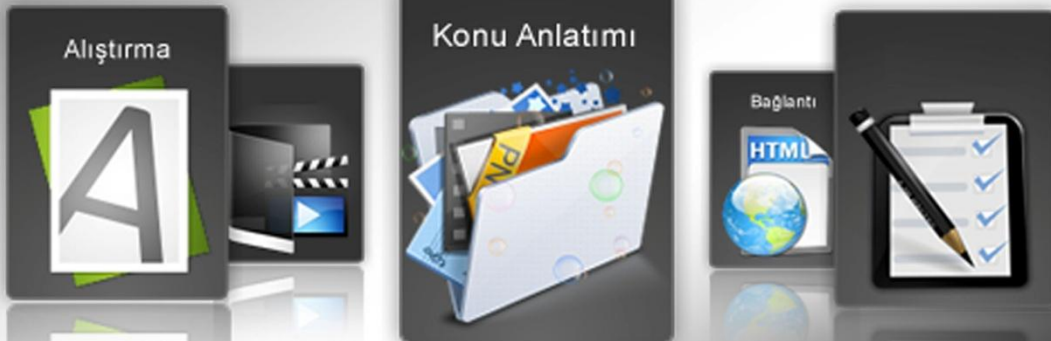


Matlab **R2012a** ile

AZ VE ÖZ MATLAB



Muhammet ÇAĞATAY

DÜZENLEME : 11

22.12.2013

MATLAB HAKKINDA GENEL BİLGİLER.....	3
MATLAB VE DEĞİŞKENLER.....	3
DEĞİŞKEN TANIMLAMA KURALLARI	3
OPERATÖRLER.....	4
MATEMATİKSEL FONKSİYONLAR.....	4
DEĞİŞKEN TİPLERİ.....	4
SAYI FORMATLARI.....	5
ÖZEL DEĞER VE FONKSİYONLAR.....	5
BİR DEĞİŞKENE DIŞARIDAN DEĞER ATAMA.....	5
DİZİLER.....	6
MATLABDA VEKTÖR OLUŞTURMAK.....	6
DİZİ İŞLEMLERİNDE KULLANILAN TEMEL FONKSİYONLAR.....	7
MATLABDA PROGRAMLAMA.....	8
SONUÇ YAZDIRMA (ÇIKTI VERME)	9
KARŞILAŞTIRMA OPERATÖRLERİ	9
İF KOŞUL YAPISI	10
ELSE İF KOŞUL YAPISI	12
FOR DÖNGÜSÜ	13
İÇ İÇE DÖNGÜLER	15
WHILE DÖNGÜSÜ.....	16
BREAK VE CONTINUE DEYİMİ.....	16
SWITCH – CASE KARAR YAPISI.....	18
TRY – CATCH YAPISI.....	19
PROGRAMI SONLANDIRMAK (RETURN VE ERROR).....	20
MATLABDA MATRİS İŞLEMLERİ.....	20

ÖRNEK UYGULAMALAR 1.....	24
YER DEĞİŞTİRME ALGORİTMASI.....	28
SIRALAMA ALGORİTMASI.....	29
MATLABTA FONKSİYON OLUŞTURMAK VE PROGRAMA ÇAĞIRMAK.....	31
KABARCIK SIRLAMA (BUBLE SORT) ALGORİTMASI.....	35
MATLABDA KULLANILAN BAZI YUVARLAMA VE STRİNG FONKSİYONLARI.....	37
MATLABDA TABAN ARİTMETİĞİ İŞLEMLERİ.....	42

Matlab; c , c++ , fortran gibi gelenekselleşmiş dillerin aksine , matrislere dayalı olarak, bilim adamları ve mühendislere problem çözmeye yardımcı olan, yüksek performanslı bir paket programıdır.

1. Matris laboratuvarı olarak da isimlendirilir.
2. Matlab bir yorumlayıcıdır yani sonuç bir hesap makinesi tarzında ekrana yazılı bir metin olarak döndürülür.
3. Diğer dillerde olduğu gibi bir derleyiciye (decompiler'e) ihtiyaç yoktur.
4. Matlab açılımı İngilizce **"MATrix LABoratory"**den gelmektedir.

Matlab kullanarak;

1. Matematiksel hesaplamalar,
2. Algoritma geliştirme ve kod yazma (programlama),
3. Grafiksel arayüz (GUI) oluşturma,
4. Modelleme ve simülasyon (benzetim),
5. Resim işleme, 2D ve 3D grafiklerinin çizimi
6. Veri analizi,
7. Lineer cebir, istatistik, Fourier analizi, filtreleme, optimizasyon, sayısal integrasyon vb. konularda hazır matematik fonksiyonlara ulaşım,
8. Gerçek dünya şartlarında uygulama geliştirme kolaylıkla yapılabilir.

