

## ZADATAK 1 - 12 brojeva

```
niz = [];  
  
brojac = 0;  
while brojac < 12  
  
    broj = input("Unesi broj");  
    if ismember(broj,niz)  
        disp("upisali ste vec upisani broj, molim Vas unesite drugi broj")  
  
    else  
        niz = [niz broj];  
        brojac = brojac+1;  
    end  
end  
  
drugiNiz = niz;  
  
sortiraniNiz = sort(drugiNiz, "descend");  
  
matrica = reshape(sortiraniNiz, 3, 4)';  
  
disp("Matrica je : ");  
disp(matrica);
```

## %% ZADATAK 2 - Riješiti sistem jednačina

```
A = [3 5 -3 8; 3 -5 -3 8; 1 8 9 -2; 3 0 0 8];  
B = [9; 9; 16; 13];
```

$X = \text{inv}(A) \cdot B$

$Y = A \setminus B$

## **%% ZADATAK 3 - Nacrtati graf funkcije**

% Definisanje mreže tačaka

[x, y] = meshgrid(-2:0.25:2);

% Izračunavanje funkcije

z = x .\* exp(-x.^2 - y.^2);

% Postavljanje figure za subplots

figure;

% Prikazivanje contour3 grafika u subplotu

subplot(2, 3, 1);

contour3(x, y, z, 30);

colormap hsv;

xlabel('x osa');

ylabel('y osa');

zlabel('z osa');

colorbar;

title('Contour3 Grafik');

grid off;

% Prikazivanje surf grafika u subplotu

subplot(2, 3, 2);

surf(x, y, z);

colormap hsv;

xlabel('x osa');

ylabel('y osa');

zlabel('z osa');

colorbar;

```
title('Surf Grafik');
```

```
grid on;
```

```
% Prikazivanje mesh grafika u subplotu
```

```
subplot(2, 3, [3,6]);
```

```
mesh(x, y, z);
```

```
colormap hsv;
```

```
xlabel('x osa');
```

```
ylabel('y osa');
```

```
zlabel('z osa');
```

```
colorbar;
```

```
title('Mesh Grafik');
```

```
grid on;
```

```
% Prikazivanje ezsurf grafika u subplotu
```

```
subplot(2, 3, 4);
```

```
ezsurf('x.*exp(-x.^2-y.^2)', [-2, 2, -2, 2]);
```

```
colormap hsv;
```

```
xlabel('x osa');
```

```
ylabel('y osa');
```

```
zlabel('z osa');
```

```
colorbar;
```

```
title('EZSurf Grafik');
```

```
grid on;
```

```
% Prikazivanje ezmesh grafika u subplotu
```

```
subplot(2, 3, 5);
```

```
ezmesh('x.*exp(-x.^2-y.^2)', [-2, 2, -2, 2]);
```

```
colormap hsv;
```

```
xlabel('x osa');
```

```
ylabel('y osa');
```

```
zlabel('z osa');
```

```
colorbar;
```

```
title('EZMesh Grafik');
```

```
grid on;
```

## **%% ZADATAK 4 - abeceda**

```
red = input('Unesite broj redova matrice: ');
```

```
kolona = input('Unesite broj kolona matrice: ');
```

```
% Inicijalizacija matrice
```

```
matrica = zeros(red, kolona);
```

```
% Unos elemenata matrice
```

```
for i = 1:red
```

```
    for j = 1:kolona
```

```
        matrica(i, j) = input('Unesite element matrice: ');
```

```
    end
```

```
end
```

```
% Pravljenje niza od elemenata matrice
```

```
niz = matrica';
```

```
niz = niz(:)';
```

```
% Definisanje abecede
```

```
abeceda = 'abcdefghijklmnopqrstuvwxyz';
```

```
% Inicijalizacija dodatnog niza za rezultate
```

```
dodatniNiz = '';
```

```
% Zamena brojeva slovima
```

```
for i = 1:length(niz)
```

```
    if niz(i) <= 26
```

```
        dodatniNiz(i) = abeceda(niz(i));
```

```
    else
```

```
        dodatniNiz(i) = upper(abeceda(mod(niz(i), 26)));
```

```
end
```

```
end
```

```
% Ispis rezultata
```

```
disp('Niz je:');
```

```
disp(dodatniNiz);
```

