traversing each element of the array |
|  | // and check if it is palindrome or not |
|  | for (let i = 0; i < n; i++) |
|  | { |
|  | let ans = isPalindrome(arr[i]); |
|  | if (ans == false) |
|  | return false; |
|  | } |
|  | return true; |
|  | } |
|  |  |
|  | IIFE : |
|  |  |
|  | ( function (arr, n) |
|  | { |
|  | // Traversing each element of the array |
|  | // and check if it is palindrome or not |
|  | for (let i = 0; i < n; i++) |
|  | { |
|  | let ans = isPalindrome(arr[i]); |
|  | if (ans == false) |
|  | return false; |
|  | } |
|  | return true; |
|  | })([1,2,3] , 3) |
|  |  |
|  | Arrow : |
|  | Palindrome = (arr, n) => |
|  | { |
|  | // Traversing each element of the array |
|  | // and check if it is palindrome or not |
|  | for (let i = 0; i < n; i++) |
|  | { |
|  | let ans = isPalindrome(arr[i]); |
|  | if (ans == false) |
|  | return false; |
|  | } |
|  | return true; |
|  | } |
|  | Q.Return median of two sorted arrays of same size |
|  | Q.Remove duplicates from an Array |
|  | Anonymous Function : function(array){ |
|  | let dup = [...new Set(array)]; |
|  | console.log(dup); |
|  | } |
|  | IIFE : (function(array){ |
|  | let dup = [...new Set(array)]; |
|  | console.log(dup); |
|  | })([1,1,2,3,4]) |
|  |  |
|  |  |
|  | Q.Rotate an array by K times |
|  |  |
|  | function reverse(array , li , ri){ |
|  | while(li < ri){ |
|  | int temp = a[li]; |
|  | a[li]= a[ri]; |
|  | a[ri] = temp; |
|  |  |
|  | li++; |
|  | ri--; |
|  | } |
|  | } |
|  | Anonymous function : function(array , k){ |
|  | k = k % a.length; |
|  | if(k < 0){ |
|  | k += a.length; |
|  | } |
|  |  |
|  | reverse(a, 0, a.length - k - 1); |
|  | reverse(a, a.length - k, a.length - 1); |
|  | reverse(a, 0, a.length - 1); |
|  | } |
|  |  |
|  | IIFE : (function(array , k){ |
|  | k = k % a.length; |
|  | if(k < 0){ |
|  | k += a.length; |
|  | } |
|  |  |
|  | reverse(a, 0, a.length - k - 1); |
|  | reverse(a, a.length - k, a.length - 1); |
|  | reverse(a, 0, a.length - 1); |
|  | })([1,2,3,4] , 2) |

**C++ Program to Find and Print Odd Numbers in an Array**

#include<iostream>

using namespace std;

int main()

{

int arr[10], i;

cout<<"Enter 10 numbers: ";

for(i=0; i<10; i++)

cin>>arr[i];

cout<<"\nList of Odd Numbers are:\n";

for(i=0; i<10; i++)

{

if(arr[i]%2!=0)

cout<<arr[i]<<" ";

}

cout<<endl;

return 0;

}

**Convert all the strings to title caps in a string array in c++**

#include <iostream>

**using** **namespace** std;

// create a class

**class** **String** {

// private data member

**private:**

**char** str[**30**];

// public functions

**public:**

// getString() function to get the string

**void** **getString**() {

cout << "Enter String:";

cin.getline(str, **30**);

}

// titleCase() function to convert

// the string to title case

string **titleCase**() {

// char type array to store title cased

// string

**char** temp[**30**];

// int type variable for indexing

**int** index;

// for loop for traversing of whole string

**for** (index = **0**; str[index]; index++) {

// if condition for the first character

// of the string

**if** (index == **0**) {

// if condition to check if it is a alphabet

**if** ((str[index] >= 'a' && str[index] <= 'z') || (str[index] >= 'A' && str[index] <= 'Z')) {

// if condition to check if it is a lowercase alphabet

**if** ((str[index] >= 'a' && str[index] <= 'z')) {

// then convert it to uppercase and

// store it in temp string

temp[index] = str[index] - **32**;

}

// else it is copied as it is

**else** {

temp[index] = str[index];

}

}

// else it is copied as it is

**else** {

temp[index] = str[index];

}

}

// else if condition for spaces at index

**else** **if** (str[index] == ' ') {

// copying space in temp string

temp[index] = str[index];

// increment of 1 in index to

// check next character

index++;

// if condition to check if it is a alphabet

**if** ((str[index] >= 'a' && str[index] <= 'z') || (str[index] >= 'A' && str[index] <= 'Z')) {

// if condition to check if it is a lowercase alphabet

**if** ((str[index] >= 'a' && str[index] <= 'z')) {

// then convert it to uppercase and

// store it in temp string

temp[index] = str[index] - **32**;

}

// else it is copied as it is

**else** {

temp[index] = str[index];

}

}

// else it is copied as it is

**else** {

temp[index] = str[index];

}

}

// else if condition to check if it is a alphabet

**else** **if** ((str[index] >= 'a' && str[index] <= 'z') || (str[index] >= 'A' && str[index] <= 'Z')) {

// if condition to check if it is a uppercase alphabet

**if** ((str[index] >= 'A' && str[index] <= 'Z')) {

// then convert it to loweercase and

// store it in temp string

temp[index] = str[index] + **32**;

}

// else it is copied as it is

**else** {

temp[index] = str[index];

}

}

// else it is copied as it is

**else** {

temp[index] = str[index];

}

}

// transfering null in the temp string

temp[index] = **0**;

// returning temp string

**return** temp;

}

};

**int** **main**() {

// create an object

String S;

// function is called by the object

// to store the string

S.getString();

// string type variable to store

// title case string

string str;

// titleCase() function is called

// by the object to

// convert the string to title case

str = S.titleCase();

cout << "Title cased String:" << str;

**return** **0**;

}

# How to get the sum of an array

function sumArray(array){

const ourArray = [1, 4, 0, 9, -3];

let sum = 0;

for (let i = 0; i < ourArray.length; i += 1) {

sum += ourArray[i];

}

return sum;

}

console.log(sumArray([1, 4, 0, 9, -3]));

**return all the prime numbers in an array javascript**

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);

**return all the palindromes in an array in javascript**

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));