MATLAB VE DEĞİŞKENLER

1. Matlab'da diğer programlama dillerinden farklı olarak her şey bir matris olarak düşünülür.
2. Değişkenler matlab deyimleri içinde sayısal değerlerin yada metinlerin (string) yerini alan ifadelerdir.
3. Matlab, bir değişken ile karşılaştığında otomatik olarak değişken oluşturur ve yeterli kadar bellek ayırır. Eğer değişken daha önce zaten oluşturulmuş ise yalnızca içeriği değiştirilir ve gerekiyorsa yeni bellek ayırır.
4. Diğer dillerde olduğu gibi matlabın da değişkenler konusunda katı kuralları vardır.

DEĞİŞKEN TANIMLAMA KURALLARI

1. Değişken isimleri büyük küçük harf kullanımına duyarlıdır. Yani **orta** , **Orta** , **ORTA** , **oRTa** dördü de farklı değişkenlerdir.
2. Değişken isimleri 63 karakter içerebilir, 63 ten sonrasını derleyici dikkate almaz.
3. Değişken isimleri rakam ile başlayamaz. Örnek **1sayı** yerine **sayı1** demelisiniz.
4. Değişkenlerin arasında boşluk kullanılmaz. **sayı 1** yerine **sayı_1** kullanabilirsiniz.
5. Değişken tanımlanırken Harf, rakam ve alt çizgi " _ " den başka karakter kullanılmaz.
6. Değişken tanımlanırken Türkçe karakterler (ç , ğ , ı , ü , ö , ş) kullanılmaz.
7. Matlabda rakamlar için + veya - işareti ve ondalık noktası (.),
Sanal rakamlar için i veya j
Kuvvet belirtmek için ise e harfi kullanılır (örnek : $e^2 = 10^2 = 100$)

OPERATÖRLER

toplama	+	2+3
çıkarma	-	2-3
çarma	*	2*3
bölme	/	2/3
Üst alma	^	2^3
parantez	()	2*(5+3)

&& = ve operatörü	= veya operatörü
-------------------	------------------

Not 1 : toplam = toplam + 1 toplam değerini 1 arttırır. toplam++ diye bir ifade yoktur !

Not 2 : Matlabda matematikteki işlem öncelik sırası aynen geçerlidir. (yukarıdaki tabloya göre alltan üste doğudur.)

MATEMATİKSEL FONKSİYONLAR

Fonksiyonlar matlabta temel programlama araçlarıdır ve matlabın kendi hazır fonksiyonları bulunduğu gibi, kullanıcı endi fonksiyonunu kendisi de yazabilmektedir.

sinüs	sin	sin(x)
kosinüs	cos	cos(pi)
tanjant	tan	tan(pi)
arcsinüs	asin	asin(pi)
Üstel (e ² , e ^x)	exp	exp(2)
Doğal logaritma ln (log e ¹⁰)	log	log(10)
10 tabanlı logaritma	log10	log10(10)
karekök	sqrt	sqrt(b)
Mutlak değer	abs	abs(3)
Log ₅ (x) =	Log10 (x) / log10 (5) Taban değiştirme	

Not 3 : Trigonometrik fonksiyonlar açı değerlerini radyan cinsinden alırlar, dereceye çevirmek için $\frac{d}{180} = \frac{\text{radyan}}{\pi}$ formülü kullanılır.

Örnek 1 : $ab - c + d - b + da = a*b - c + d - b + d*a$

Örnek 2 : $a/b + \sqrt{c^3} - b * d^2 + 2ab / b^2 = a/b + \text{sqrt}(c^3) - b * (d^2) + (2*a*b) / (b^2)$

MATLABDA DEĞİŞKEN TİPLERİ

Matlabda bir değişen tipi matris, vektör (satır veya sütün) , string (karakter dizisi) , skaler (numerik yada karakter) lerden birisi olabilir.

Bu tiplerin her biri, bir matris olarak düşünülebilir, her değişkenin 2 tane parametresi vardır.

Double : skaler yada vektörlerden oluşan sayısal değişkenlerdir.

Char : tek bir karakter veya karakter grubundan oluşan değişkendir.

MATLABDA SAYI FORMATI

Matlab bir işlemin sonucunu varsayılan olarak 4 ondalık ile gösterir ve bu gösterim isteğe göre **>>format** fonksiyonu ile değiştirilebilir. **Örnek sayı = 15.973**

>> format long	15.973000000000001	Virgülden sonra 15 basamak yazdırır.
>> format bank	15.97	Virgülden sonra 2 basamak yazdırır.
>> format rat	15973/1000	Sonuç rasyonel sayı olarak döner.
>> format yada >> format short	15.9730	Virgülden sonra 4 basamak yazdırır. Varsayılan değere dönmek için kullanılır.

Bir sayının istenilen hanesinin **gösterilmesi** için **fprintf** veya **sprintf** komutları kullanılır:

```
>>a=100004545.99923423499111;  
>>fprintf('%1.10f',a)  
  
100004545.9992342300
```

Matris elemanlarının istenilen hanede gösterilmesi için, **printmatrix** fonksiyonu oldukça kullanışlıdır (File-exchange sayfasından download edilebilir!)