## **%% ZADATAK 5 - graf**

```
x = 0:0.1:10*pi;
```

```
y = 0:0.1:10*pi;
```

```
z = (cos(x)).^2 + sin(y).^3;
```

```
s = 50;
```

```
% Kreiranje subplotova
```

```
figure;
```

```
% Meshgrid
```

```
[X, Y] = meshgrid(x, y);
```

```
% Pretvaranje vektora z u matricu Z
```

```
Z = (cos(X)).^2 + sin(Y).^3;
```

```
% Surf plot
```

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

```
surf(X, Y, Z);
```

```
xlabel('x osa');
```

```
ylabel('y osa');
```

```
zlabel('z osa');
```

```
title('Surf plot');
```

```
% Mesh plot
```

```
subplot(2, 2, 2);
```

```
mesh(X, Y, Z);  
xlabel('x osa');  
ylabel('y osa');  
zlabel('z osa');  
title('Mesh plot');
```

```
% Contour3 plot  
subplot(2, 2, 3);  
contour3(X, Y, Z, 30);  
xlabel('x osa');  
ylabel('y osa');  
zlabel('z osa');  
title('Contour3 plot');
```

```
% Scatter3 plot  
subplot(2, 2, 4);  
scatter3(x, y, z, s);  
xlabel('x osa');  
ylabel('y osa');  
zlabel('z osa');  
title('Scatter3 plot');
```

## **%% ZADATAK 6 - brojevi telefona**

```
clc;  
clear all;  
n = input('unesite broj korisnika\n');  
for i=1:2  
    for j=1:n  
        if(i==1)  
            fprintf('Unesite ime %d-og korisnika: ',j);  
            Korisnici(j).ime = input('', 's');  
        end  
        if(i==2)
```

```

fprintf('Unesite broj telefona %d-og korisnika\n',j);
zb=0;
for brojac=1:9
    prosao = 0;
    while(prosao==0)
        xb = input('');
        if(xb<0 | xb>9)
            fprintf('pogresan unos, unesi tu cifru ponovo\n');
        end
        if(xb>=0 & xb<=9)
            prosao=1;
            zb = zb+xb;
        end
    end
    end
    Korisnici(j).broj = zb;
end
end
end
opcija = input('odaberi opciju\n');
fprintf('korisnik je %s i ima vrijednost %d', Korisnici(opcija).ime, Korisnici(opcija).broj);

```

## %% ZADATAK 7 - MATRICA

% Unos prve matrice

clear all

clc

r1 = input('Unesite broj redova prve matrice: ');

k1 = input('Unesite broj kolona prve matrice: ');

for i = 1:r1

for j = 1:k1

prvaMatrica(i,j) = input('Unesite elemente prve matrice: ');

end

end

% Unos druge matrice

r2 = input('Unesite broj redova druge matrice: ');a

k2 = input('Unesite broj kolona druge matrice: ');

for i = 1:r2

for j = 1:k2

drugaMatrica(i,j) = input('Unesite elemente druge matrice: ');

end

end

% Unos treće matrice

r3 = input('Unesite broj redova treće matrice: ');

k3 = input('Unesite broj kolona treće matrice: ');

for i = 1:r3

for j = 1:k3

trecaMatrica(i,j) = input('Unesite elemente treće matrice: ');

