FON
Materijal ZA RAD U MATLABU
NIKOLA GROZDANOVIĆ

UVOD

>> **%** simbol procenta služi za komentare poput ovog teksta

---

>> %inicijalizacija promenljive izgleda ovako:
>> **y = 3;**

%posto smo napisali **;** neće se ispisati vrednost y

% ali čim pozovemo y
>> y

y =
3

>> %dobijamo njenu vrednost

>> z = 2*y + **...**
2

z =

8

>> z je čekao na još neku vrednost (...)

>> %komandom **ans** možemo pozvati neku operaciju kojoj nismo dodelili
%promenljivu,npr:
>>
>> 8+z*y;
>>
>> ans

ans =

32

>> %**pi** je u matlabu oznaka za istoimeni trig. broj
>> pi

ans =
3.1416

>> %**inf** je oznaka za beskonačno, kao npr

```
>>
>> 5/0
ans =
 Inf
```

>> %**NaN** je oznaka za not a number, kao npr

```
>> 0/0
ans =
   NaN
```

>> %opcijom **whos** listamo sve promenljive koje su definisane

```
>> whos
  Name      Size            Bytes  Class     Attributes

  ans     1x1                 8  double
  y       1x1                 8  double
  z       1x1                 8  double
```

>> %sa **clear** brišemo sve promenljive a mozemo i pojedinačne navodeći ključnu reč pored koje se stavlja naziv promenljive

---

```
>> s = 10/7

s =
   1.4286

>> format short
>> s

s =
   1.4286

>> format long
>> s

s =
  1.428571428571429
```

>> %uz pomoć **format short** (što je default u matlabu) i **format long**
% određujemo koji format zaokruživanja želimo, duži ili kraći

---

%Neke od standardnih matematičkih funkcija su:

```
>> sqrt(36)

ans =
    6
```

% **sqrt()** je korena funkcija

```
>> 2^3
ans =
```

8

% **^** za stepenu funkciju

>>log(exp(1))

ans =

1

% **exp** () je eksponencijalna funkcija koja broj e diže na stepen u zagradi, a **log** je logaritam sa osnovom e - u matematici često označavan sa ln

>> log2(2)

ans =

1

% **log_(x)** za logaritam sa bilo kojom osnovom koja se upisuje pre broja u zagradi

---

>> factorial(3)

ans =

6

% **factorial()** za faktorijel nekog broja

>> sin(pi/2) + cos(pi/2) + tan(pi/4) + cot(pi/4)

ans =

3

% **sin() cos() tan() cot()** za sinus, kosinus , tangens i kotangens

---

>> round(0.89)

ans =

1

% **round** zaokružuje na ceo broj
% **fix** zaokružuje ka nuli, **floor** ka minus beskonačno, **ceil** ka plus beskonačno

---

>> sign(-3.3)

ans =
-1

%funkcija **sign()** vraca +1 ili -1 u zavisnosti od znaka broja

---

>> rem(4,2)

ans =

0

>> rem(4,3)

ans =

1

% rem(x,y) ostatak pri deljenju broja x brojem y

>> i^2

ans =

-1

% **i** oznaka za imaginarnu jedinicu čiji je kvadrat jednak -1

VEKTORI I MATRICE

>> %prikaz predstavljanja vektora reda

>> **v=[1 2 3 4 5]**

v =
1 2 3 4 5

>>% vektor kolone
>> **s = [1;2;3;4;5]**

s =
1
2
3
4
5

>> %kombinacijom vektora kolona i redova dobija se MATRICA

matrica = [1 2 3; 4 5 6; 7 8 9]
matrica =

1 2 3
4 5 6
7 8 9

>> matrica(4)

ans =

2
%pristupanje elementu matrice kreće se brojanjem po kolonama (po vertikali)

```
>> praznaMatrica = []
praznaMatrica =

    []

>> praznaMatrica(1) = 7

praznaMatrica =

    7
```

%punjenje prazne matrice

```
matrica =

    1    2    3
    4    5    6
    7    8    9

>> matrica(1,3)

ans =

    3
```

%prvi argument odgovara broju reda a drugi argument se odnosi na broj po redu koji se trazi(tj koja kolona)

```
>> matrica(2,[1,2])

ans =

    4    5
```

% iz drugog reda izdvojiti 1 i 2. element

```
matrica =

    1    2    3
    4    5    6
    7    8    9

>> matrica([1,3],[2,3])

ans =

    2    3
    8    9
```

%pravi se podmatrica, argumenti u prvi uglastim zagradama označvaju koji redovi se biraju a drugi na koje elemente iz tih redova

---

```
>> matrica(end)

ans =
```

```
    9
% poslednji član iz matrice (član poslednjeg reda)
```

>> **matrica(end,:)**

```
ans =

    7    8    9
%poslednji red matrice
```

>> **matrica(:,end)**

```
ans =

    3
    6
    9
%poslednja kolona matrice
```

>> **matrica(end-1,:)**

```
ans =

    4    5    6
%pretposlednji red matrice
```

>> **matrica = [matrica;[5 5 5]]**

```
matrica =

    1    2    3
    4    5    6
    7    8    9
    5    5    5
%proširenje matrice za novi red na kraju
```

```
>> matrica=[[4 4 4];matrica]

matrica =

    4    4    4
    1    2    3
    4    5    6
    7    8    9
    5    5    5
%proširenje matrice za novi red na pocetku
```

```
>> linspace(0,1,5)

ans =

  Columns 1 through 4
```

