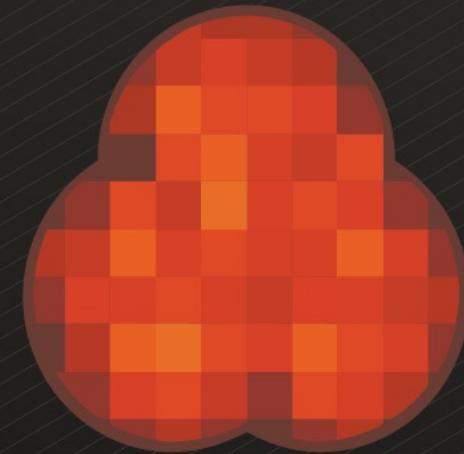


LJUBAVNI KALKULATOR

Tomislav Jakopec

2021
KULENDAYZ
IT INNOVATION CONFERENCE



SLOWDOWN.
03-05 September
in Osijek, Croatia

Thanks to sponsors and organizers

ORGANIZED BY



MSCommunity

OSIJEK
SOFTWARE
{ITY_}



PMI
PMI Udruga Hrvatska

BIOS
Poduzetnički Inkubator



Akademija
za umjetnost
i kulturu u Osijeku

SPONSORED BY



Microsoft



FERIT

KING ICT
INFORMATION & COMMUNICATION TECHNOLOGIES

NEPHOS

citus grupa
Internet Development Company

spin[®]
INFORMATICA



MEDIA PARTNERS

cropc.net
Tehnologija kao stil života

CARNet
HRVATSKA AKADEMSKA I ISTRAŽIVAČKA MREŽA
CROATIAN ACADEMIC AND RESEARCH NETWORK

programiranje

programiranje, priređivanje ili odabir skupa naredbi prema kojima će, ovisno o promjeni određenih ulaznih vrijednosti, djelovati neki automatski uređaj, stroj, sustav ili dr. (→ [AUTOMAT](#)). Pojam se može odnositi na jednostavno namještanje kakve elektromehaničke ili elektron. naprave za vođenje nekoga procesa po unaprijed zadanome vremenskom rasporedu, kakve u kućanstvima služe za uključivanje i isključivanje grijanja ili hlađenja po dnevnom ili tjednom programu, upravljanje radom perilice rublja ili snimanja radijskih ili televizijskih emisija. U užem smislu, pojam se odnosi na izradbu programa za rad uređaja zasnovanih na [mikroprocesoru](#), napose elektroničkih računala (→ [RAČUNALNI PROGRAM](#)). Zadaci koji se rješavaju računalom mogu biti vrlo jednostavnii i lako razumljivi (primjerice rješavanje kvadratne jednadžbe), ali i složeni (vođenje poslovanja velikoga poduzeća). Pripremanje složenih programa djelatnost je koja zahtijeva znanja iz onoga područja iz kojega potječe zadatak, znanja o algoritmima prikladnim za njegovo rješavanje te znanja i vještine pisanja programa, pa se zbog složenosti pristupa danas ta djelatnost sve češće naziva [programskim inženjerstvom](#). Prema tom je pristupu programiranje samo jedan od koraka u pripremanju programa, tj. korak njegove stvarne izrade (konkretnizacije). Programiranje se provodi odabirom prikladnih tipova podataka i njihovom organizacijom u odgovarajuću strukturu, te nizanjem naredbi odabranoga programskog jezika. Program napisan u višem programskom jeziku (koji izgleda kao posebno oblikovan tekst), tzv. izvorni oblik programa, posebnim se programom za prevodenje prevodi u strojni oblik pogodan za izvođenje na računalu. Kako programiranje sve složenijih programa postaje sve teže, taj se posao na različite načine pokušava pojednostavniti. Tako se skupine naredbi koje se češće pojavljuju može objediti u tzv. potprogram ili proceduru i koristiti ih kao jednu složenu operaciju. Ako se uz to realni objekt opiše prikladnom strukturo podataka i pripreme procedure za djelovanje nad tom strukturo, riječ je o objektno-orientiranom programiranju. U najnovije doba, uvođenjem dodatnih formalizama, objektno orientirano programiranje prerasta u komponentno orientirano programiranje, kod kojega se programi sastavljaju od unaprijed pripremljenih komponenti. Veze među komponentama uspostavljaju se na osnovi pravila komuniciranja opisanih njihovim sučeljima, a povezivanjem komponenti uspostavljaju se odgovarajuće strukture programa, tzv. arhitektura programskoga sustava.

programiranje. Hrvatska enciklopedija, mrežno izdanje. Leksikografski zavod Miroslav Krleža, 2021. Pristupljeno 3. 9. 2021. <<http://www.enciklopedija.hr/Natuknica.aspx?ID=50555>>.