MATLABDA ÖZEL DEĞER VE FONKSİYONLAR

Matlabın yapısında hazır tanımlanmış ve kullanıma hazır, birebir değerler ve özel fonksiyonlar vardır. Bunlardan bazıları:

>> pi	Pi sayısını gösterir.
>> date	Tarihi verir.
>> clock	Saati verir
>> fix	Yuvarlama fonksiyonu.
>> clc	Ekranı temizler. (değişkenleri silmez)
>> who	Tanımlanan değişkenleri gösterir.
>> whos	Tanımlanan değişkenleri ve özelliklerini gösterir.
>> clear	Tanımlanan tüm değişkenleri siler.
>> tic ve >>toc	Matlabda bir işlemin ne kadar zaman aldığını gösterir.
>> help xx	xx komutu hakkında yardım almak için kullanılır. Bunun yerine xx yazıp klavyeden F1 tuşuna basmak daha kolay olur.

BİR DEĞİŞKENE DIŞARIDAN DEĞER ATAMA

input fonksiyonu matlaba dışarıdan değer ataması için kullanılır ve kullanıcı bir değer girene kadar matlab işleme devam etmez.

```
>> sayi = input(' bir sayi giriniz ');  
>> yazi = input (' bir metin giriniz ', 's' );
```



string = metin

MATLABDA DİZİLER

Dizi (array) : en genel tanımı ile numerik ya da metinsel değerler topluluğudur. Matlabda her şey matris olduğu için diziler de matris olarak düşünlür. Dizi en temel veri elemanıdır.

1 x 1 dizisi bir skaler gösterir.	m x 1 dizisi bir vektör gösterir.
1 x n dizisi de bir vektör gösterir.	m x n dizisi bir matris gösterir.

Bir dizinin eleman sayısı satır ile sütun sayısının çarpımıdır.

Bir dizinin herhangi bir elemanı satır ile sütun numaraları ile indekslenir. [**A (2,4)**]

MATLABDA VEKTÖR OLUŞTURMA

Matris ve vektörler [] köşeli parantezleri ile tanımlanır.

Matlabda vektör tanımlamanın 3 yolu vardır;

1 - DİREKT OLARAK:

Vektor1 = [1 2 3]; yada **vektor1 = [1, 2, 3];** = satır vektör

Vektor2= [1 ; 2 ; 3]; = sütun vektör

Örneğin:

$$A = \begin{bmatrix} 1 & 3 & 5 \\ 7 & 8 & 11 \\ 100 & 1 & 4 \end{bmatrix}$$

1.yol

```
A=[1 3 5  
7 8 11  
100 1 4]
```

2.yol

```
A=[1 3 5;7 8 11;100 1 4]
```

3.yol

```
A(1,1)=1,    A(1,2)=3,  A(1,3)=5  
A(2,1)=7,    A(2,2)=8,  A(2,3)=11  
A(3,1)=100,  A(3,2)=1,  A(3,3)=4
```

2 – EŞİT ARALIKLI ELEMANLAR KULLANARAK:

F= ilk değer : değişim miktarı : son değer

Örnek : F = 1 : 2 : 99 (Değişim miktarı belirtilmedi ise 1 olarak alınır)

3 - ELEMAN SAYISI VE ADEDİ BELİRTEREK:

İnpace komutu ile bir aralık doğrusal parçalara ayrılır. Eğer iki parametrelili kullanılırsa 100 adet parça oluşturulur.

inspace (a , b , n) ifadesi **a ve b arasında n tane parça oluşturur.**

DİZİ İŞLEMLERİNDE KULLANILAN TEMEL FONKSİYONLAR

Vektör ve matrisler aşağıdaki fonksiyonlar yardımı ile oluşturulabilir.

>> zeros(1,n)	>> zeros(n,1)	>> zeros(n)	>> zeros(n,n)
>> ones(1,n)	>> ones(n,1)	>> ones(n)	>> ones(n,n)

>> Rand (m , n) veya **Rand (n)** = 0 ile 1 arasında değişen rastgele sayı değeri üretir.

Not : Rand fonksiyonu ile oluşturulan sayı değerleri **>> round** fonksiyonu ile tam değere yuvarlanabilir. 0,5 ten büyük olanların sonucu **1** , küçük olanların sonu ise **0** olur.

Örnek : 0 ile 10 arasında 5 tane sayı üretmek için **>> round (10 * rand (1,5))** komutu kullanılır.

>> floor (a) = a sayısını tam değer olana yuvarlar. **Örnek :** 2,87 yi 2 yapar

>> lenght (a) = a vektörünün eleman sayısını verir.

>> size (a) = a matrisinin boyutunu verir.

>> size (a,1) Satır sayısını ve **>> size (a,2)** sütun sayısını verir.

>> sum (a) = a vektörünün elemanlarının toplamını verir. Eğer bu fonksiyon bir matrise uygulanırsa her bir sütundaki elemanların toplamını verir.

>> min (a) ve **>> max (a) =** a vektörünün minimum ve maksimum elemanlarını verir. Bu fonksiyon bir matrise uygulanırsa her bir sütundaki minimum ve maksimum elemanları buluruz.