0 0.250000000000000 0.500000000000000 0.750000000000000

Column 5

1.000000000000000

% **linspace(x,y,z)** pokazuje inkrementiranje od x do y kroz z tacaka

---

100:3:200

ans =

Columns 1 through 13

100 103 106 109 112 115 118 121 124 127 130 133 136

Columns 14 through 26

139 142 145 148 151 154 157 160 163 166 169 172 175

Columns 27 through 34

178 181 184 187 190 193 196 199

% **X:Y:Z** inkrementiranje kada znamo veličinu inkrementa, od broja X do broja Z pomerajući se za Y

---

matrica =

1 2 3
4 5 6

>> matrica(1:2, 3)

ans =

3
6

% **matrica(X:Y,Z)** u svakom redu od X:Y, prikazuje se Z. po redu element

---

matrica =

1 2 3
4 5 6

>> matrica(:,2)=[]

matrica =

1 3
4 6

%brisanje druge kolone matrice

>> matrica(1,:)=[]

matrica =

   4   6

%brisanje prvog reda

---

matrica =

   1   2   3
   4   5   6
   7   8   9

>> matrica(:,end+1)=[0 0 0]

matrica =

   1   2   3   0
   4   5   6   0
   7   8   9   0

%dodavanje nove poslednje kolone

>> matrica(end+1,:)=[-1 -1 -1 -1]

matrica =

   1   2   3   0
   4   5   6   0
   7   8   9   0
  -1 -1 -1 -1

%dodavanje reda na kraju

>> k =**zeros**(3) %matrica nula, dimenazije 3

k =

   0   0   0
   0   0   0
   0   0   0

>> q = **ones(**3) %matrica jedinica, dimenzije 3

q =

   1   1   1
   1   1   1
   1   1   1

>> **eye**(4)

ans =

   1   0   0   0
   0   1   0   0
   0   0   1   0
   0   0   0   1

%matrica 4x4 ciji su elementi na glavnoj dijagonali jednaki 1 a ostali 0

---

rand(3,3)

ans =

```
  0.157613081677548   0.485375648722841   0.421761282626275
  0.970592781760616   0.800280468888800   0.915735525189067
  0.957166948242946   0.141886338627215   0.792207329559554
```

% rand(x,y) matrica random slučajnih brojeva, dimenzije x sa y, po uniformnoj raspodeli;
randn generiše po normalnoj raspodeli

```
>> fix(100*rand(3,3))
ans =

   95    84    75
   65    93    74
    3    67    39
```

%ovo je primer matrice nenegativnih celih random brojeva od 0 do 100 (množenjem se zarez pomera za dva mesta, a pomocu funkcije fix zaokružujemo brojeve ka nuli)

---

matrica =

```
   1    2    3
   4    5    6
   7    8    9
```

>> **triu(matrica)**

ans =

```
   1    2    3
   0    5    6
   0    0    9
```

%gornja trougaona matrca

>> **tril(matrica)**

ans =

```
   1    0    0
   4    5    0
   7    8    9
```

%donja trougaona matrica

>> magic(3)
ans =

```
   8    1    6
   3    5    7
   4    9    2
```

%magic generiše matricu čiji je zbir po svim kolonama i redovima jednak (15 u ovom slučaju)

>> **matrica*matrica**

```
ans =

   30   36   42
   66   81   96
  102  126  150
% množenje po principu prvi red prvom kolonom
```

```
matrica =

    1    2    3
    4    5    6
    7    8    9

>> matrica*2

ans =

    2    4    6
    8   10   12
   14   16   18
%množenje svih elemenata matrice skalarom
```

```
>> matrica(1,[1,2,3]) = 2*matrica(1,[1,2,3])

matrica =

    2    4    6
    4    5    6
    7    8    9
%množenje prvog reda skalarom
```

```
matrica =

    2    4    6
    4    5    6
    7    8    9
>> matrica(:,2)=3*(matrica(:,2))

matrica =

    2   12    6
    4   15    6
    7   24    9
%množenje druge kolone skalarom
```

```
matrica =

    2   12    6
    4   15    6
    7   24    9

>> matrica.*[1 1 1;2 2 2 ; 3 3 3 ]

ans =
```

```
    2    12     6
    8    30    12
   21    72    27
```
%množenje svakog elementa redova obeju matrica odgovarajućim skalarom

---

matrica =

```
    2    12     6
    4    15     6
    7    24     9
```

>> matrica.^2

ans =

```
    4   144    36
   16   225    36
   49   576    81
```
%što je ekvivalentno da smo napisali matrica.*matrica

>> matrica.^[2 2 2; 3 3 3; 4 4 4]

ans =

```
        4        144         36
       64       3375        216
     2401     331776       6561
```
%dizanje na stepen u uglastim zagradama, po svakom redu matrice

---

matrica =

```
    1     2     3
    4     5     6
    7     8     9
```

>> sum(matrica)

ans =

```
   12    15    18
```
%suma po kolonama matrice(po defaultu)

>> sum(matrica,2)

ans =

```
    6
   15
   24
```
%za sumu po svim redovima

>>**sum(sum(matrica))** suma cele matrice

>> l = **matrica'**

l =

```
    1     4     7
    2     5     8
```

```
   3    6    9
%transponovana matrica (kolone i redovi zamene mesta)
```

>>**diag(l)**

```
ans =

   1
   5
   9
% diag() izdvaja vektor glavne dijagonale matrice
```

```
l =
   1    2    3
   4    5    6
   7    8    9
```

