//////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

Fibonacci series - Fn = Fn-1 + Fn-2

function fibonacci(num: number) {

let num1 = 0;

let num2 = 1;

let sum;

// let collect: number[] = []

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

sum = num1 + num2 //1,2,3,5,8,13,21,34

num1 = num2 //1,1,2,3,5,8,13,21

num2 = sum //1,2,3,5,8,13,21,34

// collect.push(num2)

}

return num2

//return collect

}

console.log(`val for specified number in fibonacci series is `+fibonacci(9))

/////////////////////////////////////////////////////////

Number reversal

function reverseNumbers(val: number) {

let num = val.toString()

let collectNum = ''

for(let i = num.length - 1; i >= 0; i-- ) {

collectNum += num[i]

}

return parseInt(collectNum)

}

let num = 2356

console.log(reverseNumbers(num));

function numberReverse(val: number) {

let temp = val.toString()

let rev: string = '';

let revNumber: number

for(let i = temp.length-1; i>=0; i--) {

rev += temp[i]

}

revNumber = parseInt(rev)

return revNumber \*Math.sign(val)

}

console.log(numberReverse(-54896));

////////////////////////////////////////////////////////////////

Reverse numbers

function reverseNumbers(val: number) {

let reveNumber = 0

while(val != 0){

reveNumber = reveNumber \* 10 + (val % 10); //1->; 0+(12345 % 10)=>t;5 =>; 5

//2-&g>; 50 + (1234 % 10)=&g>;4 =&g>; 54

//3-&g>; 540 + (123 % 10)=&g>; 3 =&g>; 543

//4-&g>; 5430 + (12 % 10)=&g>; 2 =&>t; 5432

//5-&g>; 5432 + (1 % 10)=&g>;1 =&g>; 54321

val = Math.floor(val/10);//1 -&g>; 12345/10 =&g>; 1234.5 =&g>;1234

//2 -&g>; 1234/10 =&g>; 123.4 =&g>; 123

//3 -&g>; 123/10 =&g>; 12.3 =&g>; 12

//4 -&g>; 12/10 =&g>; 1.2 =&g>; 1

//5 -&g>; 1/10 =&g>; 0.1 =&g>; 0

}

return reveNumber

}

console.log(reverseNumbers(12345)); //54321

////////////////////////////////////////////

Palindrome Check

function palindromeCheck(val: string) {

let rightSide = val.length-1//m,a,l,a

for(let i=0; i<val.length/2; i++) { //1,2,3,4

if(val[i] != val[rightSide]) { //m,a,l,a

return false

}

rightSide--;

}

return true

}

console.log(palindromeCheck('vaiav'));

/////////////////////////////////////////////////////

Separate number and string

function seperateNumAndSring(val: any) {

// let numbers = "";

// let string = "";

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

// if(isNaN(val[i])) {

// string += val[i]

// }

// if(!isNaN(val[i])) {

// numbers += val[i]

// }

// }

// console.log(string)

// console.log(+numbers)

let matchString = val.replace(/[^a-z^A-Z]/g, "")

let matchNumber = val.replace(/[^0-9]/g, "")

if(matchString) {

console.log(matchString)

}

if(matchNumber) {

console.log(+matchNumber)

}

}

seperateNumAndSring("1ab23cAf8f6Bf");

Armstrong number

function armStrongNumber(val : number) {

let num: number = val;

let arm = 0;

let rem = 0;

while(num > 0) {

rem = num % 10 //3,5,1

arm = arm + (rem\*rem\*rem) //27,125,1 ==> 153

num = Math.floor(num/10) //15,1,0

}

if(arm === val) {

return true

}

else {

return false

}

}

console.log(armStrongNumber(153))

Remove duplicate object in array

function removeDuplicateObjInArray(array : any) {

const obj: any = {}

let uniqueObj = []

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

//console.log(array[i].name)

if(!obj[array[i].name]) {

obj[array[i].name] = true;

uniqueObj.push(array[i])

}

}

console.log(uniqueObj)

}

const array = [

{name: "surya", job: "developing"},

{name: "karthi", job: "testing"},

{name: "karthi", job: "design"},

]

removeDuplicateObjInArray(array)

function removeDuplicateObjInArray() {

let arrayOne = [{id: 1, value: 'red'}, {id: 2, value: 'green'}, {id: 3, value: 'blue'}]

let arrayTwo = [{id: 4, value: 'white'}, {id: 5, value: 'yellow'}, {id: 3, value: 'blue'}]

const combainedArr = [...arrayOne,...arrayTwo]

const uniqeArr = []