>> det (a) = a matrisinin determinantını verir. (karesel matrisler için geçerlidir)

>> inv (a) = a matrisinin tersini verir.

>> diag (a) = a matrisinin çapraz elemanlarını verir.

>> num2str(x) = Sayıyı metinsel ifadeye çevirir.

MATLABDA PROGRAMLAMA

Bilgisayar programları yazılırken genelde belli bir kalıp ve kurallara uyulur. Bunlar ;

1. Problemin belirlenmesi (ne için)
2. Çözüm yönteminin ve adımlarının belirlenmesi (algoritma ve çakma kod)
3. Kodlama (programlama diline çevirme)
4. Sınama (test)
5. Güncelleme

Matlabda programlama 2 yolla yapılır;

1. Komut satırında
2. m dosyaları ile
 - scripts (düz yazı m dosyaları)
 - function (fonksiyon olan m dosyaları)

senaryo olan (script) düz yazı m dosyası, özel bir görevi yerine getirmek için gerekli matlab komutlarının saklandığı bir metin dosyasıdır.

Bu dosyayı oluşturmak ve matlab komutlarını yazmak için editör kullanılmalıdır.

Diğer dillerin aksine matlabın kendi editörü bu iş için çok kullanışlıdır.

- Değişken sayısının fazla olduğu durumlarda,
- Akış diyagramı uygulamalarında,
- Ve Giriş , çıkış işlemlerinde, m dosyalarını kullanmak gereklidir.

Örnek : dışarıdan girilen yarıçap değerine göre dairenin alan ve çevresini hesaplayan program ;

```
clc;
r = input(' yarıcap giriniz : ');
alan = pi * r * r;
cevre = pi * 2 * r;
fprintf(' alan : %f, \n cevre : %f \n ', alan, cevre);
```

MATLABDA SONUÇ YAZDIRMA (ÇIKTI VERME)

Matlabda ekrana bir ifadeyi yazdırmak için **disp** yada **fprinf** komutları kullanılır.

Disp komutu genellikle **disp(toplam)** ; yada **disp(' merhaba ')** ; gibi düz bir ifadeyi yada sonucu göstermek için kullanılır.

fprinf() ; komutu daha kullanışlıdır ve bu komut kullanılırken ;

\n = işlemi yeni (alt) satıra atlatır.

\t = Bir tab boşluk bırakır.

%s = String (metinsel) değişken.

%d = Tam sayıları yazdırmak için.

%f = Ondalık olarak 6 hane şeklindeki reel sayıları yazdırmak için.

%e = Üstel bir biçimde ondalıklı sayıları yazdırmak için.

%g = %f ile %e arasında kompakt bir biçimde seçim yapılarak gösterilmesini sağlar. Ondalık 5 basamak olarak sonuç verir.

Örnek kullanım :

```
fprinf ('sayilarin ortalamasi = %f ',ort);  
sayilarin ortalamasi = 8.000000 >>
```

Örnek : Kullanıcıdan 3 tane sayı isteyip toplayıp ortalamasını bulan programı yazınız.

```
clc;  
a=input(' 1. sayiyi giriniz ');  
b=input(' 2. sayiyi giriniz ');  
c=input(' 3. sayiyi giriniz ');  
ort=(a+b+c)/ 3;  
fprinf ('sayilarin ortalamasi = %f', ort);
```

KARŞILAŞTIRMA OPERATÖRLERİ

== : Eşit midir ?
> : Büyük müdür ?
< : Küçük müdür ?
<= : Küçük eşit midir ?
>= : Büyük eşit midir ?
~= : Eşit değil midir ?

MATLABDA KONTROL İFADELERİ

1. **IF DEYİMİ** : Şartlı dallanma adı verilen işlemi gerçekleştirir.

if Şart

1. işlem;
2. işlem;
3. işlem;

end

if Şart

1. işlem;
Else
2. işlem;
end

Örnek : Klavyeden girilen bir sayının tamsayı olup olmadığını araştırınız.

```
clc;  
sayi = input(' bir sayi giriniz : ');  
if (round(sayi)- sayi)==0  
    fprintf('sayı tamdır');  
else  
    fprintf('sayı tam değildir');  
end
```

Örnek : kullanıcıdan x isteyip sonucu fonksiyonlara göre bulan programı yazınız.

$$f(x) \begin{cases} x > 0 & f(x) = \frac{1}{(1+\ln x)} & g(x) = \frac{1}{(x+\ln x)} \\ x \leq 0 & f(x) = \frac{1}{(x+x^2)} & g(x) = \frac{1}{(1+x+x^2)} \end{cases}$$

```
clc;  
x = input(' x degeri giriniz : ');  
if ( x <=0 )  
    f=1/(1+x^2);  
    g=1/(1+x^2+x);  
else  
    f=1/(1+log(x));  
    g=1/(x+log(x));  
end  
fprintf(' f : %g, \n g : %g \n ',f,g);
```

Örnek : bir uçak firması satıcının günlük ücretini şöyle hesaplamaktadır.

