let contacts ={

    'John Smith': +123456584

}

contacts['Lisa Smith'] = +4564632489;

console.log(contacts);

{ 'John Smith': 123456584, 'Lisa Smith': 4564632489 }

let contacts ={

    'John Smith': +123456584,

}

let value = 'Lisa Smith';

contacts[value] = +4564632489

console.log(contacts);

{ 'John Smith': 123456584, 'Lisa Smith': 4564632489 }

let value = 'Lisa Smith';

let contacts ={

    'John Smith': +123456584,

    [value]: +4564632489

}

console.log(contacts);

{ 'John Smith': 123456584, 'Lisa Smith': 4564632489 }

let value = 'Lisa Smith';

let contacts ={

    'John Smith': +123456584,

    [value]: +4564632489

}

console.log(contacts);

let phone = contacts[value];

console.log(phone);

{ 'John Smith': 123456584, 'Lisa Smith': 4564632489 }

4564632489

let contacts ={

    'John Smith': +123456584,

    'Lisa Smith': +4564632489,

    'John Doe': +2546851557,

    'Pesho Ivanov': +4587811287,

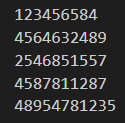
    'Gosho Hristov': +48954781235

}

for(let key in contacts){

console.log(contacts[key]);

}



let contacts ={

    'John Smith': +123456584,

    'Lisa Smith': +4564632489,

    'John Doe': +2546851557,

    'Pesho Ivanov': +4587811287,

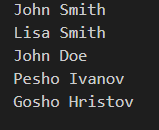
    'Gosho Hristov': +48954781235

}

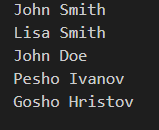
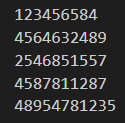
for(let key in contacts){

console.log(key);

}

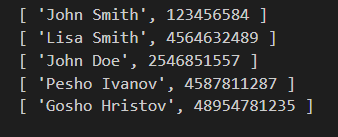


Object.keys(contacts).forEach(x => console.log(x));

Object.values(contacts).forEach(x => console.log(x));

Object.entries(contacts).forEach(x => console.log(x));



function solve(arr){

    let obj = {};

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

        let input = arr[i].split(' ');

        let name = input[0];

        let number = input[1];

        obj[name] = number;

        // console.log(input);

    }

        for(let key in obj){

                console.log(`${key} -> ${obj[key]}`)

            }

}

solve([

    'Tim 0834212554',

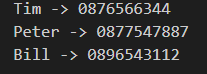
    'Peter 0877547887',

    'Bill 0896543112',

    'Tim 0876566344'

  ]);

  ]);



let contacts ={

    'John Smith': ['+123456584','+1233548615'],

    'Lisa Smith': +4564632489,

    'John Doe': +2546851557,

    'Pesho Ivanov': +4587811287,

    'Gosho Hristov': +48954781235

}

console.log(contacts['John Smith'][1]);

+1233548615

function solve(arr){

    let obj = {};

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

        let [name, number] = arr[i].split(' ');

        obj[name] = number;

    }

        Object.keys(obj).forEach(key => console.log(`${key} -> ${obj[key]}`));                     }

solve([

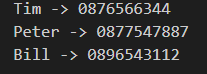
        'Tim 0834212554',

        'Peter 0877547887',

        'Bill 0896543112',

        'Tim 0876566344'

]);



let contacts = new Map();

console.log(contacts);

contacts.set('John Smith', '+35421583');

contacts.set('Joe Doe', '+7895421586');

console.log(contacts);

contacts.delete('Joe Doe');

if(contacts.has('Joe Doe')){

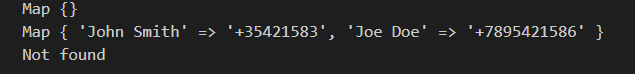
console.log('Found her');

console.log(contacts.get('Joe Doe'));

