




UUM – 2. kolokvij

Otvori pomoću aplikacije



Zadatak 1

a) Generisati i sabrati čiste tonove frekvencije 300 Hz i 700 Hz, frekvencije uzrokovanja $f_s=4000$ Hz. Generisani zvučni signal prikazati u vremenskom i frekvencijskom domenu.

b) Projektovati visokopropusni filter ako su zadani ulazni parametri: frekvencija uzrokovanja $f_s=44000$ Hz, granica propusnog pojasa $W_p=3000$ Hz, granica nepropusnog pojasa $W_s=2900$ Hz, valovitost u propusnom pojasu $R_p=1$ Db i slabljenje u nepropusnom pojasu $R_s=40$ dB.

```
% a) Generisati i sabrati čiste tonove frekvencije 300 Hz i 700 Hz,
frekvencije uzrokovanja fs=4000 Hz. Generisani zvučni signal prikazati u
vremenskom i frekvencijskom domenu
% b) Projektovati visokopropusni filter ako su zadani ulazni parametri:
frekvencija uzrokovanja fs=44000 Hz, granica propusnog filtera ako su
Wp=3000 Hz, granica nepropusnog pojasa Ws=2900 Hz, valovitost u propusnom
pojasu Rp=1 Db i slabljenje u nepropusnom pojasu Rs=40 dB
% 1. a)
Fs=4000;
T=1/Fs;
L=1000;
t=(0:L-1)*T;

y=0.7*sin(2*pi*300*t)+sin(2*pi*700*t);
subplot(211)
plot(t(1:200),y(1:200))
title('Sinusni signal');
xlabel('Vrijeme');
ylabel('Amplituda');

NFFT=2^nextpow2(L);
Y=fft(y,NFFT)/L;
f=Fs/2*linspace(0,1,NFFT/2+1);
subplot(212)
plot(f,2*abs(Y(1:NFFT/2+1)),'r')
xlabel('Frekvencija');
ylabel('Funkcija y');

%% b)
clear all
clc

fs=44000;
Wp=[2*3000/fs];
Ws=[2*2900/fs];
Rp=1;
Rs=40;
```

```

d=fdesign.highpass('Fst,Fp,Ast,Ap',Ws,Wp,Rs,Rp);
f=design(d,'cheby2');
info(f)
fvtool(f)
[Y,FS]=wavread('matlab.wav');
t=0:1/fs:1/fs*(length(Y)-1);
plot(t,Y)
xlabel('Vrijeme [s]')
ylabel('Amplituda')
hold on
izlaz=filter(f,Y);
plot(t,izlaz,'r-')
legend('Ulazni signal','Filtrirani signal')

```

ZADATAK 3

Nacrtati graf funkcije $f = \sin(3\theta)$ u polarnim koordinatama i izvršiti njegovu rotaciju za puni krug u vremenu od dvije sekunde. Također nacrtati plohu $z = \sin(x^2 + y^2)$ i izvršiti njegovu rotaciju za dva kruga u vremenu od 5 sekundi.

```

% 3. nacrtati graf funkcije f=sin(3theta) u polarnim koordinatama i
izvršiti njegovu rotaciju za puni krug u vremenu od 2s. Također nacrtati
plohu z=sin(x^2+y^2) i izvršiti njegovu rotaciju za dva kruga u vremenu od
5s
figure
theta=0:pi/100:2*pi;
rho=sin(theta*3);
subplot(2,1,1),h=polar(theta,rho,'go');
pause
for i=1:60
    zdir=[0 0 1];
    center=[0 0 0];
    pause(2/60)
    rotate(h,zdir,6,center)
end
pause
[x,y]=meshgrid([-2:0.1:2]);
z=sin(x.^2+y.^2);
subplot(2,1,2),h=surf(x,y,z)
pause
axis tight
for i=1:100
    zdir=[0 0 -1];
    center=[0 0 0];
    rotate(h,zdir,7.2,center)
    pause(5/100)
end

```

broj riječi u tekstu.

Zadatak 2. Učitati sliku sa računara u boji i zatim izvršiti filtriranje upotrebom filter: 'motion', 'sobel', 'log', 'disk', 'unsharp'. Originalnu sliku i filtrirane slike prikazati naredbom subplot.

```
%2. Učitati sliku sa racunara u boji i zatim izvorsiti filtriranje upotrebom
filter: 'motion', 'sobel', 'log', 'disk', 'unsharp'. Originalnu sliku i
filtrirane slike prikazati naredbom subplot
s=imread('buket.jpg');
I=im2double(s);
subplot(3,2,1),imshow(I), title('Original');

H=fspecial('motion',30,60);
Motion=imfilter(I,H,'replicate');
subplot(3,2,2),imshow(Motion),title('Motion');

H=fspecial('sobel');
Sobel=imfilter(I,H,'replicate');
subplot(3,2,3),imshow(Sobel),title('Sobel');

H=fspecial('log',[30,100],0.2);
Log=imfilter(I,H,'replicate');
subplot(3,2,4),imshow(Log),title('Log');

H=fspecial('disk',20);
Disk=imfilter(I,H,'replicate');
subplot(3,2,5),imshow(Disk),title('Disk');

H=fspecial('unsharp');
Unsharp=imfilter(I,H,'replicate');
subplot(3,2,6),imshow(Unsharp),title('Unsharp');
```

Zadatak 4.