- Günlük satış adedi 50 den az ise 40 tl sabit ücret ve satılan ürün başına 80 tl prim vermektedir.
- Günlük satış miktarı 50 veya daha fazla ise 100 tl ve sattığı ilk 50 ürün başına 90 tl, 50 den sonraki her ürün için 150 tl prim vermektedir.
- Adedi kullanıcıdan isteyip satıcının günlük alacağı ücreti hesaplayan matlab programını yazınız.

```
clc;
adet = input(' satış adedi giriniz : ');
if ( x < 50 )
    maas = 40 + ( adet * 80 );
else
    maas = 100 + ( 50 * 90 ) + ( (adet -50) * 150 );
end
fprintf(' maas : %g \n ',maas);
```

Tek bir if deyimi programın iki seçenektan birini seçmek için kullanılır. Fakat birçok program için tek bir if deyimi yetersiz kalır ve bu gibi durumlarda iç içe **if** ya da **if – else if** yapısı kullanılır. İç içe if kullanılırken dikkat edilmesi gereken husus bir if içerisinde yalnızca bir tane ele kullanılmasıdır. Bir else kendinden önce gelen en yakın if ile ilişkilidir.

Örnek : Girilen üç adet sayının en küçüğünü bulan programı yazınız.

```
clc;
a = input(' sayı 1 giriniz : ');
b = input(' sayı 2 giriniz : ');
c = input(' sayı 3 giriniz : ');

if ( a < b )
    if( a < c)
        fprintf(' en küçük sayı : %d \n ',a);
    else
        fprintf(' en küçük sayı : %d \n ',c);
    end
else
    if( b < c)
        fprintf(' en küçük sayı : %d \n ',b);
    else
        fprintf(' en küçük sayı : %d \n ',c);
    end
end
```

2. **ELSE - İF DEYİMİ** : Birbirini izleyen çok sayıda koşulun kontrolünü gerçekleştiren işlemlerde **if – else if** yapısı kullanılır.

if Şart

1.işlem;

Elseif Şart

2.işlem;

else

3.işlem;

end

Örnek : 0 ile 10 arasında rastgele üretilen bir sayıyı bulmak için kullanıcıdan sayı isteyen ve girilen sayının üretilen sayıdan büyük mü, küçük mü yada eşit mi olduğunu bulan matlab programını yazınız.

```
clc;
uretilen = round ( 10 * rand (1,1));
girilen = input(' tahminizi giriniz : ');

if ( uretilen < girilen )

    fprintf(' daha küçük bir sayı giriniz ');

elseif ( uretilen > girilen )

    fprintf(' daha büyük bir sayı giriniz ');

else

    fprintf(' tebrikler buldunuz ');

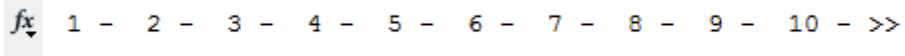
end
```

3. **FOR DÖNGÜSÜ** : başlangıç, bitiş ve adım değerleri arasında dönen bir dögüdür. Eğer adımlarınız birer birer artıyor ise adım değerini yazmasanız da olur.

```
for parametre = başlangıç değeri : artım değeri : bitiş değeri
    olay1
    olay2
    ....
end
```

Örnek : 1 den 10 a kadar olan sayıları ekrana yazdıran programı yazınız.

```
clc;
for i=1:1:10
    fprintf(' %d - ',i);
end
```

EKRAN ÇIKTISI : 

Örnek : Kullanıcıdan istenen bir sayının faktöriyelini hesaplayan matlab programını yazınız.

```
clc;
sayi =input(' bir sayi giriniz : ');
if(sayi>0)
    faktoriyel =1;    & çarpım olduğundan ilk değeri 1 aldık
    for i=1:1:sayi
        faktoriyel = faktoriyel * i;
    end
    fprintf(' %d ',faktoriyel);
else
    fprintf(' sayınız 1 den küçük olamaz ');
end
```

Örnek : $m = 100000$ olmak üzere, $4 \sum_{k=0}^m \frac{(-1)^2}{2k+1} = ?$ Soncunu hesaplayan matlab programını yazınız.

```
clc;
toplam =0;
for i=0:1000000
    sonuc=(((-1)^i))/((2*i)+1);
    toplam=toplam+ (4* sonuc);
end
fprintf(' toplam = %g' ,toplam);
```

Örnek : Bilinen bir alt değişkenden üst değışkene kadar olan sayıların ardışık toplamını ekrana yazdıran matlab programı yazınız.

```
clc;
toplaml=0;
bas=input('bir başlangıç değeri giriniz : ');
son=input('bir bitiş değeri giriniz : ');
if bas>son
    fprintf('başlangıç değeri bitiş değerinden büyük olmamalı !');
else
    for i=bas:1:son
        toplaml=toplaml+i;
    end
    fprintf('toplaml : %d ',toplaml);
end
```

Örnek : Klavyeden girilen bir sayı için, eşitliliğini kontrol eden matlab programını yazınız.