} else {

    console.log('Not found');

}



let object = {

    name: 'Pesho',

    age: 30,

}

delete object.name;

console.log(object);

let contacts = new Map();

contacts.set('John Smith', '+35421583');

contacts.set('Joe Doe', '+7895421586');

console.log(contacts.keys());

[Map Iterator] { 'John Smith', 'Joe Doe' }

Връща итератор - нещо, което може да бъде обходено

let contacts = new Map();

contacts.set('John Smith', '+35421583');

contacts.set('Joe Doe', '+7895421586');

// console.log(contacts.keys());

let iterator = contacts.keys();

for(let key of iterator){

    console.log(key);

}



for(let value of contacts.values()){

    console.log(value);

}

Може и с променлива итератор, може и директно в цикъла. Ползва се for of цикъл за целта.....може да се търсят ключовете, а може и стойностите

for(let entries of contacts.entries()){

    console.log(entries);

}

function solve(input){

    let storage = new Map();

    input.forEach(line => {

        let [item, quantity] = line.split(' ');

        quantity = Number(quantity);

        if(storage.has(item)){

            let oldQuantity = storage.get(item);

            storage.set(item, oldQuantity + quantity);

        } else {

            storage.set(item, quantity);

        }

    });

    // console.log(storage);

    // for(let key of storage.keys()){

    //     console.log(`${key} -> ${storage.get(key)}`);

    // }

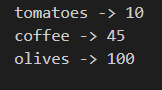
    for(let kvp of storage.entries()){

        console.log(`${kvp[0]} -> ${kvp[1]}`);

    }

}

solve([ 'tomatoes 10', 'coffee 5', 'olives 100', 'coffee 40' ]);

Set - гарантира уникални стойности, ако му дадеш нещо ново,връща същото

let names = ['Pesho', 'Gosho', 'Pesho'];

let uniqueNamesSet = new Set(names);

let uniqueNames = Array.from(uniqueNamesSet);

console.log(uniqueNames);

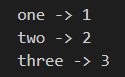
[ 'Pesho', 'Gosho' ]

let map = new Map();

map.set('one', 1);map.set('two', 2);map.set('three', 3);

let sorted = Array.from(map.entries())

    .sort((a, b) => a[1] - b[1]);    for(let kvp of sorted){

        console.log(`${kvp[0]} -> ${kvp[1]}`);     }

function solve(input){

    let words = {};

  input.forEach(word => {

      if(words.hasOwnProperty(word)){

        words[word]++;

      } else {

      words[word] = 1;

      }

  });

// console.log(words);

let wordEntries = Object.entries(words);

// console.log(wordEntries);

let sortedWordEntries = wordEntries.sort((a, b) => b[1] - a[1]);

    // let wordCountA = a[1];

    // let wordCountB = b[1];

    // console.log(sortedWordEntries);

for(let kvp of sortedWordEntries){

    console.log(`${kvp[0]} -> ${kvp[1]} times`);

}

}

solve([

    'Here',     'is',

    'the',      'first',

    'sentence', 'Here',

    'is',       'another',

    'sentence', 'And',

    'finally',  'the',

    'third',    'sentence'

  ]

  );

function solve(input){

let students = {};

    input.forEach(line => {

        let lineArr = line.split(' ');

        // console.log(lineArr);

        let name = lineArr.shift();

        let grades = lineArr.map(Number);

        // console.log(name);

        // console.log(grades);

        if(students.hasOwnProperty(name)){

            let oldGrades = students[name];

            let allGrades = oldGrades.concat(grades);

            students[name] = allGrades;

        } else {

            students[name] = grades;

        }

    });

    // console.log(students);

    let studentsEntries = Object.entries(students);