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

let current : any = combainedArr[i]

let uniquee = true

for(let j = 0; j < combainedArr.length; j++) {

if(i === j) continue

if(current.id === combainedArr[j].id) {

uniquee = false;

}

}

if(uniquee) uniqeArr.push(current);

}

return uniqeArr;

}

console.log(removeDuplicateObjInArray())

function removeDuplicateObjInArray() {

let arrayOne = [{id: 1, value: 'red'}, {id: 2, value: 'green'}, {id: 3, value: 'blue'}]

let arrayTwo = [{id: 4, value: 'white'}, {id: 5, value: 'yellow'}, {id: 3, value: 'blue'}]

const reducedArray :any = []

const uniqObj : any = {}

arrayOne.forEach((obj) => {

if(!uniqObj[obj.id]){

uniqObj[obj.id] = true

reducedArray.push(obj);

}

})

arrayTwo.forEach((obj) => {

if(!uniqObj[obj.id]) {

uniqObj[obj.id] = true

reducedArray.push(obj);

}

})

return reducedArray

}

console.log(removeDuplicateObjInArray())

Remove duplicates in array

function removeDuplicatesInArray(array : number[]) {

let uniqueArray : number[] = []

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

if(uniqueArray.indexOf(array[i]) == -1) {

uniqueArray.push(array[i]);

}

}

console.log(uniqueArray)

}

const array : number[] = [1,2,1,3,2,4,5,3]

removeDuplicatesInArray(array) //[1,2,3,4,5]

////////////////////////////////////////////////////////////////

function removeDuplicatesInArray(array : number[]) {

const obj: any = {}

for(let i of array) {

obj[i] = true;

}

console.log(Object.keys(obj))

}

const array : number[] = [1,2,1,3,2,4,5,3]

removeDuplicatesInArray(array) //[1,2,3,4,5]

function mapToGetNameId(array : any) {

const userID = array.map((user : any) => user.name + " " + user.id)

console.log(userID)

}

const array = [

{name: "surya", job: "developing", id: 1},

{name: "karthi", job: "testing", id: 2},

{name: "karthi", job: "design", id: 2},

]

mapToGetNameId(array) // [LOG]: ["surya 1", "karthi 2", "karthi 2"]

//////////////////////////////////////////////////////////////

function filterUsingObject(array : any) {

const userJob = array.filter((user : any) => user.job === "developing").map((name : any) => name.name)

console.log(userJob);

}

const array = [

{name: "surya", job: "developing", id: 1},

{name: "karthi", job: "testing", id: 2},

{name: "karthi", job: "design", id: 2},

{name: "krishna", job: "developing", id: 3},

]

filterUsingObject(array) [LOG]: ["surya", "krishna"]

function getTotalMarks(marks: number[]) {

let total = 0

for(let mark of marks) {

total = total+mark

}

return total

}

function getPercentage(marks: number[]) {

let maxMark: number = 500;

let totalMark = getTotalMarks(marks);

return (totalMark /maxMark)\*100;

}

function getResult(marks: number[]) {

let percentage = getPercentage(marks);

return percentage >= 45 ? "pass" : "fail";

}

function arryMethods() {

let students = [

{name: "srk", username: "std001", password:11111,marks: [90,95,100,60,70]},

{name: "tsv", username: "std002", password:11113,marks: [91,96,100,50,45]},

{name: "kms", username: "std003", password:11113,marks: [92,94,99,50,60]},

{name: "svk", username: "std004", password:11114,marks: [91,92,95,40,50]},

{name: "svk", username: "std005", password:11115,marks: [91,92,95,60,40]}

]

interface objDetails {

name: string,

total: number

percentage: number

result: string

}

let details :any = []

let obj : objDetails = {

name: "",

total: 0,

percentage: 0,

result: ""

};

students.forEach((data) => {

let totalMark = getTotalMarks(data.marks)

let percentage = getPercentage(data.marks)

let result = getResult(data.marks)

obj = {

name: data.name,

total: totalMark,

percentage: percentage,

result: result

}

details.push(obj)

})

console.log(details)

}

arryMethods();