$$\prod_{k=1}^n \left(1 - \frac{1}{k^2}\right) = \frac{n+1}{2n}$$

```
clc;
carpim=1;
n = input(' bir sayi giriniz : ');
for k =1:n
    carpim = carpim * (1 - ( 1 / (k^2)));
end
sonuc = (n+1)/(2*n);
if carpim==sonuc
    fprintf('girdiğiniz sayi eşitliği sağlıyor');
else
    fprintf('nalesef girdiğiniz sayi eşitliği sağlamıyor');
end
```

Örnek : Küpü 4000 den büyük ve 3 ile bölünebilen en küçük doğal sayıyı bulan programı yazınız.

```
clc;
sayi=1;
x=1;
while x==1;

    if ( (sayi^3)>=4000 && mod(sayi,3)==0 )
        fprintf('\n aranan rakam : %d ',sayi);
        x=0;
    else
        fprintf('olmadı : %d ',sayi);
        sayi=sayi+1;
    end
end
```

İÇ İÇE DÖNGÜLER

Algoritma geliştirirken bir döngü içerisine bir veya daha fazla döngü kullanmak gerekebilir. Bu gibi durumlarda dışarıdaki döngünün bir kez dönmesi sonucu içerdeki döngü tam döngüsünü tamamlayacaktır. Yani ilk olarak içerdeki döngü tamamlanacaktır.

Örnek : Resimdeki gibi 10 x 10 çarpım tablosunu ekrana yazdıran matlab programını yazınız.

	1	2	3	4	5	6	7	8	9	10
1	1	2	3	4	5	6	7	8	9	10
2	2	4	6	8	10	12	14	16	18	20
3	3	6	9	12	15	18	21	24	27	30
4	4	8	12	16	20	24	28	32	36	40
5	5	10	15	20	25	30	35	40	45	50
6	6	12	18	24	30	36	42	48	54	60
7	7	14	21	28	35	42	49	56	63	70
8	8	16	24	32	40	48	56	64	72	80
9	9	18	27	36	45	54	63	72	81	90
10	10	20	30	40	50	60	70	80	90	100

```
clc;
fprintf('      %3d %3d %3d %3d %3d %3d %3d %3d %3d %3d %3d\n',1,2,3,4,5,6,7,8,9,10);
fprintf('-----\n');
for i=1:10
    fprintf('%2d |',i);
    for j=1:10
        fprintf('%3d ',i*j);
    end
    fprintf('\n');
end
```

Örnek : Aşağıdaki işlemin sonucunu ekrana yazdıran matlab programını yazınız.

$$\frac{1}{2}! + \frac{2}{3}! + \frac{3}{4}! + \dots + \frac{14}{15}!$$

```
clc;
toplam=0;
for i=2:15
    faktoriyel=1;
    for j=1:i
        faktoriyel=faktoriyel*j;
    end
    sonuc = (i-1)/faktoriyel;
    toplam =toplam+sonuc;
end
fprintf('sonuç : %g |', toplam);
```


WHILE DÖNGÜSÜ

1. Belirlenen şart sağlandıkça dönen döngü tipidir.
2. Bu döngüde dikkat edilmesi gereken nokta şart sağlandıkça döngünün sürekli döneceğidir. Eğer önlem alınmaz ise sonsuz döngüye sebep olur.
3. While ve for döngüleri birbirinin yerine kullanılabilir, doğru olanı seçmek bizim görevimizdir.

while şart
şart sağlandıkça olaylar
end

BREAK DEYİMİ

1. Bir döngü yada switch case yapısını sona erdirmek için kullanılan komuttur.
2. Matlab bu komutu gördüğünde içinde bulunduğu döngüyü sonlandırır ve bu komutun altındaki komutları değerlendirmez.

CONTINUE DEYİMİ

Döngüyü sonraki çevirmeye geçirir. Bu komutun altındaki komutlar o anki çevirme için değerlendirilmez.

Örnek : Continue kullanarak 1 ile 100 arasındaki tek sayıları ekrana alt alta yazdıran matlab programını yazınız.

```
clc;
for i=1:100
    if mod(i,2)==0
        continue
    end
    fprintf(' %d \n ',i);
end
```

Örnek : 10 tl nin yıllık yüzde 30 faiz ile, 50 tl yi geçtiği ada eşit olduğu ilk yılı ve toplam para değerini bulan bir matlab programı yazınız.

```
clc;
toplam_para=10;
yil =0;
while(toplam_para <50)
    yuzde = toplam_para * 3 /10;
    toplam_para = toplam_para + yuzde;
    yil =yil +1;
end
fprintf(' ay : %d para : %g \n ',yil, toplam_para);
```

Örnek : kullanıcı negatif sayı girinceye kadar yeni sayı isteyen ve girilen sayıları toplayan bir matlab programı yazınız.

```
clc;
toplama=0;
while(1)
sayi = input('sayi giriniz = ');
    if( sayi < 0)
        break;
    end
    toplama=toplama + sayi ;
end
fprintf('toplama = %g ', toplama);
```