Računalno programiranje

Programiranje je pisanje uputa [računalu](#) što i kako učiniti, a izvodi se u nekom od [programskega jezika](#). Programiranje je umjetnost i umijeće u stvaranju programa za računala. Stvaranje programa sadrži u sebi pojedine elemente dizajna, umjetnosti, znanosti, matematike kao i inžinjeringu. Osoba koja stvara program zove se [programer](#).

Sadržaj [[sakrij](#)]

1. [Programski jezici](#)
2. [Razvoj softvera](#)
3. [Predlošci](#)
4. [Vidi još](#)
5. [Izvori](#)
6. [Vanjske poveznice](#)

Programski jezici [[uredi](#) | [uredi kôd](#)]

Glavni članak: [Programski jezik](#)

Programi ili upute za računalo pišu se u programskom jeziku upotrebom određene sintakse i pravila koja vrijede za svaki [programski jezik](#) (ili tip), koji se potom prevodi u [strojni jezik](#) koje je osobito za određeno računalo te je ovisno o njegovoj arhitekturi. Prevodenje s višeg programskog jezika na strojni provodi se putem [programa prevodioca](#) (kompajler) ili se naredbe u višem jeziku izravno prevode preko takozvanog [p_koda](#) u strojni jezik.

Računalno programiranje. Wikipedia. Pristupljeno 3.9.2021 <https://hr.wikipedia.org/wiki/Ra%C4%8Dunalno_programiranje>.