//////////////////////////////////////////////////////////////////////////

function reverseArray(val: string[]) {

const result = [];

for(let i= val.length - 1; i >= 0; i--) {

result.push(val[i]);

}

console.log(result);

}

reverseArray(["one","two","three","four","five"]);

//////////////////////////////////////////////////////////////////

Array methods

function arrayMethods() {

let names = ["srk","tsv","kms","svk","svk"]

let res = names.splice(1,3)

console.log(res); //splice array - ["tsv", "kms", "svk"]

console.log(names); //original array - ["srk", "svk"]

let res = names.slice(1,names.length-1)

console.log(res); //slice array - ["tsv", "kms", "svk"]

console.log(names); //original array -["srk", "tsv", "kms", "svk","svk"]

let res = names.splice(0);

console.log(res) //copy of array ["srk", "tsv", "kms", "svk", "svk"]

let res2 = [...names];

console.log(res2) //copy of array using spread operator["srk", "tsv", "kms", "svk", "svk"]

///////////////////////////////////////////////////////////////////////

let arr1 = [10,20,30];

let arr2 = [40,50,60];

// joins two array into single array

let cobain = arr1.concat(arr2);

console.log(cobain) //[10, 20, 30, 40, 50, 60]

let combain2 = [...arr1,...arr2]

console.log(combain2) //[10, 20, 30, 40, 50, 60]

let arr1 = [1001,2002,3003];

let resString = arr1.join()

console.log(resString); // joins an array as a string - "1001,2002,3003"

}

arrayMethods();

// implementation of filter & delete

let students = [

{name: "srk", username: "std001", password:11111,marks: [90,95,100,60,70]},

{name: "tsv", username: "std002", password:11113,marks: [91,96,100,50,45]},

{name: "kms", username: "std003", password:11113,marks: [92,94,99,50,60]},

{name: "svk", username: "std004", password:11114,marks: [11,12,15,40,50]},

{name: "svk", username: "std005", password:11115,marks: [91,92,95,60,40]}

]

function getTotalMarks(marks: number[]) {

let total = 0

for(let mark of marks) {

total = total+mark

}

return total

}

function getPercentage(marks: number[]) {

let maxMark: number = 500;

let totalMark = getTotalMarks(marks);

return +((totalMark /maxMark)\*100).toFixed(0);

}

function getResult(marks: number[]) {

let percentage = getPercentage(marks);

return percentage >= 45 ? "pass" : "fail";

}

function displayStudents(studentsArray : any) {

interface objDetails {

name: string,

total: number

percentage: number

result: string

}

let details :any = []

let obj : objDetails = {

name: "",

total: 0,

percentage: 0,

result: ""

};

studentsArray.forEach((data : any) => {

let totalMark = getTotalMarks(data.marks)

let percentage = getPercentage(data.marks)

let result = getResult(data.marks)

obj = {

name: data.name,

total: totalMark,

percentage: percentage,

result: result

}

details.push(obj)

})

console.log(details)

}

//displayStudents(students);

// implementation of filter

function filterStudents(filterText : string) {

let filteredStudents = [];

switch(filterText) {

case 'pass' :

filteredStudents = students.filter((std) => getResult(std.marks) === 'pass');

break;

case 'fail' :

filteredStudents = students.filter((std) => getResult(std.marks) === 'fail');

break;

default : filteredStudents = students;

}

displayStudents(filteredStudents);

}

filterStudents("pass")

//Implementation of delete student

function deleteStudent(userName : string) {

let index = students.findIndex((std) => std.username === userName);

students.splice(index,1);

displayStudents(students);

}

deleteStudent("std003")

Finding maximum and minimum number of array

function findMax() {

let number = [2,7,10,16,-1,14,-9,15,-7]

let max = number.reduce((acc,cur) => {

if(acc > cur)

return acc

else

return cur

},number[0])

console.log(max);

}

findMax()

///////////////////////////////////////////////////////

function findMin() {

let number = [2,7,10,16,-1,14,-9,15,-7]

let min = number.reduce((acc,cur) => {

if(acc > cur)

return cur

else

return acc

},number[0])

console.log(min);

}

findMin()

Using find() method get specific student details

function getTotal(mark : number[] | undefined) {

let total = 0;

mark?.reduce((acc,cur) => {

return total = acc + cur

},0)

return total

}

function getPercentage(mark : number[] | undefined) {

let maxMark = 500;

let totalMark = getTotal(mark)

let percentage = +((totalMark/maxMark) \* 100).toFixed(0);

return percentage

}

function getStatus(mark : number[] | undefined) {

let percentage = getPercentage(mark);

return percentage >= 45 ? "pass" : "fail"