>> **fliplr(l)**

```
ans =

   3    2    1
   6    5    4
   9    8    7
% lik matrice u ogledalu
```

```
matrica =

   1    4    7    4
   2    5    8    4
   3    6    9    4
   5    5    5    4
```

>> **trace(matrica)**

```
ans =

   19
%suma glavne dijagonale kvadratne matrice
```

```
matrica =
   1    4    7    4
   3    6    9    4
   5    5    5    4
```

>> **max(matrica)**

```
ans =
   5    6    9    4
%najveci element iz svake kolone
```

>>**min(matrica)**

```
%analogno daje najmanje elemente po kolonama
>> min(matrica)
```

ans =

1 4 5 4

>>**min(matrica,[],2)** i **max(matrica,[],2)** daju najmanje i najveće elemente po redovima matrice

% **max(max(matrica))** i **min(min(matrica))** najveći i najmanji elementi matrice

---

matrica =

1 4 4
3 6 4
5 5 4

>> **length(matrica)**

ans =

3

%dužina(dimenzija) matrice

>> **[a,b]=size(matrica)** %vraća broj kolona i redova matrice

---

matrica =
1 2 3
4 5 6
7 8 9
>> **repmat(matrica,1,2)**
ans =
1 2 3 1 2 3
4 5 6 4 5 6
7 8 9 7 8 9

>> **repmat(matrica,1,3)**
ans =
1 2 3 1 2 3 1 2 3
4 5 6 4 5 6 4 5 6
7 8 9 7 8 9 7 8 9

>> repmat(matrica,2,1)
ans =
1 2 3
4 5 6
7 8 9
1 2 3
4 5 6
7 8 9
%repliciranje matrice

---

matrica =

1 2 3

```
   4    5    6
   7    8    9
```

>> any(matrica)

```
ans =

   1    1    1
```

%barem jedan po koloni je različit od nule, kada vrati 1 to znaci da postoji, kada je 0 onda su svi u koloni 0

>> **all(matrica)**

```
ans =

   1    1    1
```

%ispituje da li su svi po kolonama razliciti od nule

% **all(all(matrica))** i **any(any(matrica))** ispituje se da li su svi u matrici različiti od nule odnosno da li u celoj postoji makar jedan koji je različit od nule

---

```
matrica =

   1    2    3
   4    5    6
   7    8    9
```

>> **mean(matrica)**

```
ans =

   4    5    6
```

%prosečna vrednost svake kolone

>> **mean(mean(matrica))**

```
ans =

   5
```

%prosecna vrednost cele matrice

>> **mean(matrica')**

```
ans =

   2    5    8
```

% ovo moze biti efikasan nacin da pronađemo srednju vrednost po redovima matrice tako što upotrebimo funkciju mean a kao parametar prosledimo transponovanu matricu čije su kolone sada redovi matrice

a mozemo i ovako:
>> **mean(matrica,2)**

```
ans =
```

2
5
8

---

matrica =
1 2 3
4 5 6
7 8 9

>> **median(matrica)**
ans =

4 5 6

>> **median(median(matrica))**

ans =

5
%statistička veličina koja deli ukupan broj elemenata na jednake delove (medijana)

---

matrica =--
1 2 3
4 5 6
7 8 9

>> std(matrica)

ans =

3 3 3
%standardna devijacija kolona matrice

>> **std(std(matrica))**

ans =

0
%standardna devijacija elemenata matrice (koren iz varijanse po formuli)

---

matrica =

1 2 3
4 5 6
7 8 9

>> **prod(matrica)**

ans =

28 80 162
%množenje elemenata međusobno po kolonama
>> **prod(prod(matrica))**

```
ans =

     362880
%proizvod cele matrice

>>prod(matrica,2)

ans =
    6
  120
  504
%proizvod po redovima
```

```
y =
    1    20     0
   34    12    90
   34    33    11

>> sort(y)

ans =

    1    12     0
   34    20    11
   34    33    90
%sortiranje elemenata u kolonama matrice

>> sort(y,'descend')

ans =

   34    33    90
   34    20    11
   1     12     0
%sortiranje elemenata kolona matrice od većeg ka najmanjem; suprotno je 'ascend'
```

```
y =
    1    20     0
   34    12    90
   34    33    11

>> det(y)
ans =

     50882  %za računanje determinante matrice (primenom npr. Saruzovog pravila)
>> rank(y)

ans =

    3
%rang matice- najveca podmatrica čija je determinanta različita od nule


matrica =
```

```
    1    2    3
    4    5    6
    7    8    9

>> [x]=find(matrica>4)

x =

    3
    5
    6
    8
    9
%pozicije u matrici koje zadovoljavaju uslov

>> [x,y]=find(matrica>7)

x =

    3
    3

y =

    2
    3
%x predstavlja koji je red a y koja kolona u čijem preseku se nalazi broj koji zadovoljava uslov
```

```
matrica =

    1    2    3
    4    5    6
    7    8    9

>> [q,w]=eig(matrica)

q =
  -0.231970687246286  -0.785830238742067   0.408248290463864
  -0.525322093301234  -0.086751339256628  -0.816496580927726
  -0.818673499356182   0.612327560228810   0.408248290463863

w =
  16.116843969807043                   0                   0
                   0  -1.116843969807043                   0
                   0                   0  -0.000000000000001
%sopstveni vektori matrice (zadovoljavaju uslov matrica*q=q*w)
```

```
%polinom 8x+5 prvog stepena se moze zapisati kao
y = [8 5];
>> polyval(y,1)

ans =
```

```
   13
%funkcija polyval vraća vrednost polinoma datog kao prvi argument u tački koja je data kao drugi
argument

>> z=[1 2 1];
%što je zapravo polinom drugog stepena: x^2+2x+1 tj kvadrat binoma
>> roots(z)

ans =

   -1
   -1
%vraća nule polinoma
```

```
>> z=[3 4 1];
>> s=roots(z)

s =

  -1.000000000000000
  -0.333333333333333

>> poly(s)

ans =

   1.000000000000000   1.333333333333333   0.333333333333333
%ako znamo nule polinoma lako dolazimo i do koeficijenata  preko funkcije poly
```

```
>> q = [1 2 3];
>> s = [5 6 7];
>> s+q

ans =

    6    8   10
%klasično sabiranje polinoma gde se redovi polinoma moraju poklapati
```

```
>> a = [1 0 1 ];
>> b = [2 7];
>> c = conv(a,b);
>> %c je vektor koeficijenata ispred polinoma dobijenog množenjem dva polinoma u konkretnom slucaju
množenjem polinoma x^2+1 i 2x+7 dobija se polinom x^3+7x^2+2x+7 čiji su koeficijenti upravo:

c =

    2    7    2    7
```