Programiranje je rješavanje **problema**



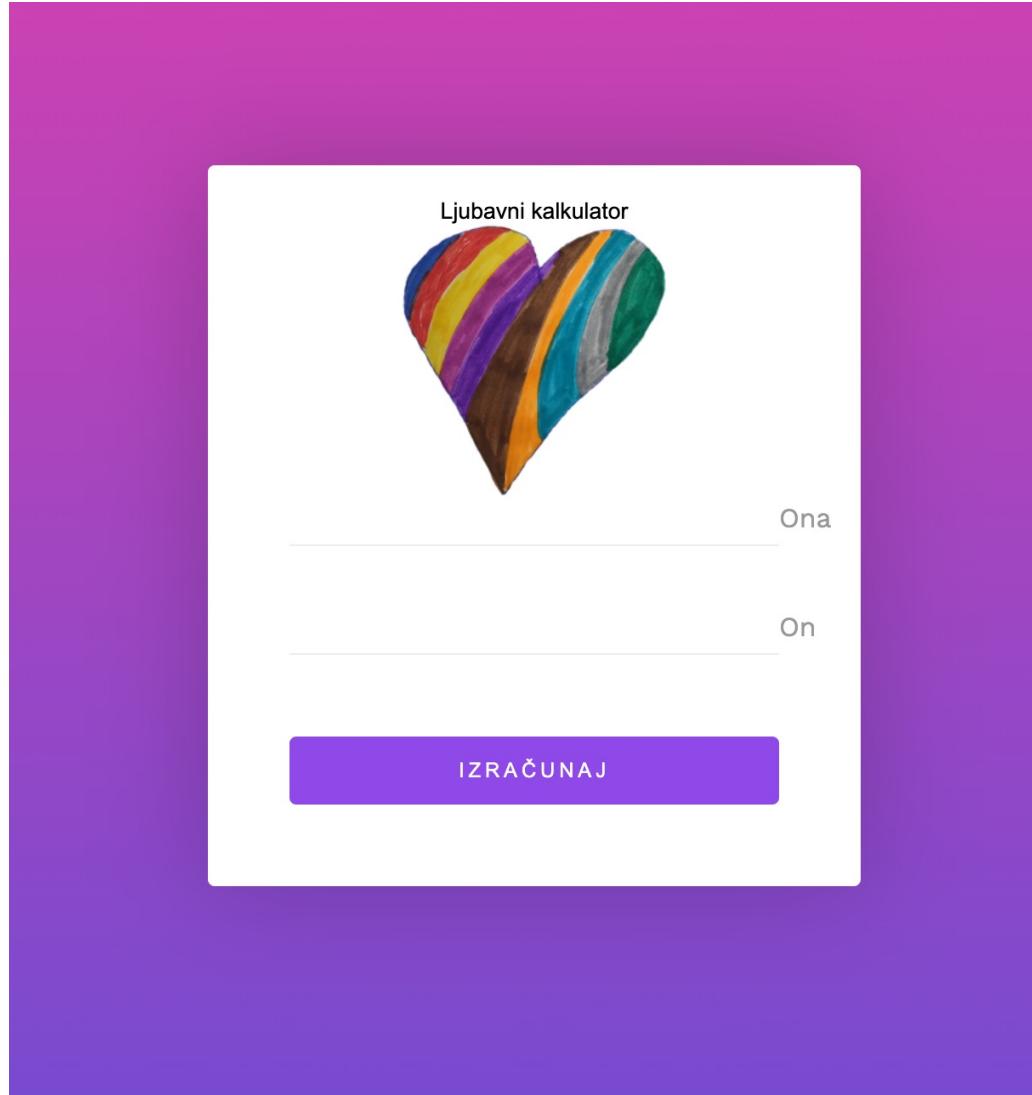
Mi smo u
problemu
jer nemamo
problem







Rješenje



2021. netko je već sigurno nešto napravio



ljubavni kalkulator

Ljubavni kalkulator

Komentara • 1 minuta čitano



♥ ♥ ♥ Ljubavni Kalkulator ♥ ♥ ♥

Reklamni Prostor:

Upišite svoje ime i ime partnera ili prijatelja u tekst polja, pa sa klikom na "Izračunaj" na osnovu vaših upisanih imena, proverite vašu zajedničku kalkulaciju ljubavi i kompatibilnosti u procentima.
Rezultat ove ljubavne kalkulacije ćete videti u srcu između upisanih imena.

Reklamni Prostor:

Ovde upišite Svoje Ime 100% Ime Partnera ili Prijatelja:

Izračunaj

Ljubavni kalkulator

Što ljubavni kalkulator radi

Odredite vjerojatnost svoje ljubavi. Pokušajte vidjeti ima li vaša ljubav šanse. Stvarno smo naporno radili kako bi ovaj algoritam funkcionirao. To je 100% legitimna metoda za upoznavanje svoje ljubavi.

Tvoje ime

Unesite svoje ime

Njegovo / njezino ime

Svoju simpatiju:-)

Izračunati

Test ljubavi

Napiši svoje ime i ime svoje simpatije.

Nakon toga ispod svakog slova napiši koliko se puta ono ponavlja. Žbroji prvi broj ispod svog imena i posljednji broj ispod imena tvoje simpatije, i tako ponovi sa svakim sljedećim brojem. Kad završiš s prvim retkom, prijeđi na zbrajanje brojeva u drugom retku, a onda u trećem. Na kraju, kad dobiješ zbroj u četvrtom retku, saznat ćeš koliko posto šanse imaš razviti ljubav sa svojom simpatijom. Što si bliže broju 100, to bolje! Sretno!