}

function findStudentDetails(username : string, password : number) {

let students = [

{name: "srk", username: "std001", password:11111,marks: [90,95,100,60,70]},

{name: "tsv", username: "std002", password:11111,marks: [91,96,100,50,45]},

{name: "kms", username: "std003", password:11111,marks: [92,94,99,50,60]},

{name: "svk", username: "std004", password:11111,marks: [11,12,15,40,50]},

{name: "svk", username: "std005", password:11111,marks: [91,92,95,60,40]}

]

interface stdDetails {

name : string | undefined,

total : number | undefined,

percentage: number | undefined,

status: string | undefined

}

let stdDetails : stdDetails = {

name : "",

total : 0,

percentage: 0,

status: ""

}

let student = students.find((std) => {

return std.username === username && std.password === password

})

stdDetails = {

name : student?.name,

total : getTotal(student?.marks),

percentage: getPercentage(student?.marks),

status: getStatus(student?.marks)

}

return console.log(stdDetails);

}

let username = "std005"

let password = 11111

findStudentDetails(username, password)

Object and properties

let user : any = {

name: "srk",

age: 30,

job: "developer",

address: {town : "tgode", dist: "namakkal"}

}

// manipulation object properties

user.name = "tsv"

user.address = {town: "velechery", dist: "chennai"}

user.dob = "apr11"

user['name'] = "srk"

user.gender = "male"

user = {

name: "kms",

age: 30,

job: "developer",

address: {town : "tgode", dist: "namakkal"}

}

console.log(user);

////////////////////////////////////////////////////////////////

//Objects and methods

let person = {

name: "srk",

gender: "male",

doy: 1992,

calculateAge: function() {

return 2024 - this.doy

}

}

console.log(person.name);

console.log(person.calculateAge());

//Loop over objects in Javascript

let person = {

name: "srk",

age: "30",

dob: "july11",

gender: "male"

}

//To get keys of object

let objKeys = Object.keys(person);

console.log(objKeys);

//To get values of object

let objValues = Object.values(person)

console.log(objValues);

//To get both keys and values of object

let entries = Object.entries(person);

for(let [key,value] of entries) {

console.log(`${key} : ${value}`);

}

//Destructuring a object

let person : any = {

namee: "srk",

age: 30,

dob: "july11",

area:["london","uk"],

workDetails: {experience: 5, company: "google"}

};

// it takes reference its properties

//let{name,age,dob} = person;

let{

name : firstname,

age : a,

dob : db,

area: [city,country],

workDetails : {experience, company : org}} = person;

console.log(firstname);

console.log(db);

console.log(experience);

console.log(org);

console.log(city);

console.log(country);

function returnObj() {

return {id: 101, user: "srk"}

}

let{id,user} = returnObj();

console.log(id);

console.log(user);

Destructuring an array

let person : any = ["srk",30,"july11",["london","uk"]];

// it takes reference its position

let[name,age,dob,[city,country]] = person;

console.log(name);

console.log(city);

function destructureArray() {

return ["srk",30,"july11"]

}

let[name,age,dob] = destructureArray()

console.log(name);

console.log(dob);

////////////////////////////////////////////////////////

Count of the given number

function countOfNumber(val : number) {

let num = val;

let count = 0;

while(num > 0) {

count++;

num = Math.floor(num/10);

}

console.log(count);

}

countOfNumber(983471)//6

///////////////////////////////////////////////////

Sum of the given number

function sumOfNumber(val : number) {

let num = val;

let rem = 0;

let sum = 0;

while(num > 0) {

rem = num % 10;

sum += rem;

num = Math.floor(num/10);

}

console.log(sum);

}

sumOfNumber(983471);//32

//Converting array to object

let products :any = [

{id: "a001",product: "mobile", prize: 11000},

{id: "a002",product: "tv", prize: 7000},

{id: "a003",product: "washing machine", prize: 9000}

]

function arrToObj() {

let productsObj :any = {}

products.forEach((val : any) => {

if(!productsObj[val.id]){

productsObj[val.id] = val

}

})

return productsObj

}

console.log(arrToObj());