end

end



```
disp("Prva matrica koju ste unijeli: ");  
disp(prvaMatrica);
```

```
disp('Druga matrica koju ste unijeli: ');  
disp(drugaMatrica);
```

```
disp('Treća matrica koju ste unijeli: ');  
disp(trecaMatrica);
```

```
% Izbornik za operacije  
disp('Izaberite koje dvije matrice želite koristiti za operacije:');  
disp('1. Prva matrica i druga matrica');  
disp('2. Prva matrica i treća matrica');  
disp('3. Druga matrica i treća matrica');
```

```
izbor = input('Unesite broj izbora: ');
```

```
switch izbor
```

```
    case 1
```

```
        matrica1 = prvaMatrica;  
        matrica2 = drugaMatrica;
```

```
    case 2
```

```
        matrica1 = prvaMatrica;  
        matrica2 = trecaMatrica;
```

```
    case 3
```

```
        matrica1 = drugaMatrica;  
        matrica2 = trecaMatrica;
```

```
    otherwise
```

```
        disp('Pogrešan izbor.');
```

```
        return;
```

```
end
```

```
% Provjera za oduzimanje matrica
if isequal(size(matrica1), size(matrica2))
    disp('Matrice se mogu oduzimati. ');
    oduzimanjeMatrica = matrica1 - matrica2;
    disp("Rezultat oduzimanja matrica je:");
    disp(oduzimanjeMatrica);
else
    disp('Matrice se ne mogu oduzimati. ');
end
```

```
% Provjera za množenje matrica
if size(matrica1, 2) == size(matrica2, 1)
    disp('Matrice se mogu množiti. ');
    mnozenjeMatrica = matrica1 * matrica2;
    disp('Rezultat množenja je:');
    disp(mnozenjeMatrica);
else
    disp('Matrice se ne mogu množiti. ');
end
```

```
% Dodavanje operacija sabiranja i dijeljenja
if isequal(size(matrica1), size(matrica2))
    disp('Matrice se mogu sabirati. ');
    sabiranjeMatrica = matrica1 + matrica2;
    disp("Rezultat sabiranja matrica je:");
    disp(sabiranjeMatrica);
else
    disp('Matrice se ne mogu sabirati. ');
end
```

```
if size(matrica1, 2) == size(matrica2, 1)
    disp('Matrice se mogu dijeliti. ');
    dijeljenjeMatrica = matrica1 / matrica2;
```

```
disp('Rezultat dijeljenja je:');  
disp(dijeljenjeMatrica);  
else  
disp('Matrice se ne mogu dijeliti.');
```

end

## ZADATAK 8 - rand

```
clear all
```

```
clc
```

```
% Inicijalizacija broja studenata i odsjeka
```

```
brojStudenata = 10;
```

```
brojOdsjeka = 5;
```

```
% Generiranje prosječnih ocjena za 10 studenata za 5 odsjeka (ocjene između 6 i 10)
```

```
prosjecneOcjene = rand(brojStudenata, brojOdsjeka) * 4 + 6;
```

```
% Pronalaženje najboljih i najlošijih studenata za svaki odsjek
```

```
[najboljeOcjeneOdsjeka, najboljiIndeksiOdsjeka] = max(prosjecneOcjene);
```

```
[najlosijeOcjeneOdsjeka, najlosijiIndeksiOdsjeka] = min(prosjecneOcjene);
```

```
% Pronalaženje najboljeg i najlošijeg studenta među svim odsjecima
```

```
[najboljaOcjena, najboljiStudentIndeks] = max(prosjecneOcjene(:));
```

```
[najlosijaOcjena, najlosijiStudentIndeks] = min(prosjecneOcjene(:));
```

```
% Pretvaranje linearnog indeksa u indeks redova i kolona
```

```
[najboljiStudentRed, najboljiStudentKolona] = ind2sub(size(prosjecneOcjene), najboljiStudentIndeks);
```

```
[najlosijiStudentRed, najlosijiStudentKolona] = ind2sub(size(prosjecneOcjene), najlosijiStudentIndeks);
```

```
% Izračunavanje prosječne ocjene za svaki odsjek pojedinačno
```

```
prosjecnaOcjenaOdsjeka = mean(prosjecneOcjene);
```

```

% Izračunavanje prosječne ocjene za sve odsjeke
prosjecnaOcjenaSvihOdsjeka = mean(prosjecneOcjene, 2);

% Sortiranje studenata po odsjecima prema prosječnoj ocjeni od najveće do najmanje
[prosjecnaOcjenaSvihOdsjekaSortirano, indeksiSortirano] = sort(prosjecnaOcjenaSvihOdsjeka, 'descend');

% Ispis rezultata
disp('Najbolji studenti po odsjecima:');
disp(najboljeOcjeneOdsjeka);
disp('Najlošiji studenti po odsjecima:');
disp(najlosijeOcjeneOdsjeka);
fprintf('Najbolji student svih odsjeka je student %d na odsjeku %d sa prosječnom ocjenom %.2f\n',
najboljiStudentRed, najboljiStudentKolona, najboljaOcjena);
fprintf('Najlošiji student svih odsjeka je student %d na odsjeku %d sa prosječnom ocjenom %.2f\n',
najlosijiStudentRed, najlosijiStudentKolona, najlosijaOcjena);
disp('Prosječne ocjene po odsjecima:');
disp(prosjecnaOcjenaOdsjeka);
fprintf('Prosječna ocjena svih studenata je: %.2f\n', mean(prosjecnaOcjenaSvihOdsjeka));
disp('Sortirani studenti po odsjecima prema prosječnoj ocjeni (od najveće do najmanje):');
disp(prosjecnaOcjenaSvihOdsjekaSortirano);
disp('Indeksi sortiranih studenata po odsjecima:');
disp(indeksiSortirano);