Nacrtati četverostranu piramidu čija je dužina stranica osnove $a=3$, a visina piramide $h=4$? Izračunati površinu i zapreminu piramide.

```
%4. Nacrtati četverostranu piramidu cija je duzina stranica osnove a=3, a
visina piramide h=4? Izracunati povrsinu i zapreminu piramide
clear all
close all

a=3;
h=4;
P=a^2 + 2*a*h;
V=a^2*h./3;

fprintf('Povrsina je:');
disp(P);
fprintf('Zapremina je:');
disp(V);
```

```
x = [0 3 3 0];
y = [0 0 3 3];
z = [0 0 0 0];
patch(x,y,z, 'y');
grid on;
axis([-2 5 -2 5]);
xlabel('x-osa');
ylabel('y-osa');
zlabel('z-osa');
```

```
x = [0 3 1.5];
y = [0 0 1.5];
z = [0 0 4];
patch(x,y,z, 'r');
grid on;
axis([-2 5 -2 5]);
xlabel('x-osa');
ylabel('y-osa');
zlabel('z-osa');
```

```
x = [0 0 1.5]
y = [0 3 1.5];
z = [0 0 4];
patch(x,y,z, 'g');
grid on;
axis([-2 5 -2 5]);
xlabel('x-osa');
ylabel('y-osa');
zlabel('z-osa');
```

```
x = [0 3 1.5];
y = [3 3 1.5];
z = [0 0 4];
patch(x,y,z, 'b');
grid on;
axis([-2 5 -2 5]);
xlabel('x-osa');
ylabel('y-osa');
zlabel('z-osa');
```

```
x = [3 3 1.5];
y = [0 3 1.5];
z = [0 0 4];
patch(x,y,z, 'm');
grid on;
axis([-2 5 -2 5]);
xlabel('x-osa');
ylabel('y-osa');
zlabel('z-osa');
```

ZADATAK 4

Unositi riječi upotrebom tastature sve dok se ne unese riječ koja počinje sa slovom 'i' a završava sa slovom 'd' te u sebi ima samo samoglasnike. Tu riječ treba pretvoriti u velika slova. Zatim sve unesene riječi napisati kao niz na način da svaki samoglasnik bude ispisan velikim slovom osim zadnje riječi koja je već ispisana velikim slovima. Prebrojati unesene riječi.

%4. Unositi rijeci upotrebom tastature sve dok se ne unese rijec koja pocinje sa slovom 'i', a završava sa slovom 'd' te u sebi ima samo samoglasnike. Tu rijec treba pretvoriti u velika slova. Zatim sve unesene rijeci napisati kao niz na način da svaki samoglasnik bude ispisan velikim slovom osim zadnje rijeci koja je već ispisana velikim slovima. Prebrojati unesene rijeci

```
clc;
clear;
```

```
% Inicijalizacija varijabli
```

```
rijeci = {}; % Lista za spremanje unesenih riječi
```

```
samoglasnici = 'aeiou'; % Samoglasnici
```

```
konacnaRijec = ''; % Posljednja riječ koja zadovoljava uvjet
```

```
brojac = 0; % Brojač unesenih riječi
```

```
while true
```

```
    % Unos riječi s tastature
```

```
    rijec = input('Unesite riječ: ', 's');
```

```
    rijeci{end + 1} = rijec;
```

```
    brojac = brojac + 1;
```

```
    % Provjera da li riječ zadovoljava uvjet
```

```
    if startsWith(rijec, 'i') && endsWith(rijec, 'd')
```

```
        % Provjera da li riječ sadrži samo samoglasnike (osim prvog i posljednjeg slova)
```

```
        sredina = rijec(2:end-1);
```

```
        if all(ismember(sredina, samoglasnici))
```

```
            konacnaRijec = upper(rijec);
```

```
            rijeci{end} = konacnaRijec; % Zamjena originalne riječi sa riječi u velikim slovima
```

```
            break;
```

```
        end
```

```
    end
```

```
end
```

```
% Pretvaranje svih riječi u niz s velikim samoglasnicima, osim zadnje riječi
```

```
for i = 1:length(rijeci) - 1
```

```
    rijec = rijeci{i};
```

```
    for j = 1:length(rijec)
```

```
        if ismember(rijec(j), samoglasnici)
```

```
            rijec(j) = upper(rijec(j));
```

```
        end
```

```

        end
        rijeci{i} = rijec;
    end

% Ispis rezultata
fprintf('Ukupan broj unesenih riječi: %d\n', brojac);
fprintf('Sve unijete riječi sa velikim samoglasnicima osim zadnje:\n');
disp(rijeci(1:end-1));
fprintf('Zadnja riječ:\n%s\n', konacnaRijec);

```

Zadatak 3: Program pročitaj jednu rečenicu. Nakon toga odredi najdužu riječ koja se nalazi u rečenici. Tu najdužu riječ ispiše naopako (npr. Danas je srijeda. -> rijec srijeda -> ispiše adejirs). Nakon toga ispiše datu rečenicu sa novom riječi. Definirati izuzetke u programu. Voditi računa o većem broju praznih mjesta koji se mogu naći između riječi, u riječi ne spadaju zarez, tačka itd.