Örnek : x=10 ve y=7 olmak üzere, bu sayıların toplamı 52757 den büyük oluncaya kadar, x sayısını 2 kez katlayarak toplama işlemine işleme devam ediliyor. Buna göre x sayısı kaç kez katlanmıştır.

```
clc;
x=10;
y=7;
toplama = x+y;
katlanma =0;
while(toplama<52757)
    x = 2 * x;
    toplama = x+y;
    katlanma =katlanma +1;
end
fprintf(' ay : %d ', katlanma);
```

Örnek : Aşağıdaki ekran çıktısını veren matlab programını yazınız.

```
1
j= 1 , j= 2 , j= 3 ,
2
j= 1 , j= 2 , j= 3 ,
3
j= 1 , j= 2 , j= 3 ,
4
j= 1 , j= 2 , j= 3 ,
5
j= 1 , j= 2 , j= 3 ,
```

```
clc;
for i=1:5
    fprintf('\n %d \n', i);
    for j=1:3
        fprintf('j= %d , ', j);
    end
end
```

SWITCH – CASE YAPISI

If yapısının bir benzeri olan bu yapıda anahtar değere karşılık gelen birçok seçenek içinden uygun olanı seçmelisiniz. Bu yapı ile yapılan her algoritma if ile de yapılır. Kullanımı ;

Switch (durum)

case (durum1)

işlemler

case (durum2)

işlemler

.

.

otherwise

► *isteğe bağlı*

end

Örnek : Verilen ay değerine karşılık, o ayın kaç günden oluştuğunu hesaplayan matlab programını yazınız.

```
clc;
ay = input('ay giriniz');
switch ay
    case {1,3,5,7,8,10,12}
        fprintf(' 30 gün ');
    case 2
        yıl = input('yıl giriniz');
        if(mod(yıl,4)==0)
            fprintf(' 29 gün ');
        else
            fprintf(' 28 gün ');
        end
    case {4,6,9,11}
        fprintf(' 30 gün ');
    otherwise
        fprintf(' hatalı ay girişi ');
end
```

Örnek : kullanıcının gireceği bir düzgün çokgen için, iç açılar toplamını hesaplayan bir matlab programı yazınız.

```
clc;
cokgen = input('bir çokgen giriniz : ','s');

switch cokgen
    case 'ucgen'
        kenar_sayisi=3;
    case { 'dortgen', 'dörtgen','4gen' }
        kenar_sayisi=4;
    case 'beşgen'
        kenar_sayisi=5;
    otherwise
        fprintf(' geçersiz giriş \n ' );
end
sonuc = (kenar_sayisi - 2)*180;
fprintf('sonuc : %d \n ',sonuc );
```

TRY – CATCH YAPISI

Mantıksal hatayı kontrol eder (yazı hatalarını kontrol etmez).

İf yapısına benzer şekilde dallanma sağlanır.

İki durumda hangisi mantıksal olarak doğru ise onu çalıştırır.

İkisi de doğru ise try kısmı çalışır.

```
try
    durum1
catch
    durum2
end
```

Örnek :

```
clc;
a=10; %başlangıç değeri 10
try
    a=a+20; % a nın yeni değeri 30 oldu
    a=a+g; % g diye bir sayı olmaz ( mantık hatası ) bu satırı
atlar
catch
    a=50; % yeni değeri 50
end
display(a); % sonucu 30 yerine 50 yazar , çünkü 2. satırı
atlar
```

MATLABDA PROGRAMI SONLANDIRMAK

Retrun : Yazdığımız matlab programlarının herhangi bir anda (*programın normalde sona erdiği noktadan daha önce*) sonlandırılması için kullanılır. Bir diğer deyiş ile programı durdurup kapatma komutudur.

Error : Returna benzer bir komut olan error komutunda, program bir hata ile birlikte sonlandırır. Return komutunda ise herhangi bir hata durumu yoktur.

MATLABDA MATRİS İŞLEMLERİ

Doğal olarak matrisler ile ilgili matlabda birçok hazır fonksiyon (metod) vardır.

Bu derste matlab Hazır fonksiyonlarından bazılarının algoritmalarını anlamaya çalışacağız.

Örnek : 2 x 3 Boyutlu Bir Matrisin Elemanlarını, Klavye Yolu İle Kullanıcıdan Alan ve Sonunda Bu Matrisi Ekrana Yazan Matlab Programını Yazınız.

```
clc;
A=[];      % A=[2,3]; de yazılabilir
for i=1:2
    for j=1:3
        fprintf('A(%d,%d) =',i,j);
        A(i,j)=input(' ');
    end
end
A
```

Daha farklı olarak bir matris tek bir seferde de girilebilir. Ancak girilen matrisin istenilen formda olup olmadığı kontrol edilmelidir.

```
clc;
A=[];
fprintf('2 x 3 tipine bi matris giriniz \n' );
A=input('A =');
[x,y] =size(A);
if (x~=2) || (y~=3)
    error('matris yanlış formda \n');
else
    A
End
```