MARTA MANUEL

23113 231111

342262

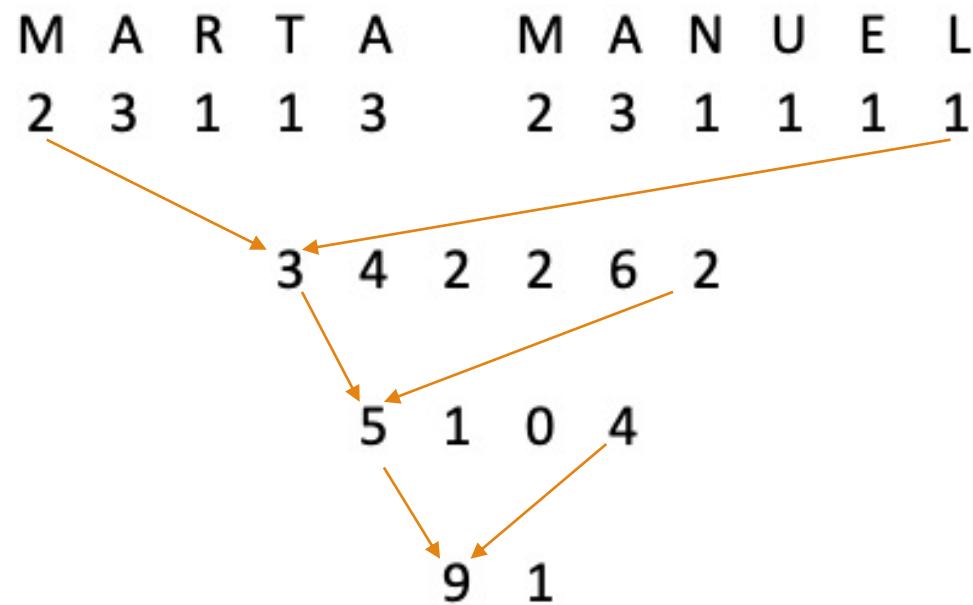
5104

91

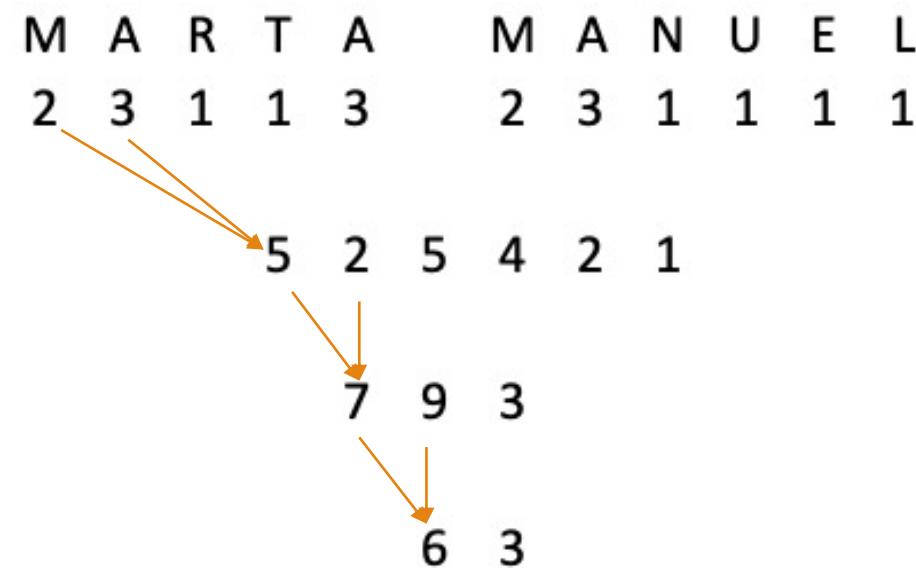


Algoritmi kao rješenje

Algoritam 1



Algoritam 2



Algoritam n

Razmišljati izvan okvira

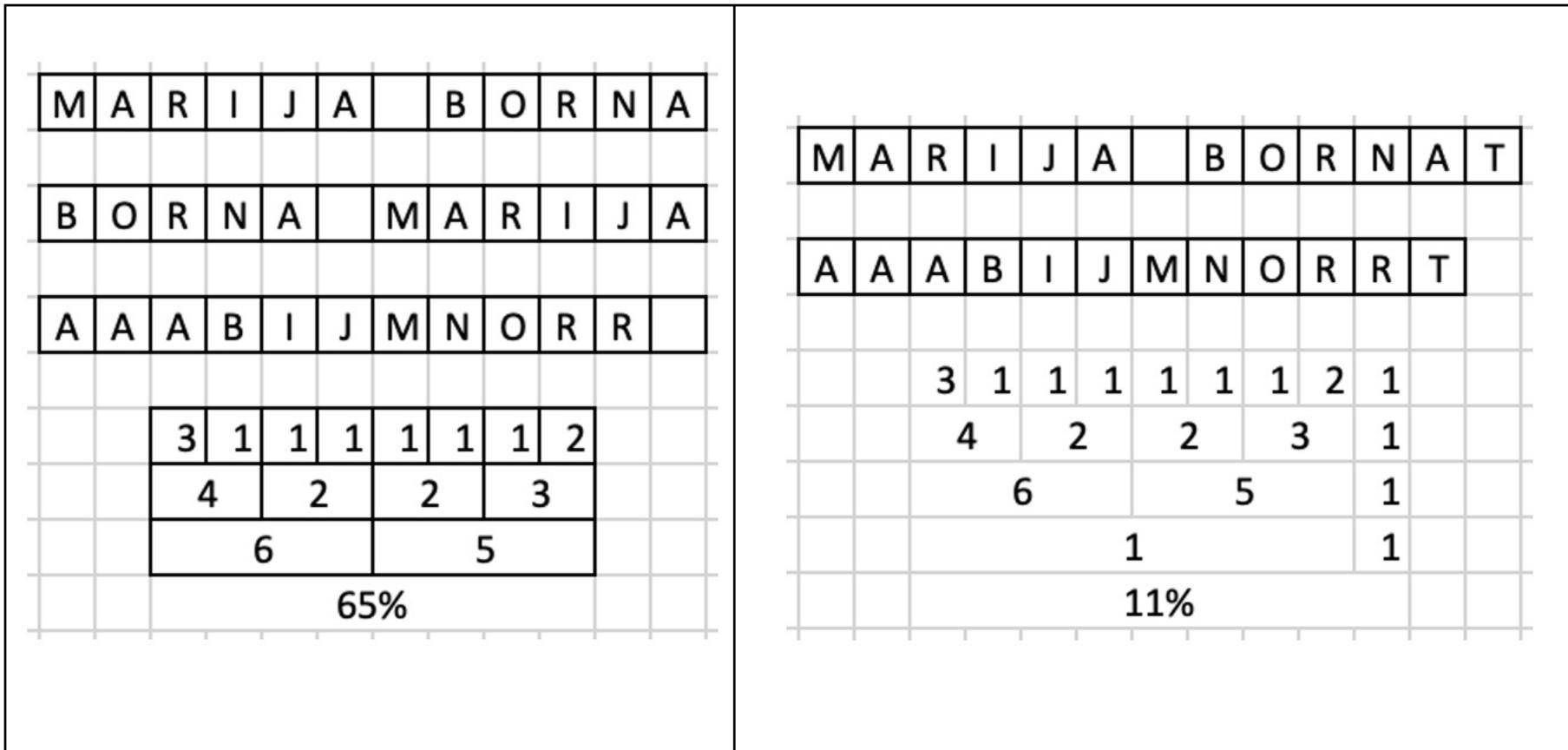
Algoritam 1

M	A	R	T	A	M	A	N	U	E	L
2	3	1	1	3	2	3	1	1	1	1
3	4	2	2	6	2					
5	1	0	4							
9	1									

Algoritam 1

M	A	N	U	E	L	M	A	R	T	A
2	3	1	1	1	1	2	3	1	1	3
5	4	2	4	3	1					
6	7	6								
1	2	7								
3	7									

Prijedlog rješenja



Koraci koji dovode do rješenja

1. Učitat će se dva niza znakova (prvo i drugo ime)
2. Posložit će se svi znakovi u novi jedinstveni niz u kojem će znakovi biti posloženi abecedno
3. Brojat će se slova iz nastalog i sortiranog niza te nastati niz brojeva. Ako se neko slovo pojavi više od 9 puta uzeti će se samo jedinični dio broja
4. Dva susjedna broja će se zbrajati i u redu ispod će se upisati jedinična vrijednost novo dobivenog broja. Ako zbroj bude veći od 9 uzeti će se samo jedinični dio broja.
 - Ovaj postupak će se ponavljati sve dok ne dobijemo broj koji je manji od 100.
5. Rezultat je postotna vrijednost koliko se to dvoje voli

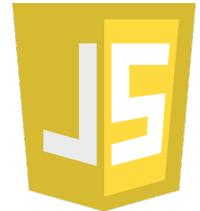
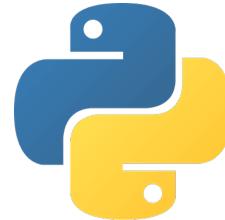
Odabir programskog jezika