%3. Program procita jednu recenicu. Nakon toga odredi najduzu rijec koja se nalazi u recenici. Tu najduzu rijec ispise naopako (npr. Danas je srijeda. ->rijec srijeda -> ispise adejirs). Nakon toga ispise datu recenicu s novom rijeci. Definirati izuzetke u programu. Voditi racuna o vecem broju praznih mjesta koji se mogu naci izmedju rijeci, u rijeci ne spadaju zarez, tacka itd

```

clear all
clc

```

```

try
    % Unos rečenice s tastature
    recenica = input('Unesite rečenicu: ', 's');

    % Zamjena zareza i tačaka s praznim mjestima
    recenica = regexprep(recenica, '[,\.]', '');

    % Razdvajanje riječi i uklanjanje praznih mjesta
    rijeci = strsplit(recenica);
    rijeci = rijeci(~cellfun('isempty', rijeci));

    if isempty(rijeci)
        error('Rečenica ne sadrži valjane riječi.');
```

```

    end

    % Pronalaženje najduže riječi
    najduzaRijec = '';
    for i = 1:length(rijeci)
        if length(rijeci{i}) > length(najduzaRijec)
            najduzaRijec = rijeci{i};
        end
    end

    % Okretanje najduže riječi
    okrenutaRijec = flip(najduzaRijec);

    % Zamjena najduže riječi u rečenici s okrenutom riječju

```

```

for i = 1:length(rijeci)
    if strcmp(rijeci{i}, najduzaRijec)
        rijeci{i} = okrenutaRijec;
    end
end

% Ispis rezultata
novaRecenica = strjoin(rijeci, ' ');
fprintf('Originalna rečenica: %s\n', recenica);
fprintf('Najduža riječ: %s\n', najduzaRijec);
fprintf('Okrenuta riječ: %s\n', okrenutaRijec);
fprintf('Nova rečenica: %s\n', novaRecenica);

catch ME
    % Obrada izuzetaka
    fprintf('Došlo je do greške: %s\n', ME.message);
end

```

Zadatak 1. Unijeti tekst sa tastature i provjeriti koliko u njemu ima slogova 'sni' i na kojim indeksima se ponavlja. Pronaći koliko taj tekst sadrži praznih mjesta i na kojim indeksima se nalaze. Provjeriti koliko u tekstu ima rečenica te koliko ima upitnih, uzvičnih i izjavnih, i ispisati broj riječi u tekstu.

1. Unijeti tekst sa tastature i provjeriti koliko u njemu ima slogova 'sni' i na kojim indeksima se ponavlja. Pronaći koliko taj tekst sadrži praznih mjesta i na kojim indeksima se nalaze. Provjeriti koliko u tekstu ima rečenica te koliko ima upitnih, uzvičnih i izjavnih, i ispisati broj riječi u tekstu

```

% Unos teksta sa tastature
tekst = input('Unesite tekst: ', 's');

% Pronalazak svih pojavljivanja sloga 'sni'
slog = 'sni';
slogIndeksi = strfind(tekst, slog);
brojSlogova = length(slogIndeksi);

% Pronalazak svih pojavljivanja praznih mjesta
praznaMjesta = find(tekst == ' ');
brojPraznihMjesta = length(praznaMjesta);

% Brojanje rečenica
recenice = regexp(tekst, '[.!?]', 'split');
brojRecenica = length(recenice) - 1;

% Brojanje upitnih, uzvičnih i izjavnih rečenica
brojUpitnih = length(find(tekst == '?'));
brojUzvicnih = length(find(tekst == '!'));
brojIzjavnih = length(find(tekst == '.'));

```

```

% Brojanje riječi u tekstu
rijeci = strsplit(tekst);
brojRijeci = length(rijeci);

% Ispis rezultata
fprintf('Broj pojavljivanja sloga ''sni'': %d\n', brojSlogova);
fprintf('Indeksi pojavljivanja sloga ''sni'': ');
disp(slogIndeksi);

fprintf('Broj praznih mjesta: %d\n', brojPraznihMjesta);
fprintf('Indeksi praznih mjesta: ');
disp(praznaMjesta);

fprintf('Broj rečenica: %d\n', brojRecenica);
fprintf('Broj upitnih rečenica: %d\n', brojUpitnih);
fprintf('Broj uzvičnih rečenica: %d\n', brojUzvicnih);
fprintf('Broj izjavnih rečenica: %d\n', brojIzjavnih);
fprintf('Broj riječi u tekstu: %d\n', brojRijeci);

```

Zadatak 3

Učitati proizvoljni zvučni signal. Izvršiti linearno pojačavanje zvučnog signala, zatim napraviti inverziju signala.

```

%3. Učitati proizvoljni zvucni signal. Izvrsiti linearno pojacavanje
zvucnog signala, zatim napraviti inverziju signala
clc
clear all
close all

[zvuk,fs] = audioread('matlab.wav');
ramp = 0:1/(length(zvuk)-1):1;
linearno_pojacanje = zvuk.*ramp';
inverzija = zvuk .* (1 - ramp');
t = 0:1/fs:1/fs*(length(zvuk)-1);
subplot(211)
plot(t,zvuk)
title('Ulazni zvucni signal');
axis([0 2 -1 1])
subplot(212)
plot(t,linearno_pojacanje)
hold on
plot(t, inverzija, 'g', 'LineWidth', 2);
plot(t, ramp, 'r', 'LineWidth', 2);
title('Linearno pojacanje i inverzija signala');
xlabel('Vrijeme (s)');
ylabel('Amplituda');
legend('Linearno pojacanje', 'Inverzija', 'Ramp', 'Location', 'best');
axis([0 2 -1 1]);

```


Zadatak 2

Učitati proizvoljnu RGB sliku, pa sačuvati u zasebnim folderima original i crvenu komponentu, original i plavu komponentu i original i zelenu komponentu. Komponente kreirati koristeći funkciju

