

1. DRUGI zadatak rok

```
clc

clear all

o = linspace(-2.5,2.5,300);

[x,y] = meshgrid(o,o);

z1 = (-21 - x - 4.*y)/6;

z2 = (21 - x - 4.*y)/6;

Z1 = zeros(1,length(x));

for i = 1:length(x)

    for j = 1:length(y)

        if (21 - x(i,j)^2 - 2*(y(i,j)^2)) > 0

            Z1(i,j) = sqrt((21 - x(i,j).^2 - 2*(y(i,j).^2))/3);

        else

            Z1(i,j) = NaN;

        end

    end

end

Z2 = zeros(1,length(x));

for i = 1:length(x)

    for j = 1:length(y)

        if (21 - x(i,j)^2 - 2*(y(i,j).^2)) > 0

            Z2(i,j) = -sqrt((21 - x(i,j).^2 - 2*(y(i,j).^2))/3);

        else

            Z2(i,j) = NaN;

        end

    end

end
```

```
end  
end  
  
figure;  
axis equal;  
hold on;  
grid on;  
view(3);  
mesh(x,y,z1);  
mesh(x,y,z2);  
mesh(x,y,Z1);  
mesh(x,y,Z2);  
  
title('3D graf traženih funkcija');
```

2. GUI KOD ZA UNESI RIJECI

```
uneseni = app.TextArea.Value;           % čitanje iz TextArea  
    uneseni = strjoin(uneseni, ' ');      % spajanje više linija u string  
    app.Label.Text = strtrim([app.Label.Text ' ' uneseni]);
```

2. ANIMACIJA SA ROKA

```
figure;  
hold on;  
axis equal;  
axis([0 13, -1, 7]);  
  
%vrata  
vx = [0 2 2 0];  
vy = [0 0 1 1];  
  
v = fill(vx,vy,'b','EdgeColor', [1,1,1], 'Linewidth', 4);  
  
%prozor  
  
px = [1 2.5 2.5 1];  
py = [2 2 3 3];  
  
p = fill(px,py,'y','EdgeColor', [1,1,1], 'Linewidth', 4);  
  
%zid  
  
zx = [0 4 4 0];  
zy = [4 4 8 8];  
  
z = fill(zx,zy,'m','EdgeColor', [1,1,1], 'Linewidth', 4);
```

```
%krov
```

```
kx = [3 5 5];
```

```
ky = [2 -1 5];
```

```
k = fill(kx,ky,'r','EdgeColor', [1,1,1], 'Linewidth', 4);
```

```
f = [];
```

```
%zid
```

```
for i = 1:100
```

```
    delete(z);
```

```
    z = fill(zx + i*8/100,zy - i*4/100,'m','EdgeColor', [1,1,1], 'Linewidth', 4);
```

```
    f = [f,getframe];
```

```
    pause(1/100);
```

```
end
```

```
%prozor
```

```
for i = 1:100
```

```
    delete(p);
```

```
    p = fill(px + i*9/100,py , 'y','EdgeColor', [1,1,1], 'Linewidth', 4);
```

```
    f = [f,getframe];
```

```
    pause(1/100);
```

```
end
```

```
%krov rotacija
```

```
zdir = [0 0 -1];
```

```
centar = [4 2 0];
```

```
for i = 1:100
```

```
    rotate(k,zdir,90/100, centar);
```

```
    pause(1/100);
```

```
    f = [f,getframe];
```

```
end
```

```
kx = k.XData;
```

```
ky = k.YData;
```

```
for i = 1:100
```

```
    delete(k);
```

```
    k = fill(kx + i*6/100,ky +i*3/100 ,'r', 'EdgeColor', [1,1,1], 'Linewidth', 4);
```

```
    pause(1/100);
```

```
    f = [f,getframe];
```

```
end
```

```
%vrata totacija
```

```
zdir = [0 0 -1];
```

```
centar = [1 0.5 0];
```

```
for i = 1:100
```

```
    rotate(v,zdir,90/100, centar);
```

```
    pause(1/100);
```

```
    f = [f,getframe];  
end
```

```
vx = v.XData;  
vy = v.YData;
```

```
for i = 1:100  
    delete(v);  
    v = fill(vx + i*(9.5-1.5)/100,vy + i*(2-1.5)/100,'b','EdgeColor',[1,1,1], 'Linewidth', 4);  
    pause(1/100);  
    f = [f,getframe];  
end
```

```
if ~exist('movie')  
    mkdir movie  
end
```

```
v = VideoWriter(fullfile('movie', 'video1'));  
open(v);  
writeVideo(v,f);  
close(v);
```