//////////////////////////////////////////////////////////////////////////

let products :any = [

{id: "a001",product: "mobile", prize: 11000},

{id: "a002",product: "tv", prize: 7000},

{id: "a003",product: "washing machine", prize: 9000},

]

function reducer(acc :any,cur :any) {

return {...acc,[cur.id] : cur}

}

let newProducts = products.reduce(reducer,{});

console.log(newProducts)

[LOG]: {

"a001": {

"id": "a001",

"product": "mobile",

"prize": 11000

},

"a002": {

"id": "a002",

"product": "tv",

"prize": 7000

},

"a003": {

"id": "a003",

"product": "washing machine",

"prize": 9000

}

}

//Converting object to array

function objToArr() {

let product :any = {id: "a001",product: "mobile", prize: 11000}

let resultArr : any = [];

let keys = Object.keys(product);

keys.map((key) => resultArr.push([key,product[key]]))

console.log(resultArr)

}

objToArr();

[LOG]: [["id", "a001"], ["product", "mobile"], ["prize", 11000]]

//Count of duplicate in array

function countDuplicates(val : number[]) {

const res = val.reduce((acc:any,cur:any) => {

if(acc[cur]){

acc[cur] += 1;

}

else{

acc[cur] = 1

}

return acc;

},{})

console.log(res);

}

countDuplicates([1,2,1,3,5,3,2,6,4,3])

function countDuplicatesInObject() {

const user = [

{name: "srk",age:28},

{name: "tsv",age:25},

{name: "kms",age:28},

{name: "svk",age:25},

{name: "mgr",age:26},

{name: "rk",age:28},

]

const sumAge = user.reduce((acc :any,cur :any) => {

if(acc[cur.age]) {

acc[cur.age] = ++acc[cur.age]

}

else{

acc[cur.age] = 1

}

return acc

},{})

console.log(sumAge);

}

countDuplicatesInObject();

// swap array elements

function swapArrayElements(val: string[]) {

const temp = val[val.length - 1];

val[val.length -1] = val[0];

val[0] = temp;

console.log(val);

}

swapArrayElements(["a","b","c","d","e"]) //["e", "b", "c", "d", "a"]

///////////////////////////////////////////////////////////////////

function swapArrayElements(val: string[]) {

//destructure

//[val[0],val[val.length-1]] = [val[val.length-1],val[0]];

[val[0],val[1],val[val.length -2],val[val.length-1]] = [val[1],val[0],val[val.length-1],val[val.length-2]];

console.log(val);

}

swapArrayElements(["a","b","c","d","e"])

//["e", "b", "c", "d", "a"]

//["b", "a", "c", "e", "d"]

// string reverse

function reverseString(val: string) {

let split = val.split(" ");

let str1 = split[0];

let str2 = split[1];

let str3 = split[split.length -1];

let revStr1 = ""

let revStr2 = ""

split.forEach((strVal) => {

for(let i = strVal.length -1; i >= 0 ; i --) {

if(str1 === strVal) {

revStr1 += strVal[i];

}

if(str3 === strVal){

revStr2 += strVal[i];

}

}

})

console.log(revStr1 +" "+str2+" "+revStr2);

}

reverseString("Welcome to Chennai"); //"emocleW to iannehC"

//////////////////////////////////////////////////////////

function reverseString(val: string) {

let valArr = val.split(" ");

[valArr[0],valArr[valArr.length -1]] =

[valArr[valArr.length - 1], valArr[0]];

console.log(valArr.join(" "));

}

reverseString("Welcome to Ooty"); //"Ooty to Welcome"

////////////////////////////////////////////////////

function reverseString(val: string) {

let arrVal = val.split(" ");

let revVal = "";

for(let i = arrVal.length -1; i >= 0; i--) {

revVal += arrVal[i] + " ";

}

console.log(revVal);

}

reverseString("Welcome to Ooty"); //"Ooty to Welcome"

//replace all vowels

function replaceVowels(value:string) {

let result = "";

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

let val = value.charAt(i)

if(val == 'a' || val == 'e' ||

val == 'i' || val == 'o' || val == 'u'){

val = "@"

}

else{

val = val

}

result += val;

}

console.log(result);

}

replaceVowels("welcome"); //"w@lc@m@"

///////////////////////////////////////////////////////////////

function replaceVowels(val:string) {

let result = val.replace(/[aeiouAEIOU]/g,"@")

console.log(result);

}

replaceVowels("WelcomE"); //"W@lc@m@"