FUNKCIJE:

```
%FUNKCIJA1 generise matricu dimenzija m,n brojeva od 0 do 100

function [N] = funkcija1(m,n)
N=fix(100*rand(m,n));
end
```

```
>> M = funkcija1(2,2)

M =

    81    12
    90    91
```

```
%funkcija koja proverava parnost unetog broja

function [] = funkcija2(broj)
    if(broj==0) disp('nula')
    else if (rem(broj,2)==0) disp('paran')
        else disp('neparan')
        end
    end
end
```

```
%funkcija proverava koji je broj veci
function [] = funkcija3(broj1,broj2 )
    if(broj1-broj2)==0 disp('brojevi su jednaki');
    else if(broj1>broj2) disp('prvi je veci');
        else disp('drugi je veci');
        end
    end
end
```

```
%funckija ispisuje niz kvadrata brojeva do unetog broja
function [] = funkcija4(broj)
niz = [];

for i=1:broj
    niz(i)=i^2;
end
disp(['Kvadrati brojeva do broja:',int2str(broj)]);
disp(niz);
end
```

```
%matrica slucajnih elemenata iz normalne raspodele
function [r] = funkcija1( m, n , mi, sigma )
r = mi + sigma.*randn(m,n);
disp (['Matrica slucajnih brojeva iz normalne raspodele sa mi= '
int2str(mi) ' i sigma= ' int2str(sigma)])
end
```

```
>> k = funkcija5(3,3,0,1)
Matrica slucajnih brojeva iz normalne raspodele sa mi= 0 i sigma=
1
```

```
k =

  -0.432564811528221   0.287676420358549   1.189164201652103
  -1.665584378238097  -1.146471350681464  -0.037633276593318
   0.125332306474831   1.190915465642999   0.327292361408654
```

```
%matrica dimenzije m puta n, slucajnih elemenata iz uniformne raspodele
sa parametrima a i b
function [ r ] = funkcija2( m, n, a, b )

r = a+(b-a).*rand(m, n);

disp( ['Matrica slucajnih brojeva od ' int2str(a) ' do ' int2str(b)
'je: '])

end
```

```
>> k = funkcija6(3,3,0,1)
Matrica slucajnih brojeva od 0 do 1je:

k =

   0.632359246225410   0.546881519204984   0.157613081677548
   0.097540404999410   0.957506835434298   0.970592781760616
   0.278498218867048   0.964888535199277   0.957166948242946
```

```
%Funkcija koja dodaje na kraj kolonu dvostruko vecu u odnosu na
poslednju kolonu, sve dok zbir ne predje 100, kada prestaje sa radom
function [ r ] = funkcija3( A )

while (sum(sum(A))<100)
    A= [A 2*A(:, end)];
end

disp(A)
end
```

```
matrica =

     1     3     5
     2     4     6
     7     8     9

>> funkcija7(matrica)
     1     3     5    10    20
     2     4     6    12    24
     7     8     9    18    36
```

```
%Funkcija proverava da li je matrica kvadratna a ako nije pravi
kvadratnu
```

```
%od postojece (brise poslednju kolonu/red)
function [] = funkcija8(matrica)
[a,b]=size(matrica);
if(a==b)
    disp('Matrica je kvadratna')
elseif(a>b)
        while(a>b)
            matrica(end,:)=[];
            a = a-1;
        end
else
    while(a<b)
            matrica(:,end)=[];
            b = b-1;
        end
    end

disp(matrica)

end
```

matrica =

```
  1  4
  2  5
  3  6
```

>> funkcija8(matrica)

```
  1  4
  2  5
```

```
%Funkcija ispisuje kvadrate neparnih elemenata od 1 do 20
function [] = funkcija10( )
j=0;
r=[];
    while j<100
        j=j+1;
    if(rem(j,2))==0   %rem funkcija za ostatak pri deljenju
        continue   %continue preskace elemente koji ne zadovoljavaju
                    definisan uslov
    end
    if(j>20)
        break;     %break prekida petlju
    end
    r=[r j^2]; %da bi se u vektor ubacile sve vrednosti a ne samo
               %poslednja
end
disp(r)
end
```

%ista funkcija samo uz pomoc for petlje:

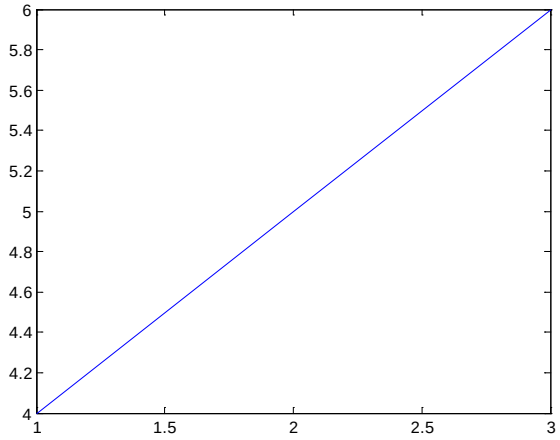
```
%funkcija ispisuje kvadrate neparnih elemenata od 1 do 20
function [] = funkcija10( )
i = 0;
red = [];
for i=1:20
```

```
    if(rem(i,2)==0)
        continue;
    end
    red = [red i^2];
end
disp(red)
end
```

```
>> funkcija10()
     1     9    25    49    81   121   169   225   289   361
```

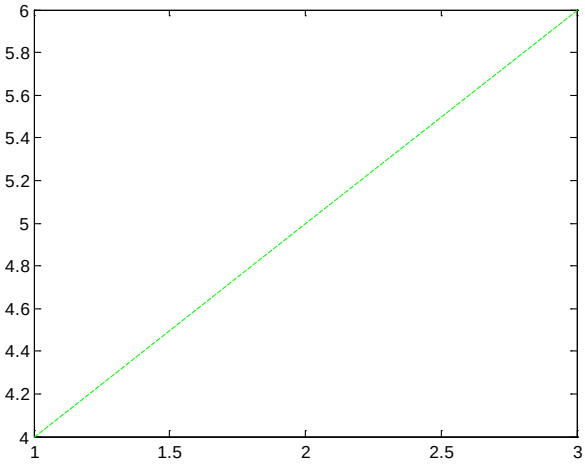
RAD SA GRAFICIMA:

```
>> x=[1 2 3];
>> y=[4 5 6];
>> plot(x,y)
```




```
>>plot(x,y,'g--')
```

%green -- stil linije
% ‘b-’ puna linija plave boje
% ‘r-’ puna linija crvene boje
% ‘--’ crtanje isprekidanom linijom
% ‘--y’ isprekidana linija žute boje
% ‘o’ crtanje malog slova o umesto linija



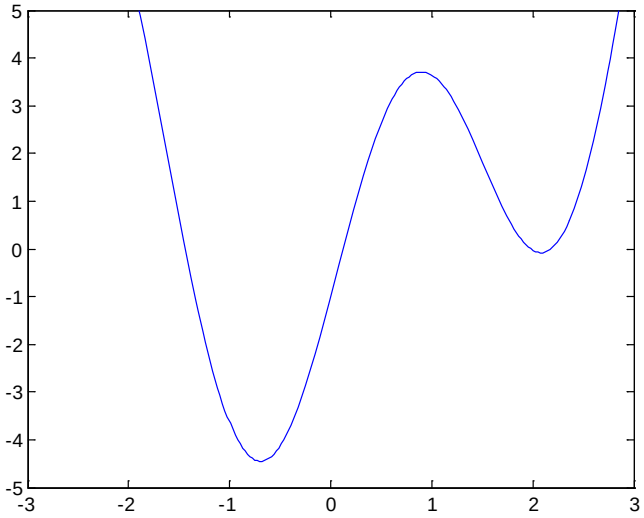
%jedan grafik - 3 funkcije (ili više, samo je bitno da se postuje oblik pisanja- prvo nezavisna promenljiva pa zavisna, i to za svaku funkciju)

```
>>plot(x,y,x,z,x,k)
```

%za postupno crtanje grafik po grafike funkcija-opcija **hold on** (čekanje) i **hold off**(za prekid čekanja)

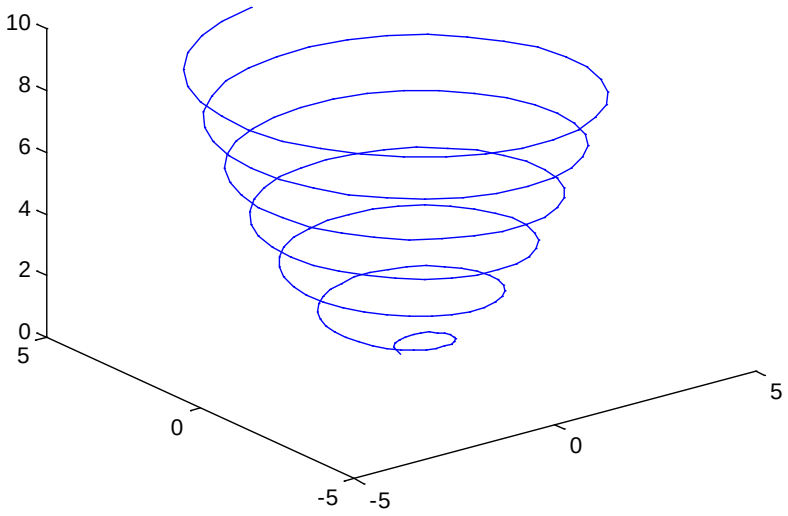
```
>> x = linspace(0,1,100);
>> fplot('x^2+4*sin(2*x)-1',[-3,3,-5,5])
```

%fplot kao argument prima polinomsku funkciju i interval za prikaz u uglastim zagradama
%1. uglaste zagrade se odnose na ograničenje u prikazu x-a a druge na y