3. OPERACIJE SA MATRICAMA

```
disp("Unos matrice A");
```

```
A = input('Unesite matricu A: ');
```

```
disp("Unos matrice B");
```

```
B = input('Unesite matricu B: ');
```

```
disp("Unos matrice C");
```

```
C = input('Unesite matricu C: ');
```

```
disp('Dostupne operacije:');
```

```
disp("1- Sabiranje");
```

```
disp("2- Oduzimanje");
```

```
disp("3- Množenje");
```

```
opcija = input('Izaberi opciju (1 / 2 / 3): ');
```

```
disp("Matrice");
```

```
disp('1- A');
```

```
disp('2- B');
```

```
disp('3- C');
```

```
prva = input('Izaberi prvu matricu: ');
```

```
druga = input('Izaberi drugu matricu: ');
```

switch prva

case 1

M1 = A;

case 2

M1 = B;

case 3

M1 = C;

otherwise

disp('Nepoznata opcija!');

return;

end

switch druga

case 1

M2 = A;

case 2

M2 = B;

case 3


```
M2 = C;
```

```
otherwise
```

```
    disp('Nepoznata opcija!');
```

```
    return;
```

```
end
```

```
switch opcija
```

```
case 1
```

```
    if size(M1) == size(M2)
```

```
        rez = M1+M2;
```

```
        disp('Rezultat zbrajanja: ');
```

```
        disp(rez);
```

```
    else
```

```
        disp('Sabiranje nije moguće matrice nisu istih dimenzija');
```

```
    end
```

```
case 2
```

```
    if size(M1) == size(M2)
```

```
        rez = M1-M2;
```

```
        disp('Rezultat oduzimanja: ');
```

```
        disp(rez);
```

```
    else
```

```

        disp('Nemoguce oduzimanje matrice nisu iste duzine');
    end

    case 3
        if size(M1, 2) == size(M2, 1)
            rez = M1*M2;
            disp(rez);

        else
            disp('Greska');
        end
    otherwise
        disp('Nepoznata operacija!');
    end
end

```

4. FUNKCIJA SA T OMEGA I FI

```
omega = 2 * pi;
```

```
fi = pi/4;
```

```
T_vrijednosti = [1, 2, 5];
```

```
t = 0:0.01:10;
```

```
figure;
```

```

for i = 1:3
    T = T_vrijednosti(i);
    f = exp(-t / T).*sin(omega * t - fi);

    subplot(3,1,i);
    plot(t, f);
    grid on;
    xlabel('t');
    ylabel('f(t)');
    title('T');
end

```

5. FREKVENCije

```

Fs = 3000;
t1 = 0:1/Fs:1;
t2 = 0:1/Fs:2;

ton = sin(2*pi*100*t1);

sum_signal = rand(1, length(t2));

sum_jedna_sekunda = sum_signal(1:length(t1));

```

```
spojeni = ton + sum_jedna_sekunda;
```

```
x = 0.5 * sin(2*pi*t1);
```

```
finalni = spojeni .* x;
```

```
figure;
```

```
subplot(4, 1, 1);
```

```
plot(t1, ton);
```

```
grid on;
```

```
subplot(4, 1, 2);
```

```
plot(t1, sum_jedna_sekunda);
```

```
grid on;
```

```
subplot(4, 1, 3);
```

```
plot(t1, spojeni);
```

```
grid on;
```

```
subplot(4, 1, 4);
```

```
plot(t1, finalni);
```

```
grid on;
```

6. JAVA PRIKAZ SLIKE

% 1. Učitaj sliku

```
img = imread('Macka.jpg'); % zamijeni sa stvarnim imenom slike
```

% 2. Konvertuj u Java BufferedImage

```
javaImg = im2java(img);
```

% 3. Prikaz slike kao Java objekat

```
import javax.swing.*
```

```
import java.awt.*
```

```
icon = ImageIcon(javaImg);
```

```
label = JLabel(icon);
```

```
frame = JFrame('Java Prikaz Slike');
```

```
frame.getContentPane().add(label);
```

```
frame.setSize(size(img, 2), size(img, 1)); % širina, visina
```

```
frame.setVisible(true);
```

7. STRUKTURA IME PREZIME I BROJ INDEXA

```
clear all;
```

```
clc
```

```
studenti(3) = struct('ime', '', 'prezime', '', 'brojIndexa', 0);
```

```
for i = 1:3
```

```
    studenti(i).ime = input('Ime: ', 's');
```

```
    studenti(i).prezime = input('Preizime', 's');
```

```
    studenti(i).brojIndexa = input('Index: ');
```

```
end
```

```
Pretraga = input('Unesi ime studenta za pretragu: ', 's');
```

```
pronadjen = false;
```

```
for i = 1:3
```

```
    if strcmp(studenti(i).ime, Pretraga)
```

```
        disp('Trazeni student: ');
```

```
        disp(studenti(i).ime);
```

```
        disp(studenti(i).prezime);
```

```
        disp(studenti(i).brojIndexa);
```

```
        pronadjen = true;
```

```
        break;
```

```
    end
```

```
end
```

```

if pronadjen == false
    disp('Trazeni student se ne nalazi u strukturi!');

end