// count of vowels

function countVowels(val :string) {

let vowelCnt = 0;

let consonantCnt = 0;

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

let chr = val[i];

if(chr == 'a' || chr == 'e' || chr == 'i' ||

chr == 'o' || chr == 'u') {

vowelCnt++;

}

else {

consonantCnt++;

}

}

console.log("count of vowels " + vowelCnt);

console.log("count of consonants " + consonantCnt);

}

countVowels("education");

//check valid mobile number

function checkValidMobileNo(val:string) {

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

let nonSplCrt = val.replace(/[^a-zA-Z0-9]/g, "");

let nonAlphabet = nonSplCrt.replace(/[^a-zA-Z]/g,"")

if(val.length == 10 && !nonAlphabet.length){

return "valid"

}

else{

return "invalid"

}

}

}

console.log(checkValidMobileNo("96888Ajhd@#$&\*55555")); // invalid

///////////////////////////////////////////////////////////////////

//changing first letter alone caps

function changeToCaps(val :string) {

let arrVal = val.split(" ");

let reqArr :any = [];

arrVal.forEach((val) => {

reqArr.push(val.charAt(0).toUpperCase() + val.substring(1));

});

console.log(reqArr.join(" "));

}

changeToCaps("welcome to chennai");// "Welcome To Chennai"

function changeToCaps(val :string) {

let result = ""

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

if(i % 2 == 0) {

result += val[i].toUpperCase();

}

else{

result += val[i]

}

}

console.log(result);

}

changeToCaps("narendramodi");//"NaReNdRaMoDi"

//////////////////////////////////////////////////////////////

//Insert an array in middle of another array

function insertArray() {

let arr1 = ["one","two","three","four","five"];

let arr2 = ["a","b","c"];

const middleIndex = Math.floor(arr1.length/2)

arr1.splice(middleIndex,0,...arr2)

console.log(arr1);

}

insertArray(); //["one", "two", "a", "b", "c", "three", "four", "five"]

/////////////////////////////////////////////////////////////////////

//replace <tags> as our own

function replaceTags() {

let emailTemplate = "we, at <company>, believe in providing the best online services. thanks <Company>"

let regex = /<company>/gi

let result = emailTemplate.replace(regex ,"dsquad");

console.log(result);

}

replaceTags(); //"we, at dsquad, believe in providing the best online services. thanks dsquad"

/////////////////////////////////////////////////////////////

//find only name,mobileNo,cardNo

function findValueFromArray() {

let datas = [

"surya kumar","123-123-5678","1234 1234 5678 1234",

"vijay kumar","123-123-5678","1234 1234 5678 1234",

"karthik kumar","123-123-5678","1234 1234 5678 1234",

"mathi kumar","123-123-5678","1234 1234 5678 1234"

]

let regex = /[a-z]+\s[a-z]+/gi //name

let regex1 = /[0-9]{3}-[0-9]{3}-[0-9]{4}/g //mobileno

let regex2 = /[0-9]{4}\s[0-9]{4}\s[0-9]{4}\s[0-9]{4}/g //card

datas.forEach((entry) => {

let res = entry.match(regex); //regex1,regex2

console.log(res);

})

}

findValueFromArray();

//count of each vowels

function countEachVowels() {

let val = "coimbatore is a nice place"

let A = 0; let E = 0; let I = 0; let O = 0; let U = 0;

let alphabeticCount = 0;

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

if(val[i] == " ") {

continue;

}

alphabeticCount++;

if(val[i] == 'a'){

A++;

}

if(val[i] == 'e'){

E++;

}

if(val[i] == 'i'){

I++;

}

if(val[i] == 'o'){

O++;

}

if(val[i] == 'u'){

U++;

}

}

let totalVowels = A+E+I+O+U;

let totalConsonant = alphabeticCount - totalVowels;

console.log("count of 'a' " + A); //3

console.log("count of 'e' " + E); //3

console.log("count of 'i' " + I); //3

console.log("count of 'o' " + O); //3

console.log("count of 'u' " + U); //0

console.log("count of 'vowels' " + totalVowels); //11

console.log("count of 'consonants' " + totalConsonant); //11

}

countEachVowels();

///////////////////////////////////////////////////////////////////////

//count of caps and small letters

function countCapsSmall() {

let val = "Coimbatore Is a Nice Place"

let countCap = 0;

let countSmall = 0

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

if(val[i] == ' ') {

continue;