```

## %% ZADATAK 9 - mjerne jedinice

```

% Korisnik unosi broj X u centimetrima
X = input('Unesite broj u centimetrima: ');

% Korisnik bira mjerne jedinice za pretvaranje
unit = input('Unesite mjerne jedinice (in, m, cm, mm, dm): ', 's');

% Pretvaranje unešenog broja u odabrane mjerne jedinice koristeći switch case strukturu
switch unit

```

```

case 'in'
    % Pretvaranje iz centimetara u inče
    convertedValue = X / 2.54;
    fprintf('%.2f cm = %.2f in\n', X, convertedValue);
case 'm'
    % Pretvaranje iz centimetara u metre
    convertedValue = X / 100;
    fprintf('%.2f cm = %.2f m\n', X, convertedValue);
case 'cm'
    % Pretvaranje iz centimetara u centimetre (nema promjene)
    convertedValue = X;
    fprintf('%.2f cm = %.2f cm\n', X, convertedValue);
case 'mm'
    % Pretvaranje iz centimetara u milimetre
    convertedValue = X * 10;
    fprintf('%.2f cm = %.2f mm\n', X, convertedValue);
case 'dm'
    % Pretvaranje iz centimetara u decimetre
    convertedValue = X / 10;
    fprintf('%.2f cm = %.2f dm\n', X, convertedValue);
otherwise
    disp('Nepoznata mjerna jedinica. Molimo pokušajte ponovo.');
```

end

```
%% ZADACI KOJI SE PONSVLJSJU
```

## %% ZADATAK 10 - kvadratna matrica

% Unos proizvoljnog niza sa tastature

```
niz = input('Unesite niz elemenata (npr. [1 2 3 4 5 6 7 8 9]): ');
```

% Odredjivanje duzine niza

```
n = length(niz);
```

% Odredjivanje dimenzije najveće kvadratne matrice

```
dim = floor(sqrt(n));
```

% Formiranje najveće kvadratne matrice

```
matrica = reshape(niz(1:dim^2), dim, dim);
```

% Racunanje inverzne matrice

```
inverznaMatrica = inv(matrica);
```

% Mnozenje matrice sa njenom inverznom matricom

```
rezultat = matrica * inverznaMatrica;
```

% Ispis ulaznog niza

```
fprintf('Ulazni niz: ');
```

```
disp(niz);
```

% Ispis kvadratne matrice

```
fprintf('Kvadratna matrica:\n');
```

```
disp(matrica);
```