`function [boja]=komponenta (slika, RGB)`

% 2. Učitati proizvoljnu RGB sliku, pa sačuvati u zasebnim folderima original i crvenu komponentu, original i plavu komponentu i original i zelenu komponentu. Komponente kreirati koristeći funkciju

`function[boja]=komponenta(slika,RGB)`

`function [boja] = komponenta(slika, RGB)`

`boja=slika;`

`vel_u_pikselima = size(slika);`

`sirina=vel_u_pikselima(1);`

`visina=vel_u_pikselima(2);`

`if RGB == 'R'`

`for i=1:1:sirina`

`for j=1:1:visina`

`boja(i,j,2)=0;`

`boja(i,j,3)=0;`

`end`

`end`

`end`

`if RGB == 'G'`

`for i=1:1:sirina`

`for j=1:1:visina`

`boja(i,j,1)=0;`

`boja(i,j,3)=0;`

`end`

`end`

`end`

`if RGB == 'B'`

`for i=1:1:sirina`

`for j=1:1:visina`

`boja(i,j,1)=0;`

`boja(i,j,2)=0;`

`end`

`end`

`end`

`slika=imread('buket.jpg');`

`crvena=komponenta(slika,'R');`

`zelena=komponenta(slika,'G');`

`plava=komponenta(slika,'B');`

`figure(1); image(crvena);`

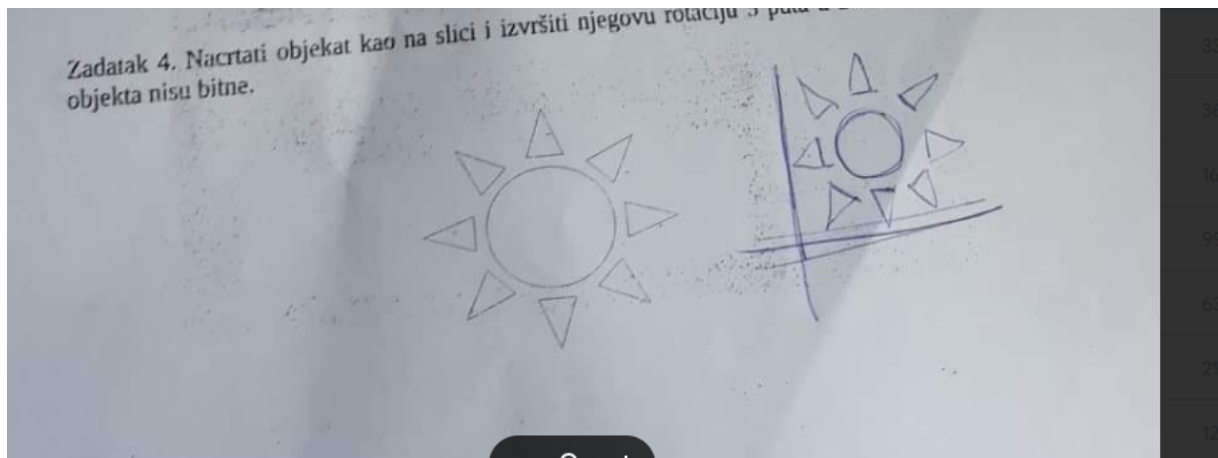
`figure(2); image(zelena);`

`figure(3); image(plava);`

```

imwrite(crvena, 'Crvena.jpg');
imwrite(zelena, 'Zelena.jpg');
imwrite(plava, 'Plava.jpg');

```



```

% Sunce
t=0:pi/50:2*pi;
x=cos(t);
y=sin(t);
krug=patch(x,y,'y');
axis([-2 2 -2 2]);

x1=[-0.225 0.225 0];
y1=[1.2 1.2 1.5];
trokut1=patch(x1,y1,'y');

x2=[-0.225 0.225 0];
y2=[-1.2 -1.2 -1.5];
trokut2=patch(x2,y2,'y');

x3=[1.2 1.2 1.5];
y3=[0.225 -0.225 0];
trokut3=patch(x3,y3,'y');

x4=[-1.2 -1.2 -1.5];
y4=[0.225 -0.225 0];
trokut4=patch(x4,y4,'y');

x5=[0.90 1 0.6];
y5=[0.65 1.1 1];
trokut5=patch(x5,y5,'y');

x6=[0.90 1 0.6];
y6=[-0.65 -1.1 -1];
trokut6=patch(x6,y6,'y');

x7=[-0.9 -1 -0.6];
y7=[0.65 1.1 1];

```

```

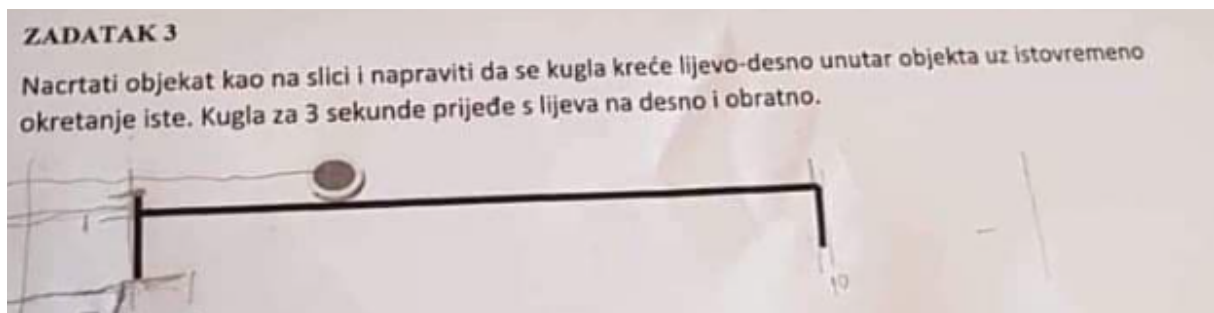
trokut7=patch(x7,y7,'y');

x8=[-0.9 -1 -0.6];
y8=[-0.65 -1.1 -1];
trokut8=patch(x8,y8,'y');

sunce=[krug trokut1 trokut2 trokut3 trokut4 trokut5 trokut6 trokut7
trokut8];

for i=1:60
    rotate(sunce, [0 0 1], 30, [0 0 0]);
    pause(2/60);
end