```

8. OKRETANJE KUGLE UNUTAR OBJEKTA

```

% Parametri

r = 0.5;      % poluprečnik kugle
L = 10;      % dužina horizontalnog puta
h = 3;      % visina zidova
fps = 60;    % broj frejmova u sekundi
v = 2;      % brzina kretanja kugle (jedinica/sek)

% Osnovna figura

figure
axis equal
axis([-2 L+2 -2 h+2])
hold on

plot([0 0 L L], [0 h h 0], 'k', 'LineWidth', 4); % U-šina

% Kugla (krug + oznaka rotacije)

theta = linspace(0, 2*pi, 100);

```

```
xk = r * cos(theta);
```

```
yk = r * sin(theta);
```

```
% Rotaciona oznaka (linija na kugli)
```

```
oznaka_x = [0 r];
```

```
oznaka_y = [0 0];
```

```
% Prva pozicija
```

```
kugla = fill(xk + 0, yk + r, 'r'); % crvena kugla
```

```
oznaka = plot(oznaka_x, oznaka_y + r, 'k', 'LineWidth', 2); % crna linija
```

```
% Petlja animacije
```

```
x = 0;
```

```
dx = v / fps;
```

```
smer = 1;
```

```
ukupno = 2 * fps * L;
```

```
for k = 1:ukupno
```

```
    x = x + dx * smer;
```

```
    if x >= L || x <= 0
```

```
        smer = -smer; % promjena pravca kad dotakne zid
```

```
    end
```

```
% Ugao rotacije = predjeni put / poluprečnik
```

```
angle = -x / r;
```



```

% Rotacija cijele kugle

x_rot = xk * cos(angle) - yk * sin(angle);
y_rot = xk * sin(angle) + yk * cos(angle);

set(kugla, 'XData', x_rot + x, 'YData', y_rot + r);

% Rotiraj oznaku na kugli (vizuelna rotacija)
oznaka_rot_x = oznaka_x * cos(angle) - oznaka_y * sin(angle);
oznaka_rot_y = oznaka_x * sin(angle) + oznaka_y * cos(angle);
set(oznaka, 'XData', oznaka_rot_x + x, 'YData', oznaka_rot_y + r);

pause(1/fps);
end

```

9. ZADATAK SA OCJENAMA I STRUKTURAMA

```

ocjena6 = 0;
ocjena7 = 0;
ocjena8 = 0;
ocjena9 = 0;
ocjena10 = 0;

studenti(10) = struct('ime', '', 'prezime', '', 'ocjena', 0);

```

```
for i = 1:10
```

```
    disp('Unesi podatke za studenta: ');
```

```
    studenti(i).ime = input('Ime: ', 's');
```

```
    studenti(i).prezime = input('Prezime', 's');
```

```
    ocjena = input('Ocjena(6-10): ');
```

```
    if ocjena < 6 || ocjena > 10
```

```
        disp('Unesena pogresna ocjena!');
```

```
    else
```

```
        studenti(i).ocjena = ocjena;
```

```
    end
```

```
switch ocjena
```

```
    case 6
```

```
        ocjena6 = ocjena6 + 1;
```

```
    case 7
```

```
        ocjena7 = ocjena7 + 1;
```

```
    case 8
```

```
        ocjena8 = ocjena8 + 1;
```

case 9

ocjena9 = ocjena9 + 1;

case 10

ocjena10 = ocjena10 + 1;

end

end

disp('Broj studenata sa ocjenom 6');

disp(ocjena6);

disp('Broj studenata sa ocjenom 7');

disp(ocjena7);

disp('Broj studenata sa ocjenom 8');

disp(ocjena8);

disp('Broj studenata sa ocjenom 9');

disp(ocjena9);

disp('Broj studenata sa ocjenom 10');

disp(ocjena10);

10. PIE GRAF I PIE 3

ocjene = [5, 4, 3, 2, 1];

```
etikete = {'Odlican', 'Vrlodobar', 'Dobar', 'Dovoljan', 'Los'};
```

```
izdvojeno = [1 0 0 0 0];
```

```
subplot(121);
```

```
pie(ocjene, izdvojeno, etikete);
```

```
subplot(122);
```

```
pie3(ocjene, izdvojeno, etikete);
```

11. SLIKA

MAIN

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

```
mkdir('Crvena');
```

```
mkdir('Plava');
```

```
mkdir('Zelena');
```

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

```
imwrite(R, 'Crvena/crvena.jpg');
```

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

```
imwrite(B, 'Plava/plava.jpg');
```

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

```
imwrite(G, 'Zelena/zelena.jpg');
```

```
mkdir('Siva');
```

```
slikasiva = rgb2gray(slika);
```

```
imwrite(slikasiva, 'Siva/siva.jpg');
```

KOMPONENTA

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

```
boja = zeros(size(slika), 'uint8');
```

```
switch RGB
```

```
case 'R'
```

```
    boja(:,:,1) = slika(:,:,1);
```

```
case 'B'
```

```
    boja(:,:,2) = slika(:,:,2);
```

```
case 'G'
```

```
    boja(:,:,3) = slika(:,:,3);
```

```
end
```