    // console.log(studentsEntries);

    let sorted = studentsEntries.sort((a, b) =>  {

    let studentGradeA = a[1];

    let studentGradeB = b[1];

    let avrGradeA = studentGradeA.reduce((acc, a) => acc + a, 0) / studentGradeA.length;

    let avrGradeB = studentGradeB.reduce((acc, b) => acc + b, 0) / studentGradeB.length;

    return avrGradeA - avrGradeB;

    });

    // console.log(sorted);

    for(let kvp of studentsEntries){

        console.log(`${kvp[0]}: ${kvp[1].join(', ')}`);

    }

}

solve([ 'Lilly 4 6 6 5', 'Tim 5 6', 'Tammy 2 4 3', 'Tim 6 6' ]);

function solve(input){

    let neighborhoodNames = input

        .shift()

        .split(', ');

    // console.log(neighborhoodNames);

    let neighborhoods = {};

    neighborhoodNames.forEach(name => {

        neighborhoods[name] = [];

    });

//    console.log(neighborhoods);

   input.forEach(line => {

       let[name, person] = line.split(' - ');

       if(neighborhoods.hasOwnProperty(name)){

            neighborhoods[name].push(person);

       }

   });

//    console.log(neighborhoods);

   let neighborhoodEntries = Object.entries(neighborhoods);

   let sorted = neighborhoodEntries.sort((a,b) => b[1].length - a[1].length);

//    console.log(sorted);

for(let kvp of sorted){

    console.log(`${kvp[0]}: ${kvp[1].length}`);

        for(let person of kvp[1]){

            console.log(`--${person}`);

        }

    }

}

solve([

    'Abbey Street, Herald Street, Bright Mews',

    'Bright Mews - Garry',

    'Bright Mews - Andrea',

    'Invalid Street - Tommy',

    'Abbey Street - Billy'

  ]

  );

function solve(input){

    let storage = {};

    input.forEach(line => {

        let [item, quantity] = line.split(' ');

        quantity = Number(quantity);

        if(storage.hasOwnProperty(item)){

            let oldQuantity = storage[item];

            storage[item] = oldQuantity + quantity;

        } else {

            storage[item] = quantity;

        }

    });

    let arr = Object.entries(storage);

    // console.log(storage);

    for(let kvp of arr){

        console.log(`${kvp[0]} -> ${kvp[1]}`);

    }

}solve([ 'tomatoes 10', 'coffee 5', 'olives 100', 'coffee 40' ]);

function solve(arr){

    let obj = {};

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

        let input = arr[i].split(' ');

        let name = input[0];

        let number = input[1];

        obj[name] = number;

        // console.log(input);

    }

        for(let key in obj){

                console.log(`${key} -> ${obj[key]}`)

            }

}

function solve(arr){

    let obj = {};

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

        let [name, number] = arr[i].split(' ');

        obj[name] = number;

    }

        Object.keys(obj).forEach(key => console.log(`${key} -> ${obj[key]}`));                     }

solve(['Tim 0834212554','Peter 0877547887','Bill 0896543112',

        'Tim 0876566344']);

***1.Word Tracker***

function solve(input){

    let wordsNeeded = input.shift().split(' ');

    // console.log(wordsNeeded);

    let words = {};

    wordsNeeded.forEach(word => {

        words[word] = 0;

    });

    input

        .slice(1)

        .forEach(line => {

            if(Object.keys(words).includes(line)){

                words[line]++;

            }

        });

    // console.log(words);

    // let wordsKeys = Object.keys(words);

    // let sorted = wordsKeys.sort((a, b) => words[b] - words[a]);

    // console.log(sorted);

    for(let key of Object.keys(words).sort((a, b) => words[b] - words[a])){

        console.log(`${key} - ${words[key]}`);

    }

}

solve([

    'this sentence', 'In',

    'this',          'sentence',

    'you',           'have',

    'to',            'count',

    'the',           'occurances',

    'of',            'the',

    'words',         'this',

    'and',           'sentence',

    'because',       'this',

    'is',            'your',

    'task'

  ]

  );

2.