



MATLAB ile ANALİZ (MIA)

DERS-5



SEÇME (SELECTION) TIPINDEKİ KONTROL DEYİMLERİ

- **for döngüsü:**
- Bir grup bilginin birkaç defa değerlendirilmesi
- gerekiyorsa bu yapı kullanılır.
- **for x=array**
- **commands**
- **end**
- Komutlar x'in bütün değerleri için işler. Örneğin;
-
- `num=[6 37 23 -1]`
- `for k=1:length(num)`
- `fprintf('%d inci elementi =%d\n', k,num(k));`
- `k=k+1;`
- `end`

çıktısı;

1 inci elementi 6
2 inci elementi 37
3 inci elementi 23
4 inci elementi -1



DÖNGÜLER

Rastgele bir dizideki pozitif ve negatif sayıları yazdırınız.

```
r = round(-10 + (20+10).*rand(1,20)) % random negatif, pozitif sayılar
i=1;
for i=1:length(r)
    if (r(i)>0)
        fprintf('r(%d) pozitif. Deđeri=%d\n',i,r(i));
    else
        fprintf('r(%d) negatif. Deđeri=%d\n',i,r(i));
    end
end
```



DÖNGÜLER

while döngüleri

Genel gösterimi;
**while expression
commands**

....

end

şeklindedir.



DÖNGÜLER

```
clc
toplam = 0;
while toplam < 100
sayi = input('sayi giriniz: ');
toplam = toplam + sayi;
fprintf('Toplam = %d \n', toplam)
end
```



DÖNGÜLER

break komutu

Break, bir for veya while döngüsünün yürütülmesini sonlandırır. Break ifadesinin ardından döngüdeki ifadeler yürütülemez. İç içe geçmiş döngüler içinde break, yalnızca içinde bulunduğu döngüden çıkar.

```
limit = 0.8;
s = 0;
clc;
while 1
    tmp = rand;
    fprintf('tmp=%2.2f\n',tmp);
    if tmp > limit
        break
    end
    s = s + tmp;
end
```

çıktısı;

```
tmp=0.75
tmp=0.26
tmp=0.51
tmp=0.70
tmp=0.89
```



DÖNGÜLER

continue

Bu yapı kontrol durumunu geçip sıradaki iterasyona ulaşmak için for ve while döngülerinde kullanılır.

```
clear;
```

```
clc;
```

```
for i=1:4
```

```
    for j=1:5
```

```
        if j>=3
```

```
            continue
```

```
        end
```

```
        fprintf('i=%d j=%d\n',i,j);
```

```
    end
```

```
end
```

çıktısı;

i=1 j=1

i=1 j=2

i=2 j=1

i=2 j=2

i=3 j=1

i=3 j=2

i=4 j=1

i=4 j=2



DÖNGÜLER

Örnek: $x = 100 - t^3$ fonksiyonu negatif değere düşünceye kadar $y = \sqrt{x}$ bulan bir kod oluşturunuz.

```
for t=1:10
    x=100-t.^3;
    if x<0
        break
    end
    y=sqrt(x);
    fprintf('t=%d x=%2.2f y=%2.2f\n',t,x,y);
end
```




Görüntü İşleme (Temel Matlab)

For ve İf örnekleri

$X^3 - 4.2x^2 + 3.3x - 4$ fonksiyonunun köklerini bulunuz.

```
tic
for x=-5:0.0001:5
    t=x.^3-4.2*x.^2+3.3.*x -4;
    % if(t==0)
    if (t>=-0.001 && t<=0.001)
        fprintf('Kök=%2.5f',t);
        fprintf(' değer=%2.5f\n',x);
    %     disp(t);
    end
end
toc
```



Görüntü İşleme (Temel Matlab)

$$f(x) = 2x - \cos x$$

Fonksiyonun köklerini döngüler kullanarak bulunuz.

```
tic
clc
clear
for x=-5:0.0001:5
    t=2*x-cos(x);

    if (t>=-0.0001 && t<=0.0001)
        fprintf('Kök=%2.5f',x);
        fprintf(' değer=%2.5f\n',t);
    end
end
```



Görüntü İşleme (Temel Matlab)

1'den 999'a kadar olan sayılardan asal sayı olanların toplamını bulunuz.