% **plot3**

```
t=0:0.1:6*pi;
x=sqrt(t).*sin(2*t);
y=sqrt(t).*cos(2*t);
z=0.5*t;
% prikaz grafika u trodimenzionalnom prostoru
plot3(x,y,z,'b','linewidth',1)
```



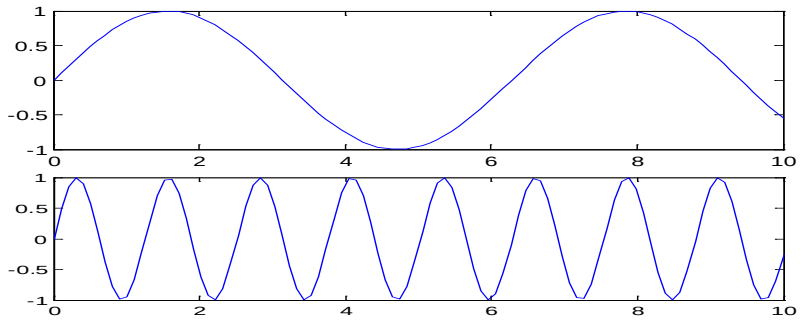
%xlabel('x');ylabel('y');zlabel('z') ovime se moze naznačiti imena osa

%**subplot(x,y,I)** omogućava da na jednom odeljku imamo bolji (veći) prikaz grafika funkcije, gde x podrazumeva proširenje po redu,y po koloni (srazmerno ponavljanje delova grafika na osama) a I koji po redu grafik crtamo

```
x = linspace(0,10);
y1 = sin(x);
y2 = sin(5*x);
```

```
subplot(2,1,1);
plot(x,y1)

subplot(2,1,2);
plot(x,y2)
```



```
x = [0.9 1.6 3 4 6 8 9.5];
y = [0.9 1.5 2.5 5.1 4.5 4.9 6.3];
p = polyfit(x,y,3)

p =

  Columns 1 through 3

   0.021377425666717  -0.391415737771255   2.590505636426509

  Column 4

  -1.450729348772463
```

%polyfit vrši aproksimaciju funkcije pomoću zadatog stepena polinoma (u ovom slučaju 3. stepen)

%pored standardnog prikaza, grafici se mogu prikazati i u okviru :

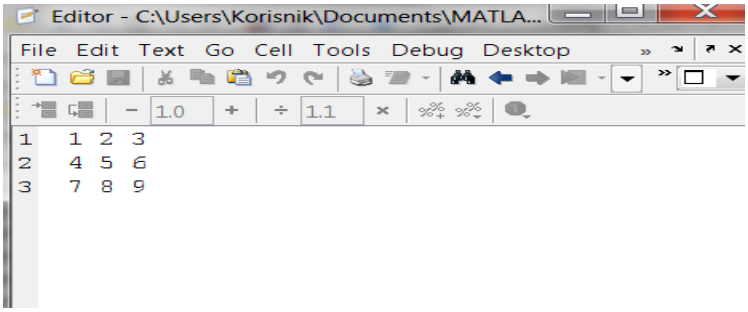
- **bar(x,y)** i **barh(x,y)** - vertikalni i horizontalni stubići,
- **stairs(x,y)** u obliku stepenica,
- **pie(x)** -pita tj. procentualno učešće.

%**plottools** komandom ulazi se u tool režim u kom je najlakše posmatrati i menjati grafik

PRIPREMA ZA PRVI KOLOKVIJUM(ZADACI SA SAJTA):

Задатак 1. Направити произвољну матрицу у неком текстуалном едитору нпр. *Notepad*. Учитати датотекуи сачувати матрицу као променљиву A. Наћи вредност полинома p дефинисаног споредном дијагоналом матрице A у тачки која је једнака најмањем елементу матрице A и приказати резултат.

U Current Directory pravimo novi M-File koji ima neko ima i ektenziju .txt u kojem unosimo u redovima elemente matrice.



A = load('Notepad.txt')

VAŽNO: komandama **xlsread('Nazivfajla.xls');** učitava se neki Excel fajl sa odgovarajućom strukturom kolona i vrsta a **xlswrite(NazivFajla, šta se upisuje, u koji sheet se upisuje)** služi za upisivanje podataka u sheet Excel-a;
rad ovih komandi može se bolje videti ukucavanje help xlsread ili help xlswrite

A =

```
   1   2   3
   4   5   6
   7   8   9
```

>> sporednaDijagonala = diag(fliplr(A))

sporednaDijagonala =

```
   3
   5
   7
```

>> minimum = min(min(A))

minimum =

```
   1
```

>> resenje = polyval(sporednaDijagonala,minimum)

resenje =

15

Задатак 2. Направити функцију која за прослеђену матрицу M проверава да ли је матрица празна и ако јесте исписује поруку о томе. Ако није празна, испитују се димензије матрице односно исписује се порука да ли је већи број врста или број колона.