```



```

%3. Nacrtati objekat ako na slici i napraviti da se kugla kreće lijevo-
desno unutar objekta uz istovremeno okretanje itse. Kugla za 3s prijedje s
lijeva na desno i obratno (kugla lijevo-desno)
x1 = 0;
y1 = -0.5:0.001:0.1;
x2 = 4;
y2 = -0.5:0.001:0;
y3 = 0;
x3 = 0:0.01:4;

plot(x1*ones(size(y1)), y1, x2*ones(size(y2)), y2, x3, y3*ones(size(x3)),
'LineWidth', 4)
hold on

t = (0:1/720:1)' * 2 * pi;
x = 0.1 + 0.1 * sin(t);
y = 0.1 + 0.1 * cos(t);
h = fill(x, y, 'b');
axis([-0.5 4.5 -0.5 2])

xr = x;
yr = y;
a = 0; b = 0; c = 0;

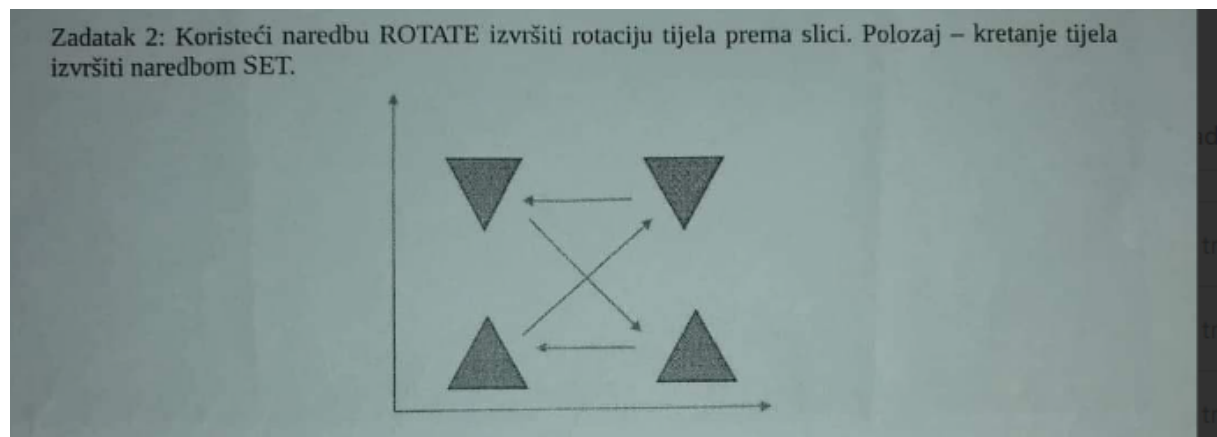
% Kretanje lopte s lijeva na desno i obrnuto
for i = 1:80
    xr = xr + (3.8 / 80); % Pomicanje po x osi za 3.8 u 80 koraka
    a = a + (3.8 / 80); % Pomicanje centra rotacije
    pause(3 / 160) % Pauza od 3/160 sekundi između koraka
    set(h, 'xdata', xr, 'ydata', yr)
    rotate(h, [0 0 1], 10, [a b 0]) % Rotacija oko z osi
end

```

```

for i = 1:80
    xr = xr - (3.8 / 80); % Pomicanje u suprotnom smjeru za istu
    udaljenost
    a = a - (3.8 / 80); % Pomicanje centra rotacije
    pause(3 / 160) % Pauza između koraka
    set(h, 'xdata', xr, 'ydata', yr)
    rotate(h, [0 0 1], 10, [a b 0]) % Rotacija oko z osi
end