1. yöntem:

```
total = 0;
for k = 1:999
    if(isprime(k))
        total = total + k;
    end
end
disp(total)
```

2. yöntem

```
total = 0;
for k = primes(999)
    total = total + k;
end
disp(total)
```



Görüntü İşleme (Temel Matlab)

Aşağıdaki fonksiyonun -5:+5 aralığında grafiğini 10 farklı renk ile çizdiriniz.

$$y = x^2 - 1$$

```
x=(-5:5);  
y=x.^2-1;  
for i=1:10  
    r1=rand(1);  
    r2=rand(1);  
    r3=rand(1);  
p=plot(x,y);  
p.Color=[r1 r2 r3];  
p.LineWidth=2;  
pause(1);  
end
```



Görüntü İşleme (Temel Matlab)

1'den 200'e kadar olan sayılardan kendisi ve 2 fazlası asal olan sayıları bulunuz.

```
for x = 1:2:200
if(isprime(x) & isprime(x+2))
fprintf('%f ve %f ikisi de asal sayıdır\n',x,x+2)
end
end
```



Görüntü İşleme (Temel Matlab)

1'den 20'e kadar olan sayıları fibonacci dizisi olarak sıralayan programı yazınız.

1. yol:

```
clc; clear;
f(1)=1;
f(2)=1;
fprintf('1.sayı=%d\n',f(1));
fprintf('2.sayı=%d\n',f(2));
for i=3:20
    f(i)=f(i-1)+f(i-2);
    fprintf('%d. sayı=%d\n',i,f(i));
end
```

2.yol:

```
clc;clear;
birincisayi=0;
ikincisayi=1;
fprintf('%d\n',ikincisayi);
for i=1:20
    ucuncusayi=birincisayi+ikincisayi;
    birincisayi=ikincisayi;
    ikincisayi=ucuncusayi;
    fprintf('%d\n',ucuncusayi);
end
```



Görüntü İşleme (Temel Matlab)

Ekrandan girilen bir sayının faktöriyelini bulunuz. Eğer sayı negatif ise uyarı verecek.

```
clc;clear;
sayi=input('Bir sayı giriniz: ');
fact=1;
if sayi<0
fprintf('Negatif sayı girdiniz!\n');
else
    for i=1:sayi
        fact=fact*i;
    end
    fprintf('Sonuç=%d\n',fact)
end
```

```
clc;clear;
sayi=input('Bir sayı giriniz: ');
tic
fact=1;
if sayi<0
fprintf('Negatif sayı girdiniz!\n');
else
    fprintf('Sonuç=%d\n',factorial(sayi));
end
toc
```



Görüntü İşleme (Temel Matlab)

İç içe geçmiş döngüler (Nested Loops)

```
for x = 1:3  
    for y = 1:2  
        fprintf('x= %.0f and y= %.0f\n',x,y)  
    end  
end
```

```
x= 1 and y= 1  
x= 1 and y= 2  
x= 2 and y= 1  
x= 2 and y= 2  
x= 3 and y= 1  
x= 3 and y= 2
```




Görüntü İşleme (Temel Matlab)

Bir matrisi satır vektöre dönüştüren döngüyü oluşturunuz.

```
[m n] = size(A);  
k=1;  
for i=1:m  
    for j=1:n  
        v(k) = A(i,j);  
        k = k+1;  
    end  
end
```



Görüntü İşleme (Temel Matlab)

Bir matrisi satır vektöre dönüştüren döngüyü oluşturunuz.

```
M=magic(6);  
[r c] = size(M);  
V = [ ];  
for col = 1:c  
for row = 1:r  
V(end+1) = M(row,col);  
end  
end  
disp(V)
```



Fonksiyon yapıları

M-Fonksiyonlar kullanılırken dikkat edilecek hususlar:

- 1- Kullanıcılar kendi fonksiyonlarını yazmak için m-fonksiyonlarını kullanabilirler.
- 2- **Function** alt programı ve ana program şeklinde iki program yazılarak bu iki program ayrı ayrı kaydedilir.
- 3- Alt programdaki *fonksiyon_adi*, m-dosyasına verilen isimle aynı olmalıdır.
- 4- Ana programdan alt program, function adı kullanılarak çağrılır.
- 5- Alt programdan da ana programa geçiş yapılabilir fakat genelde tercih edileni tersidir.
- 6- Parametre aktarımı olması durumunda alt ve ana programda eşit sayıda parametre ve giriş değişkeni olmalıdır.