Editor - C:\Users\Korisnik\Documents\MATLAB\priprema2.m*

File Edit Text Go Cell Tools Debug Desktop Window Help

Stack: Base

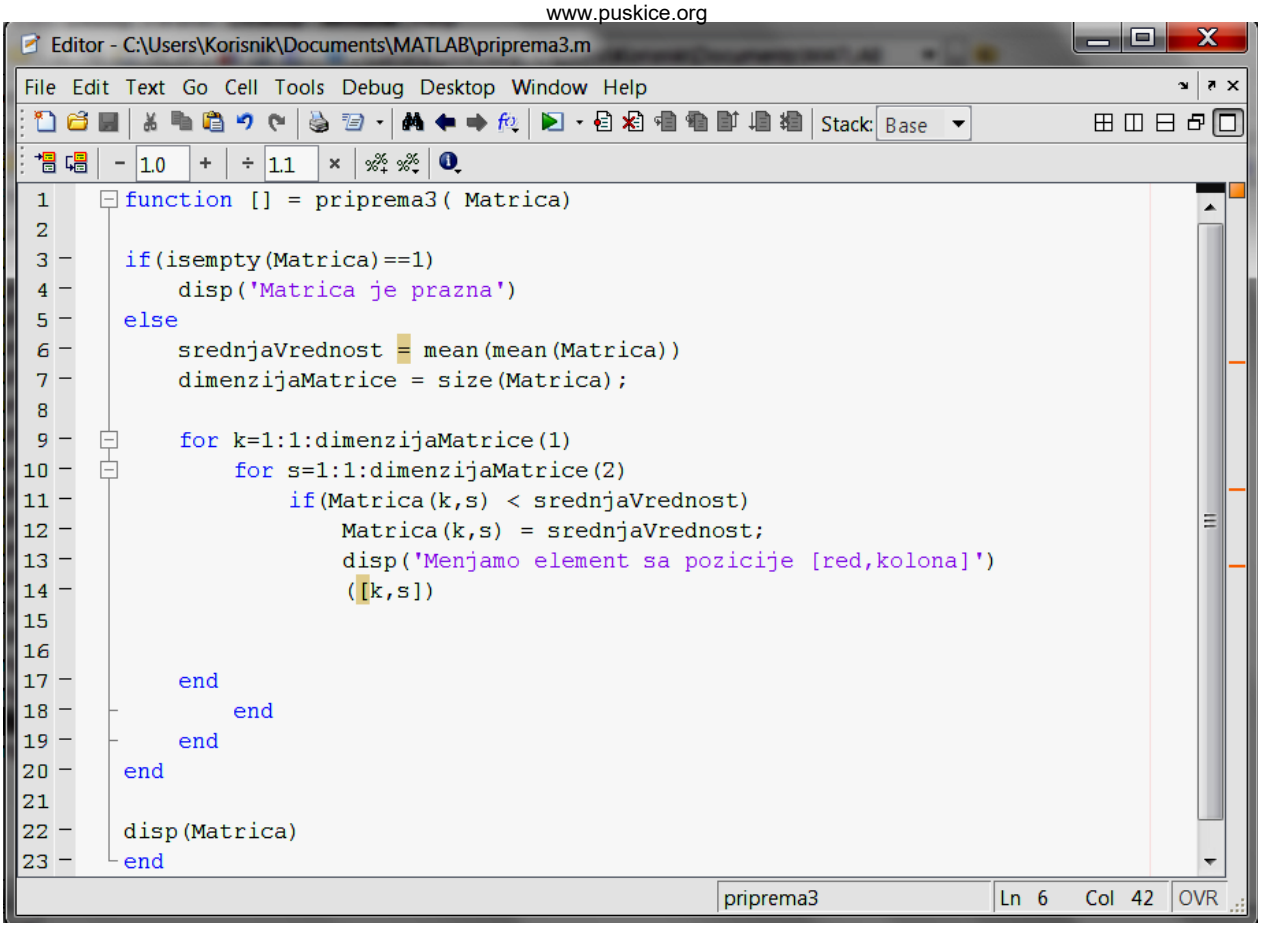
1.0 1.1

```
function [] = priprema2(M)
if(isempty(M)== 1)
    disp('Prosledjena matrica je prazna.');
elseif (isempty(M)==0)
    duzina = size(M);
    if(duzina(1)-duzina(2)==0)
        disp('Matrica je kvadratna')
    elseif(duzina(1)-duzina(2)>0)
            disp('Matrica ima vise redova nego kolona')
    elseif(duzina(1)-duzina(2)<0)
        disp('Matrica ima vise kolona nego redova')
    end
end
end


%elegantnije resenje : switch sign(duzina(1)-duzina(2))
     %case 0 disp('Matrica je kvadratna')
     %case 1 disp('Matrica ima veci broj redova')
     %case -1 disp('Matrica ima veci broj kolona')
     %otherwise disp('Greska')

```

Задатак 3. За прослеђену матрицу написати функцију која проналази и приказује позицију свих елемената матрице који су мањи од средње вредности, а затим мења све пронађене елементе средњом вредношћу матрице.




```
function [] = priprema3( Matrica)

if(isempty(Matrica)==1)
    disp('Matrica je prazna')
else
    srednjaVrednost = mean(mean(Matrica))
    dimenzijaMatrice = size(Matrica);

    for k=1:1:dimenzijaMatrice(1)
        for s=1:1:dimenzijaMatrice(2)
            if(Matrica(k,s) < srednjaVrednost)
                Matrica(k,s) = srednjaVrednost;
                disp('Menjamo element sa pozicije [red,kolona]')
                ([k,s])


    end
        end
    end
end

disp(Matrica)
end
```

Задатак 4. За прослеђену матрицу написати функцију којом се рачуна и враћа збир свих елемената испод главне дијагонале. У случају да је збир већи од збира елемената првог реда матрице исписати одговарајућу поруку.

```
function [k] = priprema4(Matrica)

k = sum(sum(tril(Matrica)))- sum(diag(Matrica));
zbirPrvogReda = sum(Matrica(1,:));

if(k>zbirPrvogReda)
    disp('Veci je zbir elemenata ispod dijagonale')
elseif (k<zbirPrvogReda)
    disp('Veci je zbir elemenata prvog reda')
else disp('Jednaki su')
end
end
```

Задатак 5.Група студената ФОН-а која спрема испите заједно одлучила је да за Нову годину једни другима купе поклоне. Пера је изабрао да купи поклоне Ани и Марку, Ана је одлучила да купи поклоне Сањи и Пери, Марко Пери и Ани, а Сања само Ани.Направити матрицу која приказује ко коме купије поклон.