```



% 2. Koristeći naredbu ROTATE izvršiti rotaciju tijela prema slici. Položaj – kretanje tijela izvršiti naredbom SET (trokut)

```

close all
clc
x1=[6 8 7 ]
y1=[3 3 2]
trougao=patch(x1,y1,'r')
x11=x1
y11=y1
axis([3 8 -1 3])
for i=1:30
    x11=x11-0.1
    y11=y11;
    set(trougao,'x',x11,'y',y11)
    pause(3/30)
end

a=x11+1
b=y11-0.5
for i=1:30
    x11=x11+0.1
    y11=y11-0.1;
    set(trougao,'x',x11,'y',y11)
    pause(3/30)
end

for i=1:30
    rotate(trougao,[0 0 1],6,[7,-0.5,0])
    pause (0.1)
end

x11=[6 8 7]

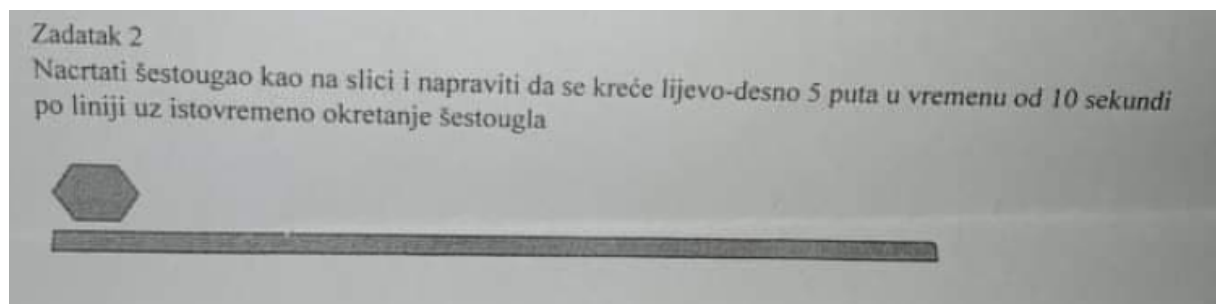
```

```

y11=[-1 -1 0]
for i=1:30
    x11=x11-0.1
    y11=y11;
    set(trougao,'x',x11,'y',y11)
    pause(2/30)
end
for i=1:30
    x11=x11+0.1
    y11=y11+0.1;
    set(trougao,'x',x11,'y',y11)
    pause(2/30)
end

for i=1:30
    rotate(trougao,[0 0 1],6,[7,2.5,0])
    pause (1/30)
end

```



%2. Nacrtati sestougao kao na slici i napraviti da se kreće lijevo-desno 5 puta u vrmeenu od 10s po liniji uz istovremeno okretanje sestougla

```

clc
clear all
close all
x=[0 10 10 0]
y=[0 0 0.8 0.8]
patch(x,y,'r')
hold on

t=(1/12:1/6:1) '*2*pi
x1=sin(t)+1
y1=cos(t)+2
h=fill(x1,y1,'y');
axis([-2 12 -12 12])
grid on
x11=x1;
y22=y1;
cx=1;

for k=1:5
    for i=1:30
        zdir=[0 0 1];
        center=[cx 2 0]

        set(h,'x',x11,'y',y22)
        rotate(h,zdir,i*30,center)
        cx=cx+0.27444;
    end
end

```

```

x11=x11+0.27444;
y22=y22;

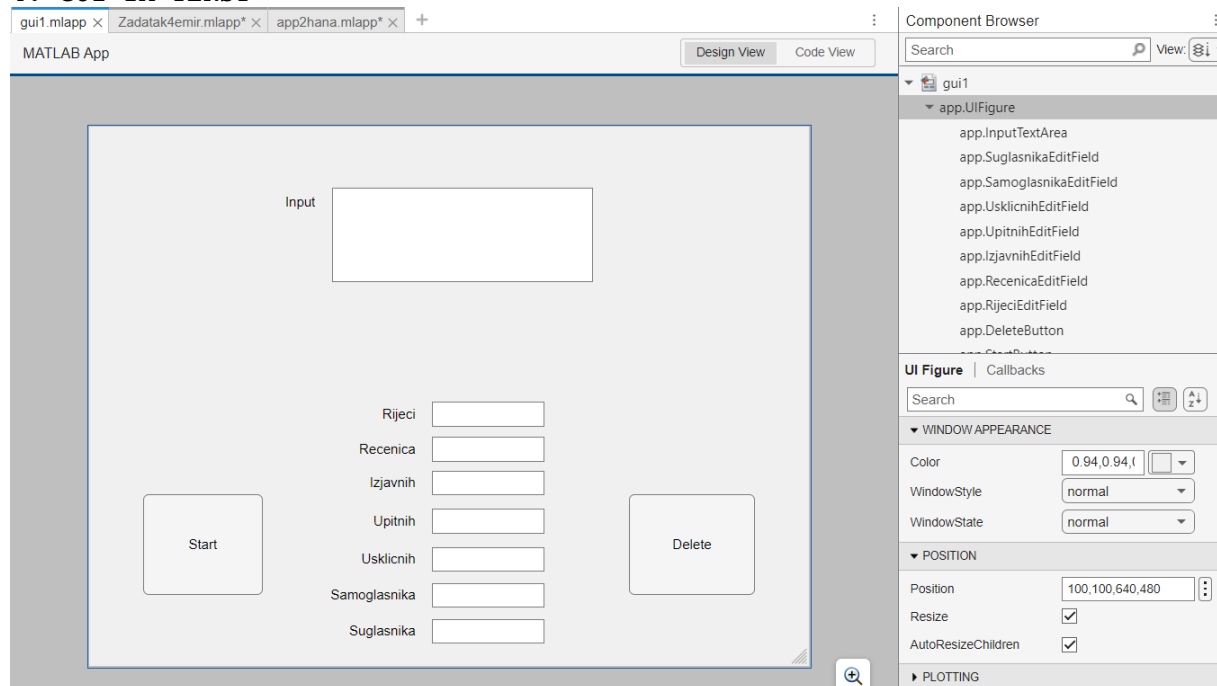
pause(0.025)
end

for j=1:30
    zdir=[0 0 1];
    cx=cx;
    center=[cx 2 0];

    set(h,'x',x11,'y',y22)
    rotate(h,zdir,j*30,center)
    x11=x11-0.27444;
    y22=y22;
    cx=cx-0.27444;
    pause(0.025)
end
end