Sigurno ćeš pogriješiti

vs

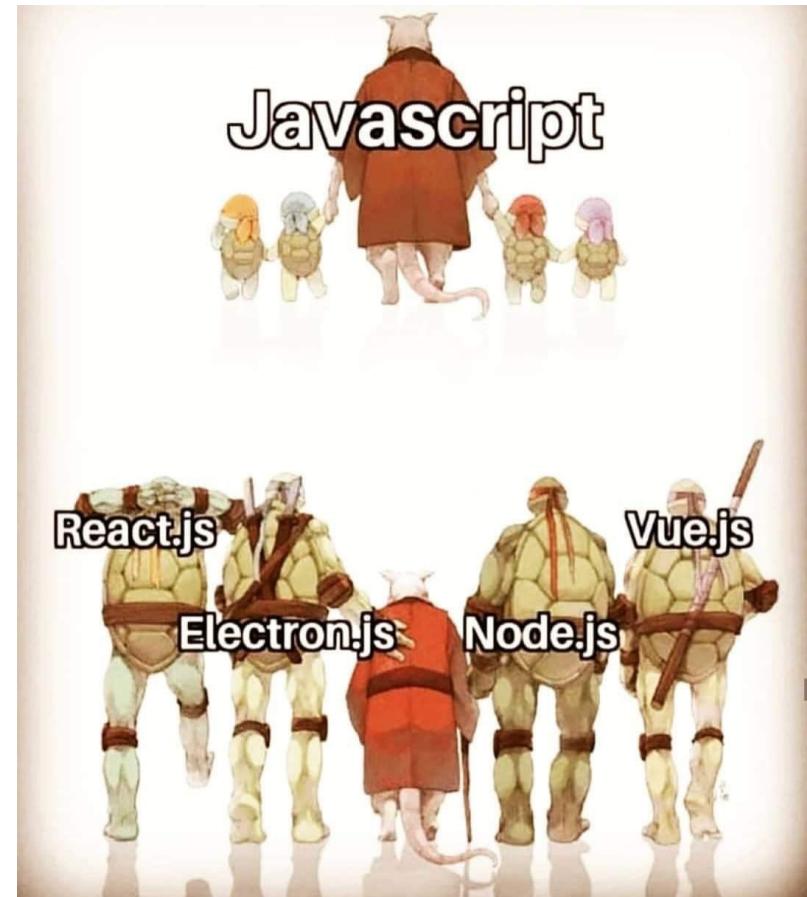
Svejedno je s kojim krenuti



Zašto ova dva jezika



Iako se nigdje izrijekom ne spominje, Python je de facto implementacijski jezik kurikuluma iz informatike u osnovnim i srednjim školama.



Programiranje

1. Učitati će se dva niza znakova (prvo i drugo ime)

Python

```
def unos(poruka):
    while True:
        ime = input(poruka)
        if ime.strip()=='':
            print('Unos obavezan')
        elif ime.isalpha():
            return ime
        else:
            print('Unos smije sadržavati samo slova')

def ucitaj_imena():
    while True:
        ime1=unos('Inesi svoje ime: ')
        ime2=unos('Unesi ime za provjeru: ')
        if ime1!=ime2:
            return {'ime1':ime1, 'ime2':ime2}
        else:
            print('Imena moraju biti različita')
```

JavaScript

```
function unos(poruka){
    var regex = /^[A-Za-z]+$/;
    while(true){
        let ime=prompt(poruka);
        if (ime.trim().length==0){
            prompt('Unos obavezan');
        } else if (ime.match(regex)){
            return ime;
        }else{
            prompt('Unos smije sadržavati samo slova');
        }
    }
}

function ucitajImena(){
    while(true){
        let ime1 = unos('Inesi svoje ime:');
        let ime2 = unos('Unesi ime za provjeru:');
        if (ime1!=ime2){
            return {'ime1': ime1, 'ime2': ime2};
        }else{
            prompt('Imena moraju biti različita');
        }
    }
}
```

2. Posložiti će se svi znakovi u novi jedinstveni niz u kojem će znakovi biti posloženi abecedno

Python

```
def sortiraj(imena):  
  
    lista = []  
    for slovo in imena['ime1'].lower():  
        lista.append(slovo)  
    for slovo in imena['ime2'].lower():  
        lista.append(slovo)  
    lista.sort()  
    return lista
```

JavaScript

```
function sortiraj(imena){  
    var lista = [];  
    for (var i = 0; i < imena.ime1.length; i++) {  
        lista.push(imena.ime1.toLowerCase().charAt(i));  
    }  
    for (var i = 0; i < imena.ime2.length; i++) {  
        lista.push(imena.ime2.toLowerCase().charAt(i));  
    }  
    return lista.sort();  
}
```

Programiranje

3. Brojati će se slova iz nastalog i sortiranog niza te nastati niz brojeva. Ako se neko slovo pojavi više od 9 puta uzeti će se samo jedinični dio broja

Python

```
def prebroji(lista):
    rjecnik = dict((i, lista.count(i)) for i in lista)
    brojevi = []
    for slovo in rjecnik:
        brojevi.append(rjecnik[slovo])
    return brojevi
```

