1. Complete the function that accepts a string parameter, and reverses each word in the string. **All** spaces in the string should be retained.

"This is an example!" ==> "sihT si na !elpmaxe" "double spaces" ==> "elbuod secaps"

**My Answer:**

function reverseWords(string) {

var arr = [];

var string2 = string.split(' ');

for (let i = 0; i < string2.length; i++) {

arr.push(string2[i] = string2[i].split('').reverse().join(''));

}

// arr.join();

return arr.join().replace(/,/g, ' ');

}

2. An isogram is a word that has no repeating letters, consecutive or non-consecutive. Implement a function that determines whether a string that contains only letters is an isogram. Assume the empty string is an isogram. Ignore letter case.

**My Answer:**

function isIsogram(string) {

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

var a = string.substr(i + 1, string.length - 1); //'ba'

/\*console.log(a);\*/

var b = string[i].toUpperCase(); // 'A'

/\*console.log(b);\*/

if (a.includes(string[i]) || a.includes(b)) { // 'ba' includes 'a' or 'A' includes 'S' // 'ring' includes 't' or 'T'

return false;

}

}

return true;

}

console.log(isIsogram("Dermatoglyphics"));

console.log(isIsogram("aaaba"));

console.log(isIsogram("moOse"));

3. The new"Avengers" movie has just been released! There are a lot of people at the cinema box office standing in a huge line. Each of them has a single 100, 50 or 25 dollar bill. An "Avengers" ticket costs 25 dollars.

Vasya is currently working as a clerk. He wants to sell a ticket to every single person in this line.

Can Vasya sell a ticket to every person and give change if he initially has no money and sells the tickets strictly in the order people queue?

Return YES, if Vasya can sell a ticket to every person and give change with the bills he has at hand at that moment. Otherwise return NO.

tickets([25, 25, 50]) // => YES tickets([25, 100]) // => NO. Vasya will not have enough money to give change to 100 dollars tickets([25, 25, 50, 50, 100]) // => NO. Vasya will not have the right bills to give 75 dollars of change (you can't make two bills of 25 from one of 50)

**My Answer:**

function tickets(bills) {

var total = 0;

var billst = [];

var a = [0, 0, 0];

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

if (bills[i] === 100) {

if ((a[0] >= 1) && (a[1] >= 1)) {

a[1]--;

a[0]--;

billst.push(bills[i]);

a[2]++;

} else {

a[0] = a[0] - 3;

billst.push(bills[i]);

a[2]++;

}

total += 25;

} else if (bills[i] === 50 && a[0] >= 1) {

billst.push(bills[i]);

a[1]++;

a[0]--;

total += 25;

} else if (bills[i] === 25) {

billst.push(bills[i]);

a[0]++;

total += 25;

}

}

console.log(bills);

console.log(billst)

console.log(a);

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

if ((total >= 0) && (a[0] >= 0) && (a[1] >= 0) && (a[2] >= 0) && (bills.length === billst.length)) {

return 'YES';

} else {

return 'NO';

}

}

}

3. Write a function that will return the count of **distinct case-insensitive** alphabetic characters and numeric digits that occur more than once in the input string. The input string can be assumed to contain only alphabets (both uppercase and lowercase) and numeric digits.

**Example**

"abcde" -> 0 # no characters repeats more than once  
"aabbcde" -> 2 # 'a' and 'b'  
"aabBcde" -> 2 # 'a' occurs twice and 'b' twice (`b` and `B`)  
"indivisibility" -> 1 # 'i' occurs six times  
"Indivisibilities" -> 2 # 'i' occurs seven times and 's' occurs twice  
"aA11" -> 2 # 'a' and '1'  
"ABBA" -> 2 # 'A' and 'B' each occur twice

**My Answer:**

function duplicateCount(text) {

var textt = text.toLowerCase();

var arr = textt.split('');

var newArr = [];

var count = 0;

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

if (textt.substring(i + 1, textt.length).includes(textt[i]) && !newArr.includes(textt[i])) {

newArr.push(textt[i]);

count++;

}

}

4. We have received some raw data and we have to calculate how many years of experience all employees have together.

const people = [

{

name: 'Joe Schmoe',

yearsExperience: 1,

department: 'IT'

},

{

name: 'Sally Sallerson',

yearsExperience: 15,

department: 'Engineering'

},

{

name: 'Bill Billson',

yearsExperience: 5,

department: 'Engineering'

},

{

name: 'Jane Janet',

yearsExperience: 11,

department: 'Management'

},

{

name: 'Bob Hope',

yearsExperience: 9,

department: 'IT'

}

];

**My Answer:**

function getTotalExperience(results) {

var total = 0;

results.forEach(function(item) {

total += item['yearsExperience'];

});

return total;

}

console.log(getTotalExperience(people)); // 41

5. We have received some raw data and we have to calculate how many people there are in each department.

const people = [

{

name: 'Joe Schmoe',

yearsExperience: 1,

department: 'IT'

},

{

name: 'Sally Sallerson',

yearsExperience: 15,

department: 'Engineering'

},

{

name: 'Bill Billson',

yearsExperience: 5,

department: 'Engineering'

},

{

name: 'Jane Janet',

yearsExperience: 11,

department: 'Management'

},

{

name: 'Bob Hope',

yearsExperience: 9,

department: 'IT'

}

];

**My Answer:**

function howManyPeopleInEachDepartment(results) {

var final = {};

results.forEach(function(item) {

if (!final.hasOwnProperty(item['department'])) {

final[item['department']] = 1;

} else if (final.hasOwnProperty(item['department'])) {

final[item['department']]++;

};

});

return final;

}

console.log(howManyPeopleInEachDepartment(people));

6. Return object with largest poster area.

var boxarts = [

{ width: 200, height: 200, url: "http://cdn-0.nflximg.com/images/2891/Fracture200.jpg" },

{ width: 150, height: 200, url: "http://cdn-0.nflximg.com/images/2891/Fracture150.jpg" },

{ width: 300, height: 200, url: "http://cdn-0.nflximg.com/images/2891/Fracture300.jpg" },

{ width: 425, height: 150, url: "http://cdn-0.nflximg.com/images/2891/Fracture425.jpg" }

];

**My Answer:**

var myF = boxarts.reduce(function (a, b) {

if (a.width \* a.height > b.width \* b.height) {

return a;

}

return b;

});

7. Implement the function\_in\_unique order, which takes as argument a sequence and returns a list of items without any elements with the same value next to each other and preserving the original order of elements.

**My Answer:**

var uniqueInOrder=function(iterable) {

var newArray = [];

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

if (iterable[i] !== iterable[i+1]) {

newArray.push(iterable[i]);

}

}

return newArray;

}