```

4. GUI ZA TEKST



```

function StartButtonPushed(app, event)
    % Get input text
    text = app.InputTextArea.Value;
    text = string(text);
    % Count words

    words = strsplit(text);
    numRijeci = length(words);

    % Count exclamation sentences
    numUsklicnih = numel(strfind(text, '!'));
    numIzjavnih = numel(strfind(text, '.'));

```

```

numUpitnih = numel(strfind(text, '?'));
numRecenica = numUsklicnih + numIzjavnih + numUpitnih;
% Count vowels
textlower = lower(text);
numa = numel(strfind(textlower, 'a'));
nume = numel(strfind(textlower, 'e'));
numi = numel(strfind(textlower, 'i'));
numo = numel(strfind(textlower, 'o'));
numu = numel(strfind(textlower, 'u'));

numSamoglasnika = numa+nume+numi+numo+numu;

% Count non-vowels
numSuglasnika = sum(isletter(text)) - numSamoglasnika; % Count
all letters minus vowels

% Update fields
app.RijeciEditField.Value = num2str(numRijeci);
app.RecenicaEditField.Value = num2str(numRecenica);
app.UsklicnihEditField.Value = num2str(numUsklicnih);
app.IzjavnihEditField.Value = num2str(numIzjavnih);
app.UpitnihEditField.Value = num2str(numUpitnih);
app.SamoglasnikaEditField.Value = num2str(numSamoglasnika);
app.SuglasnikaEditField.Value = num2str(numSuglasnika);

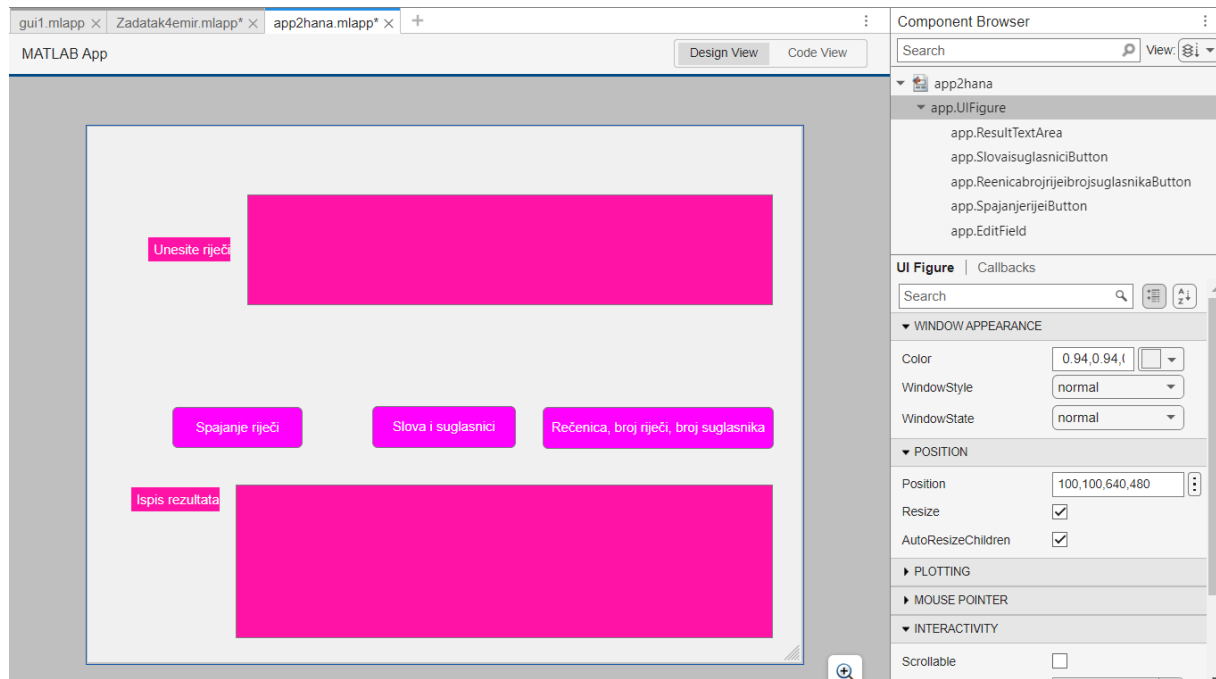
end

% Button pushed function: DeleteButton
function DeleteButtonPushed(app, event)
    app.InputTextArea.Value = '';

    % Clear the results fields
    app.RijeciEditField.Value = num2str(0);
    app.RecenicaEditField.Value = num2str(0);
    app.UsklicnihEditField.Value = num2str(0);
    app.IzjavnihEditField.Value = num2str(0);
    app.UpitnihEditField.Value = num2str(0);
    app.SamoglasnikaEditField.Value = num2str(0);
    app.SuglasnikaEditField.Value = num2str(0);