а) Израчунати број поклона који је потребно да купи и који ће да добије свако од студената.

б) Проверити да ли има неко од студената да не добија ниједан поклон, и да ли постоји студент који од свих студената добија поклон.

в) Израчунати колико сваки студент треба да издвоји за поклоне, ако се укупни износ формира преко полинома чији су коефицијенти елементи реда матрице који се односи на тог студента. Вредност полинома се израчунава за суму свих елемената матрице.

%podrazumeva se da svako sebi kupuje poklon; vertikalni i horiz. poredak je Petar,Ana,Marko,Sanja
%po horizontali se gleda ko kome kupuje a po vertikali ko od koga dobija poklon
MatricaPoklona =

```
   1   1   1   0
   1   1   0   1
   1   1   1   0
   0   1   0   1
```

a)

```
>> BrojKupljenih = sum(MatricaPoklona,2);
 %sumiranjem elemenata po redovima dobijamo broj koliko svako od njih treba da kupi
poklona(pristupanjem BrojKupljenih (1)....BrojKupljenih(4) jer je u pitanju niz vrednosti)

>> disp(['Broj poklona koje Petar kupuje : ' int2str(BrojKupljenih(1))])
Broj poklona koje Petar kupuje : 3
>> disp(['Broj poklona koje Ana kupuje : ' int2str(BrojKupljenih(2))])
Broj poklona koje Ana kupuje : 3
>> disp(['Broj poklona koje Marko kupuje : ' int2str(BrojKupljenih(3))])
Broj poklona koje Marko kupuje : 3
>> disp(['Broj poklona koje Sanja kupuje : ' int2str(BrojKupljenih(4))])
Broj poklona koje Sanja kupuje : 2

BrojDobijenih = sum(MatricaPoklona);
%sum po default-u racuna po kolonama matrice a to nam i treba da bismo izracunali br dobijenih
poklona; analogno, vrsimo ispis:
>>disp(['Broj poklona koje Petar dobija : ' int2str(BrojDobijenih(1))])
>>Broj poklona koje Petar dobija : 3
%i tako za svakog člana
```

b)

```
>> if(all(any(MatricaPoklona))==1)
disp('Svi dobijaju bar po 1');
else
disp('Ne dobijaju svi po jedan')
end
Svi dobijaju bar po 1

>> if(any(all(MatricaPoklona))==1)
disp('Postoji student koji od svih dobija poklon')
else
disp('Ne postoji student koji od svih dobija poklon')
end
Postoji student koji od svih dobija poklon  %to je Ana
```

c)

```
>> k = sum(sum(MatricaPoklona));
>> PetarTroskovi = MatricaPoklona(1,:);   %prvi red matrice

>> polyval(PetarTroskovi,k)

ans =

      1463
%i tako za svakog clana analogno
```

Задатак 6.Дата је матрица која представља оцене студената из наведених предмета.

	Математика 3	МТР	Базе података	Осн. организације	Енглески 1	ОИКТ
Стеван	9	8	6	9	10	7
Нина	10	6	7	9	9	10
Милица	8	10	9	7	7	9
Срђан	7	9	10	6	8	10
Марија	7	7	9	8	9	8

Направити функцију која за прослеђену матрицу А:

а) Налази студента и предмет са највећомпросечном оценом и исписује одговарајућу поруку.

б)За прва три студента, у засебним дијаграмима, приказује кретање оцена по предметима.

```
function [] = priprema6( Matrica )
proseciStudenti = mean(Matrica');
%redove prebacimo u kolone jer mean radi po kolonama
proseciOcene = mean(Matrica);
prosekNajboljegStudenta = max(proseciStudenti);
najvisaProsecnaOcenaPredmeta = max(proseciOcene);

imeStudenta = find(prosekNajboljegStudenta == proseciStudenti);
imePredmeta = find (najvisaProsecnaOcenaPredmeta == proseciOcene);
disp('Predmet sa najvisom prosecnom ocenom je: ')
switch(imePredmeta)
    case 1
        disp('Matematika3')
    case 2
        disp('MTR')
    case 3
        disp('Baze Podataka')
    case 4
        disp('Osnovi organizacije')
    case 5
        disp('Engleski1')
    case 6
        disp('OIKT')
    otherwise
        disp('Nema tog predmeta. Greska.')
end

disp('Najbolji student je:')
switch(imeStudenta)
    case 1
        disp('Stevan')
    case 2
        disp('Nina')
    case 3
        disp('Milica')
    case 4
        disp('Srdjan')
    case 5
        disp('Marija')
    otherwise
        disp('Nema tog studenta. Greska')
end
end
```

b)

```
[I,J] = size(M);
>> x = 1:J;  %x uzima vrednosti od 1 do broja kolona
>> y1 = M(1,x); %ocene za prvog studenta
>> y2 = M(2,x);
>> y3 = M(3,x);
>> figure  %nova figura za svako novo pokretanje
>> subplot(3,1,1);
>> plot(x,y1,'ro');
>> subplot(3,1,2);
```

```
>> plot(x,y2,'bo');
>> subplot(3,1,3);
>> plot(x,y3,'go');
```