% Ispis inverzne matrice

```
fprintf('Inverzna matrica:\n');
```

```
disp(inverznaMatrica);
```

```
% Ispis rezultata mnozenja
fprintf('Rezultat mnozenja matrice sa njenom inverznom matricom:\n');
disp(rezultat);
```

## **%% ZADATAK 11 - CRTANJE ELIPSOIDE**

```
% Postavke za prikaz
figure;
hold on;
axis equal;
xlabel('x');
ylabel('y');
zlabel('z');
grid on;

% Definicija elipsoida  $x^2 + 2y^2 + 3z^2 = 21$ 
[x, y, z] = ellipsoid(0, 0, 0, sqrt(21), sqrt(21/2), sqrt(21/3), 50);

% Crtanje elipsoida
surf(x, y, z, 'FaceAlpha', 0.5);
colormap jet;

% Tangentna ravan  $x + 4y + 6z = -21$ 
[X, Y] = meshgrid(-10:0.5:10, -10:0.5:10);
Z1 = (-21 - X - 4*Y) / 6;
Z2 = (21 - X - 4*Y) / 6;

% Crtanje tangentnih ravni
surf(X, Y, Z1, 'FaceColor', 'red', 'FaceAlpha', 0.5, 'EdgeColor', 'none');
surf(X, Y, Z2, 'FaceColor', 'blue', 'FaceAlpha', 0.5, 'EdgeColor', 'none');

% Postavke za bolji prikaz
xlim([-10 10]);
```

```

ylim([-10 10]);
zlim([-10 10]);

% Legenda za lakše razlikovanje
legend('Elipsoid  $x^2 + 2y^2 + 3z^2 = 21$ ', 'Tangenta  $x + 4y + 6z = -21$ ', 'Tangenta  $x + 4y + 6z = 21$ ');

hold off;

```

## %% Zadatak 12 - PARNI

```

% Unos prvog niza sa tastature
niz1 = input('Unesite prvi niz elemenata: ');

% Unos drugog niza sa tastature
niz2 = input('Unesite drugi niz elemenata: ');

% Pronalaženje broja parnih elemenata u prvom nizu
broj_parnih_niz1 = sum(mod(niz1, 2) == 0);

% Sabiranje svih elemenata ako je veći broj parnih elemenata u prvom nizu
if broj_parnih_niz1 > length(niz1) / 2
    rezultat = sum(niz1) + sum(niz2);
    disp('Izabrana operacija: sabiranje');
else
    rezultat = prod(niz1) * prod(niz2);
    disp('Izabrana operacija: množenje');
end

% Ispis oba niza
disp('Prvi niz:');
disp(niz1);
disp('Drugi niz:');
disp(niz2);

```



```
% Ispis sume ili proizvoda svih elemenata

if broj_parnih_niz1 > length(niz1) / 2

    disp(['Suma svih elemenata nizova: ' num2str(rezultat)]);

else

    disp(['Proizvod svih elemenata nizova: ' num2str(rezultat)]);

end
```

```
% Niz koji je nastao spajanjem nizova

spojeni_niz = [niz1, niz2];

disp('Niz koji je nastao spajanjem:');

disp(spojeni_niz);
```