```

GUI HANA - unos rijeci a)spajanje rijeci u recenicu, koliko ima slova u recenici i suglasnika, ispisati recenicu i broj slova u svakoj rijeci i broj samoglasnika (npr. u rece nici ima 5 slova, od toga 3 samoglasnika (i navesti samoglasnike))



```
% Button pushed function: SpajanjerijeButton
function SpajanjerijeButtonPushed(app, event)
    sentence = app.EditField.Value;
    app.ResultTextArea.Value = sentence;

% Button pushed function: SlovasuglasniciButton
function SlovasuglasniciButtonPushed(app, event)
    sentence = app.EditField.Value;
    vowels = 'aeiouAEIOU';
    numVowels = sum(ismember(sentence, vowels));
    numNonVowels = sum(~ismember(sentence, vowels) &
isletter(sentence));
    app.ResultTextArea.Value = sprintf('Broj slova: %d, Broj
suglasnika: %d', numVowels + numNonVowels, numNonVowels);

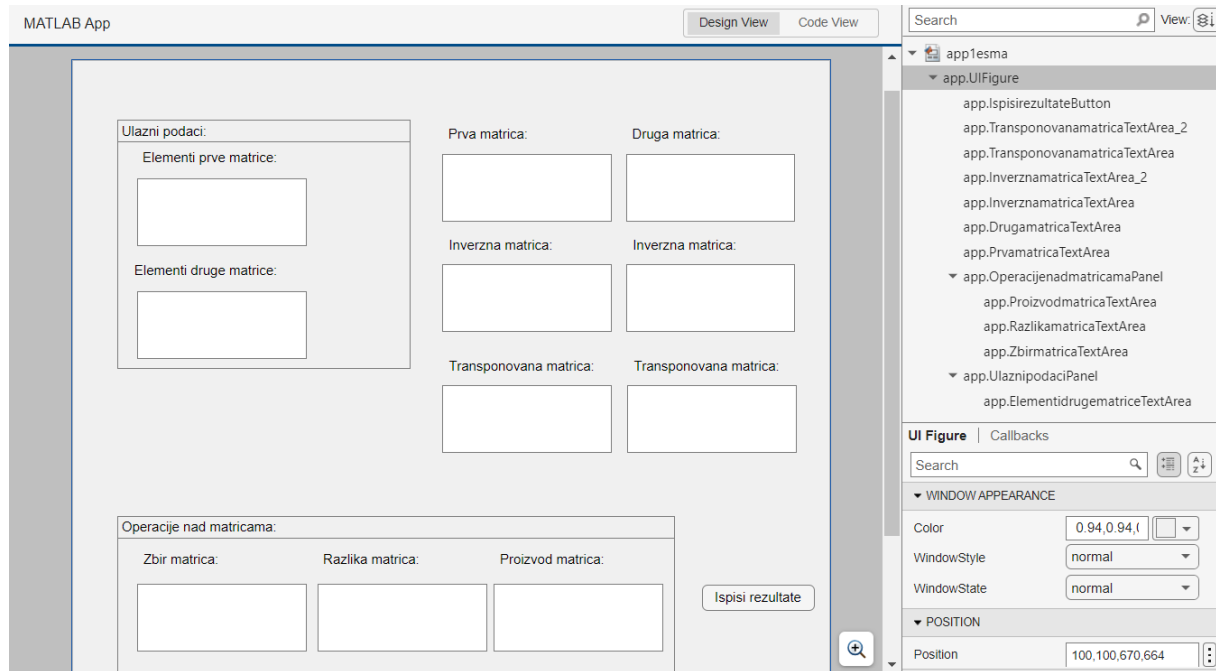
% Button pushed function: ReenicabrojrijeibrojsuglasnikaButton
function ReenicabrojrijeibrojsuglasnikaButtonPushed(app, event)
    sentence = app.EditField.Value;
    words = strsplit(sentence);
    result = '';
    vowels = 'aeiouAEIOU';

    for i = 1:length(words)
        word = words{i};
        numVowels = sum(ismember(word, vowels));
        result = [result, sprintf('Riječ: %s, Broj slova: %d, Broj
samoglasnika: %d, Samoglasnici: %s\n', ...
            word, length(word), numVowels, word(ismember(word,
vowels)))];
    end
```



```
app.ResultTextArea.Value = result;
```

GUI ESMA – matrice



```
% Size changed function: UlaznipodaciPanel
function UlaznipodaciPanelSizeChanged(app, event)
    position = app.UlaznipodaciPanel.Position;

end

% Value changed function: ElementiprvematriceTextArea
function ElementiprvematriceTextAreaValueChanged(app, event)
    value = app.ElementiprvematriceTextArea.Value;

end

% Button pushed function: IspisirezultateButton
function IspisirezultateButtonPushed(app, event)
    mat1 = str2num(char(app.ElementiprvematriceTextArea.Value));
    mat2 = str2num(char(app.ElementidrugematriceTextArea.Value));
    prikaz matrica
    app.PrvamatricaTextArea.Value = mat2str(mat1);
    app.DrugamatricaTextArea.Value = mat2str(mat2);
    % Racunanje i prikaz inverznih matrica
    if det(mat1) ~= 0
        app.InverznamatricaTextArea.Value = mat2str(inv(mat1));
    else
        app.InverznamatricaTextArea.Value = 'N/A';
    end
end
```

```

if det(mat2) ~= 0
    app.InverznamatricaTextArea_2.Value = mat2str(inv(mat2));
else
    app.InverznamatricaTextArea_2.Value = 'N/A';
end
% Prikaz transponovanih matrica
app.TransponovanamatricaTextArea.Value = mat2str(mat1');
app.TransponovanamatricaTextArea_2.Value = mat2str(mat2');
% Operacije sa matricama
if isequal(size(mat1), size(mat2))
    app.ZbirmatricaTextArea.Value = mat2str(mat1 + mat2);
    app.RazlikamatricaTextArea.Value = mat2str(mat1 - mat2);
else
    app.ZbirmatricaTextArea.Value = 'N/A';
    app.RazlikamatricaTextArea.Value = 'N/A';
end

if size(mat1, 2) == size(mat2, 1)
    app.ProizvodmatricaTextArea.Value = mat2str(mat1 * mat2);
else
    app.ProizvodmatricaTextArea.Value = 'N/A';
end

```