}

if(val[i] < 'a'){ //ascii value for a-z is 97-122 & //ascii value for A-Z is 65-90

countCap++;

}

else {

countSmall++;

}

}

console.log("count of caps " + countCap);

console.log("count of small " + countSmall);

}

countCapsSmall(); //"count of caps 4" "count of small 18"

//pagination for set of dataset

function paginationForDatas(pagNo : number) {

const datas: any = [

{id: 1, name: "a"},{id: 2, name: "b"},{id: 3, name: "c"},

{id: 4, name: "d"},{id: 5, name: "e"},{id: 6, name: "f"},

{id: 7, name: "g"},{id: 8, name: "h"},{id: 9, name: "i"},

{id: 10, name: "j"},{id: 11, name: "k"},{id: 12, name: "l"},

]

const itemPerPage: number = 3;

const selectedPgNo : number = pagNo;

const startIndex : number = (selectedPgNo - 1) \* itemPerPage;

const paginationData =

datas.slice(startIndex, startIndex +itemPerPage);

console.log(paginationData);

}

const selectPageNo: number = 3

paginationForDatas(selectPageNo);

//[{"id": 7,"name": "g"}, {"id": 8,"name": "h"}, {"id": 9,"name": "i"}]

// search specific term in a word from dataset

function searchSpecificTerm(term :string) {

const datas = ["My", "Name", "is", "Happy"];

const result = datas.filter((val) => val.includes(term));

console.log(result);

}

const searchingTerm: string = "pp"

searchSpecificTerm(searchingTerm); //["Happy"]

// check prime number

function primeNumber(val:number) {

let num = 2;

let prime = true;

while(val > num){

if(val % num == 0) {

console.log("Not a prime number")

prime = false;

break;

}

num++;

}

if(prime == true) {

console.log("Its a prime number")

}

}

primeNumber(15); // "Not a prime number"

//prime numbers up to given number

function printPrimeNumbers(val:number) {

for(let limit = 2; limit < val; limit ++) {

let prime = true;

for(let i = 2; i<limit; i++) { //2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19

if(limit % i == 0) {

//console.log(limit + " is not a prime");

prime = false;

}

}

if(prime === true) { //2,3,5,7,11,13,17,19

console.log(limit + " is a prime");

}

}

}

printPrimeNumbers(20);

////////////////////////////////////////////////////////////////

//remove selected obj from objArray

function addremoveObj(productName? : string) {

let totalProducts = [

{id: 101, type: "shirt"},

{id: 102, type: "t-shirt"},

{id: 103, type: "pant"},

{id: 104, type: "shorts"},

{id: 105, type: "formal"},

{id: 106, type: "casual"},

]

const reqIndex = totalProducts.findIndex((product) => product.type === productName);

totalProducts.splice(reqIndex,1);

console.log(totalProducts);

}

const selectedProduct: string = "formal"

addremoveObj(selectedProduct);

//add and remove selected obj from objArray

function addremoveObj(delProduct? : string,addProduct? : string) {

let totalProducts = [

{id: 101, type: "shirt"},

{id: 102, type: "t-shirt"},

{id: 103, type: "pant"},

{id: 104, type: "shorts"},

{id: 105, type: "formal"},

{id: 106, type: "casual"},

]

let cart = [

{id: 101, type: "shirt"},

{id: 102, type: "t-shirt"},

{id: 103, type: "pant"},

{id: 104, type: "shorts"},

]

let cartProducts : any = [...cart];

//removing product from cartProducts by 'type'

if(delProduct) {

cartProducts = cart.filter((val) => val.type !== delProduct);

}

//adding product from totalProducts to cartProducts by 'type'

if(addProduct) {

let selectedProduct = totalProducts.

find((product) => product.type === addProduct);

//console.log(selectedProduct);

cartProducts.push(selectedProduct);

}

console.log(cartProducts);

}

const deleteProduct: string = "shirt"

const addProduct: string = ""

addremoveObj(deleteProduct, addProduct);

////////////////////////////////////////////////////////////

//find longest word in sentence

function longestWordInSentence(val : string) {

const valArr = val.split(" ");

let result = valArr.reduce((acc :any,cur :any) => {

if(acc.length > cur.length) {

return acc

}

else{

return cur

}

},valArr[0])

console.log(result);

}

longestWordInSentence("be good do good");