JavaScript

```
function prebroji(lista){
    var mapa = new Map();
    for (const kljuc of lista) {
        mapa.set(kljuc, 0);
    }
    for (const kljuc of lista) {
        let ukupno = mapa.get(kljuc);
        mapa.set(kljuc, ukupno + 1);
    }
    return Array.from(mapa.values());
}
```

Programiranje

4. Dva susjedna broja će se zbrajati i u redu ispod će se upisati jedinična vrijednost novo dobivenog broja. Ako zbroj bude veći od 9 uzeti će se samo jedinični dio broja

- Ovaj postupak će se ponavljati sve dok ne dobijemo broj koji je manji od 100

Python

```
def izracunaj_postotak(brojevi):
    broj = int(''.join(str(b) for b in brojevi))
    print(broj)
    if broj < 100:
        return broj
    else:
        zbroj=[]
        if len(brojevi) % 2 == 0:
            for i in range(0,len(brojevi),2):
                suma = brojevi[i] + brojevi[i+1]
                if suma >= 10:
                    suma=suma % 10
                zbroj.append(suma)
        else:
            for i in range(0,len(brojevi)-1,2):
                suma = brojevi[i] + brojevi[i+1]
                if suma >= 10:
                    suma=suma % 10
                zbroj.append(suma)
        zbroj.append(brojevi[-1])
    return izracunaj_postotak(zbroj)
```

Programiranje

4. Dva susjedna broja će se zbrajati i u redu ispod će se upisati jedinična vrijednost novo dobivenog broja. Ako zbroj bude veći od 9 uzeti će se samo jedinični dio broja

- Ovaj postupak će se ponavljati sve dok ne dobijemo broj koji je manji od 100

JavaScript

```
function izracunajPostotak(brojevi){  
    let broj=parseInt(brojevi.join(''));  
    if(broj<100){  
        return broj;  
    }else{  
        var zbroj=[];  
        if (brojevi.length % 2 === 0){  
            for(var i=0;i<brojevi.length;i++){  
                var suma = brojevi[i] + brojevi[i+1];  
                if(suma>=10){  
                    suma = suma % 10;  
                }  
                zbroj.push(suma);  
            }  
        }else{  
            for(var i=0;i<brojevi.length;i++){  
                var suma = brojevi[i] + brojevi[i+1];  
                if(suma>=10){  
                    suma = suma % 10;  
                }  
                zbroj.push(suma);  
            }  
            zbroj.push[brojevi[brojevi.length-1]];  
        }  
        return izracunajPostotak(zbroj);  
    }  
}
```

Programiranje

5. Rezultat je postotna vrijednost koliko se to dvoje voli

Python

```
def ljk():
    rjecnik_imena = ucitaj_imena()
    sortirana_lista = sortiraj(rjecnik_imena)
    brojevi_slova=prebroji(sortirana_lista)
    postotak = izracunaj_postotak(brojevi_slova)
    print (rjecnik_imena['ime1'],'i',
           rjecnik_imena['ime2'],'se vole',
           postotak,'%')

ljk()
```

JavaScript

```
function ljk(){
    let rjecnikImena = ucitajImena();
    let sortiranaLista = sortiraj(rjecnikImena);
    let brojeviSlova=prebroji(sortiranaLista);
    let postotak = izracunajPostotak(brojeviSlova);
    document.write(rjecnikImena.ime1 +
                   ' i ' + rjecnikImena.ime2 +
                   ' se vole ' + postotak + ' %');
}

ljk();
```

Izvođenje programa

Python

KulenDayz2021/Python/ljubavni_kalkulator.py

Inesi svoje ime: Borna

Unesi ime za provjeru: Marija

Borna i Marija se vole 65 %

JavaScript

Na ovoj se stranici navodi sljedeće

Inesi svoje ime:

Borna

Odustani

U redu

Na ovoj se stranici navodi sljedeće

Unesi ime za provjeru:

Marija

Odustani

U redu

Borna i Marija se vole 65 %

Izvorni kod dostupan na

<https://github.com/tjakopec/KulenDayz2021>



Zadaci

1. Ispravite u kodu tekstualni tipfeler
2. Dio koda koji se ponavlja dva puta izdvojiti u def/funkciju – testirati uspješnost promjene koda
3. Prilagoditi kod da za dva imena (Vaše ime i ime Vaše simpatije) uvijek ispisuje 99%

HVALA!

kulendayz 2021.

I HAVE SPOKEN

there will be more of the speaking at KD afterparty