**%% Unijeti dvije matrice sa tastature i izvršiti provjeru da li se dvije matrice mogu oduzimati, ako mogu oduzeti te dvije matrice, ako ne mogu provjeriti da li se mogu mnoziti, ako mogu pomnoziti ih, a ako ne mogu onda ispisati komentar da se date matrice ne mogu ni oduzimati ni mnoziti. Kao rezultat ispisati te dvije matrice i operacije koje su vršene nad njima.**

```
r1 = input('Unesite broj redova matrice');
k1 = input('Unesite broj kolona matrice');

for i = 1:r1
    for j = 1:k1

        prvaMatrica(i,j) = input('Unesite elemente prve matrice')
    end
end

r2 = input('Unesite broj redova druge matrice');
k2 = input('Unesite broj kolona druge matrice');

for i = 1:r2
    for j = 1:k2

        drugaMatrica(i,j) = input('Unesite elemente druge matrice')
    end
end
```

```

end

disp("Prva matrica koju ste unijeli" );
disp(prvaMatrica);

disp('Druga matrica koju ste unijeli');
disp(drugaMatrica);

% Dvije matrice se mogu oduzeti ako im je isti broj redova i kolona

if size(prvaMatrica) == size(drugaMatrica)
    disp('Matrice se mogu oduzeti')
    oduzimanjeMatrica = prvaMatrica - drugaMatrica;

    disp("Rezultat oduzimanja matrica je")
    disp(oduzimanjeMatrica)
else
    disp('Matrice se ne mogu oduzeti')
end

% Dvije matrice se mogu mnoziti ako je broj kolona prve jednak broju redova
druge matrice.

if k1 == r2
    disp('Matrice se mogu pomnoziti');
    mnozenjeMatrica = prvaMatrica * drugaMatrica;

    disp('Rezultat mnozenja je')
    disp(mnozenjeMatrica)
else
    disp('Matrice se ne mogu pomnoziti')
end

```

## **%% ZADATAK 2**

**%Nacrtati funkciju  $f = e^{-t/T} \sin(\omega t - \phi)$  za tri razlicite vrijednosti T,  $\omega$  i  $\phi$ . Koristeci funkciju**

**%subplot prikazati dobijene dijagrame na tri razlicita polozaja koristeći naredbu figure.**

```

clear all
close all
clc

```

```

T_values = [1, 2, 3];
omega_values = [1, 2, 3];
phi_values = [0, pi/4, pi/2];

```

```

% Definisanje vremenskog intervala
t = linspace(0, 2*pi, 100);

% Kreiranje novog prozora za grafike
figure;

% Prikaz grafika za svaki set parametara koristeći subplot
for i = 1:length(T_values)
    % Izračunavanje funkcije za trenutne parametre
    f = exp(-t/T_values(i)) .* sin(omega_values(i) * t - fi_values(i));

    % Prikaz dijagrama
    subplot(3, 1, i);
    plot(t, f);

    % Podešavanje naslova i oznaka osa
    title(['T = ', num2str(T_values(i)), ', \omega = ', num2str(omega_values(i)), ', \phi = ', num2str(fi_values(i))]);
    xlabel('t');
    ylabel('f');

    % Podešavanje granica osa y da budu iste za sve grafike
    ylim([-1.5, 1.5]);
end

```

**%% Napisati program koji unosi broj studenata, imena i prezimena i ocjene studenata na ispitu. Upotrebom switch case strukture prebrojati broj studenata sa ocjenom 6,7,8,9,10. Ako se unese pogresna ocjena program mora ispisati da je unesena ocjena pogresna.**

```

brojStudenata = input("Unesite broj studenata. ");

brojSestica = 0;
brojSedmica = 0;
brojOsmica = 0;
brojDevetki = 0;
brojDesetki = 0;

for i = 1:brojStudenata
    imena = input('Unesite ime studenta', 's')
    prezime = input('Unesite prezime studenta', 's')
    ocjena = input('Unesite ocjenu studenta', 's')

```

```
switch ocjena
case 6
    brojSestica = brojSestica+1;
case 7
    brojSedmica = brojSedmica+1;
case 8
    brojOsmica = brojOsmica+1;
case 9
    brojDevetki = brojDevetki+1;
case 10
    brojDesetki = brojDesetki+1;
otherwise
    disp("Unijeli ste pogresnu ocjenu")
end
end
fprintf("Sestica ima %d, Sedmica ima %d, Osmica ima %d, Devetki ima %d,
Desetki ima %d", brojSestica,brojSedmica,brojOsmica,brojDevetki,brojDesetki);
```