function cikis_ifadesi1, 2,..., n =**fonksiyon_adi** (giris_ifadesi1, 2, ...n)



Fonksiyon yapıları

Örnek: İki nokta arasındaki uzaklığı bulan programı m-fonksiyon (alt program) kullanarak yazınız.

x1=1.noktanın x koordinati; x2=2.noktanın x koordinati

y1=1.noktanın y koordinati; y2=2.noktanın y koordinati

FUNCTION ALT PROGRAMI (uzak.m):

```
function uzaklik = uzak(x1,y1,x2,y2)  
uzaklik=sqrt((x2-x1).^2+(y2-y1).^2);
```

Bu function alt programı uzak.m olarak kaydedilir.

ANA PROGRAM:

```
ax=3; ay=4; bx=1; by=2;
```

```
uzaklik = uzak(ax,ay,bx,by); % uzak.m alt programını çağırıyor
```

```
fprintf('iki nokta arasındaki uzaklık=%f',uzaklik);
```



Fonksiyon yapıları

Adım adım gerçekleştirilen işlemler:

- Ana program herhangi bir isimle kaydedilir ve koşturulur.
- Program, **function** adına (**uzak**) geldiği zaman alt program çağrılır ve ax , ay , bx , by parametreleri sırasıyla $x1$, $y1$, $x2$, $y2$ giriş değişkenlerine aktarılır.
- **Function** alt programında hesaplama gerçekleştirilir.
- **Function**'daki çıkış değişkeni olan **uzaklik** hem alt programda hem de ana programda hesaplanan sonuç değerinin aktarıldığı değişken olarak kullanılır.
- Alt programdan ana programa parametre aktarımı zorunlu değildir. İstenirse değişkenlerin değerleri alt programda da girilebilir ve sonuç alt programda yazdırılabilir.



Fonksiyon yapıları

Uygulama: Yukarıdaki örneği ana programdan alt programa parametre aktarımı yapmadan yeniden yazınız.

(Değişkenlerin girilmesi, sonucu hesaplama ve yazdırma işlemi alt programda yapılacaktır)

FUNCTION ALT PROGRAMI:

```
function uzaklik = uzak(x1,y1,x2,y2)  
x1=3; y1=4; x2=1; y2=2;  
uzaklik=sqrt((x2-x1).^2+(y2-y1).^2);  
fprintf('iki nokta arasindaki uzaklık=%f', uzaklik);
```

ANA PROGRAM:

```
uzaklik = uzak(ax,ay,bx,by); % uzak.m alt programını çağırıyor
```



Fonksiyon yapıları

Rastgele sayı üreten bir fonksiyon yazınız.

```
function r = RastgeleSayi( altSinir, ustSinir, satir, sutun )  
r = round(altSinir + (ustSinir-altSinir).*rand(satir,sutun));  
% random neagtif, pozitif sayýlar  
  
end
```



Fonksiyon yapıları

```
function [ maks, minimum ] = MaxMinBul( vektor )
```

```
maks=max(vektor);  
minimum=min(vektor);  
end
```

```
>> j=RastgeleSayi(-40,50,1,20);  
>> [maksimum, minimum]=MaxMinBul(j)
```

```
maksimum =
```

```
45
```

```
minimum =
```

```
-29
```




Fonksiyon yapıları

```
function y = NormalizeEt01(x)
%b alt sınır, a ust sınır
[satir,sutun]=size(x);
i=1; j=1;
for i=1:satir
    for j=1:sutun
        y(i,j)=[(x(i,j)-min(x(:)))]/[max(x(:))-min(x(:))];
    end
end
```



Fonksiyon yapıları

Aşağıda verilen y fonksiyonunu hesaplayan bir matlab fonksiyonu yazınız.
X ve n değerlerinin girişleri dışarıdan verilecektir.

Fonksiyon
>> **bToplam(x,n)**
Şeklinde çalıştırılacaktır.

$$y = \sum_{k=1}^n \left(\frac{2}{x}\right)^k$$