Örnek : kullanıcı tarafından girilen 2 adet matrisin toplamını bulan ve ekrana yazan matlab programını yazınız.

```
clc;
A=input('birinci matrisi giriniz : ');
B=input('ikinci matrisi giriniz : ');
[x,y]=size(A);
[n,m]=size(B);
if (x~=n) || (y~=m)
    error('Toplanamaz !');
else
    T=[];
    for i=1:x
        for j=1:y
            T(i,j)=A(i,j)+B(i,j);
        end
    end
end
T
```

Örnek : kullanıcıdan bir matris ile bir skaler isteyip bunların çarpımını ekrana yazdıran programı yazınız.

```
clc;
A=input('matrisi giriniz : ');
k=input('skaler k değerini giriniz : ');
[x,y]=size(A);
T=[];
for i=1:x
    for j=1:y
        T(i,j)=A(i,j) * k;
    end
end
T
```

Örnek : kullanıcı tarafından girilen matrisin transpozunu ekrana yazdıran matlab programını yazınız.

```
clc;
A=input('A matrisini giriniz : ');
[x,y]=size(A);
T=[];
for i=1:y
    for j=1:x
        T(i,j)=A(j,i);
    end
end
T
```

Örnek : kullanıcı tarafından girilen 2 matrisin çarpımını bulan ve ekrana yazan matlab komutlarını yazınız.

```
clc;
A=input('A matrisini giriniz : ');
B=input('B matrisini giriniz : ');
[x,y]=size(A);
[n,m]=size(B);
toplama =0;
carpim =1;
if(y~=n)
    error('bu matrisler carpilamaz. ');
else
    C=zeros(x,n);
    for i=1:x
        for j=1:y
            for k=1:m
                C(i,j) = C(i,j) + A(i,k) * B(k,j);
            end
        end
    end
end
C
```

Örnek : satır ve sütun sayısı kullanıcı tarafından girilen değerleri 0 ile 15 aralığındaki tamsayılardan rastgele olarak oluşan bir matristeki, sıfırların adresini ve sayısını bulan matlab programını yazınız.

```
clc;
x=input('satır giriniz : ');
y=input('sütun giriniz : ');
A=round(rand(x,y)*15)
sifirlarin_sayisi=0;
for i=1:x
    for j=1:y
        if(A(i,j)==0)
            sifirlarin_sayisi =sifirlarin_sayisi +1;
            fprintf('A(%d,%d) = 0 \n',i,j);
        end
    end
end
fprintf('A matrisindeki 0 larin sayisi : %d \n', sifirlarin_sayisi);
```

Örnek : kullanıcı tarafından girilen bir satır vektörünün elemanlarını tersten yazan başka bir satır vektörü oluşturunuz.

```
clc;
A=input('Matris giriniz : ');
if( size(A,1)~= 1) %size(A,1) satır sayısını verir
    error('malesef bu satır matrisi ( vektör ) değil ');
else
    T=[];
    y=size(A,2); %size(A,2) sütun sayısını verir
    k=y;
    for i=1:y
        T(k)=A(i);
        k=k-1;
    end
end
T
```

Örnek : kullanıcı tarafından girilen bir matrisin yalnızca ilk satırını terse çeviren diğer satırlarını aynen yazan matlab komutlarını yazınız.

```
clc;
A=input('bir Matris giriniz : ');
T=A;
sutun_sayisi=size(A,2);
sayac=sutun_sayisi;
for i=1:sutun_sayisi
    T(1,sayac)=A(1,i);
    sayac=sayac-1;
end
A
T
```

SONUÇ :

A =

2	3	4
5	6	7

T =

4	3	2
5	6	7

Örnek : Kullanıcı tarafından girilen bir matris ile bu matrisin transpozunu çarpan matlab programını yazınız. (Osmangazi Üniversitesi - matematik ve bilgisayar bilimleri – temel bilgi teknolojileri 1 dersi - 2012 vize2 sınav sorusu)

```
clc;
A=input('A matrisini giriniz : ');
[x,y]=size(A);
T=[];
for i=1:y
    for j=1:x
        T(i,j)=A(j,i);
    end
end
C=zeros(x,x);
for i=1:x
    for j=1:y
        for k=1:y
            C(i,j) = C(i,j) + A(i,k) *
T(k,j);
        end
    end
end
A
T
C
```

```
A matrisini giriniz : [1,2;4,5]
A =
     1     2
     4     5
T =
     1     4
     2     5
C =
     5    14
    14    41
>> |
```

ÖRNEK UYGULAMALAR 1

Örnek : $\sum_{k=0}^n \frac{1}{(2k+2)(2k+1)}$, n=50 için yandaki işlemin sonucunu bulan matlab programını yazınız.

```
clc;
sonuc=0;
for i=0:1:50
    yenisonuc = 1/ ((2*i)+1) * ((2*i)+2) ;
    sonuc=sonuc+yenisonuc;
end
fprintf('sonuc %g ',sonuc);
```

Örnek : $3 \sum_{k=12}^m \frac{5^{3k}}{7k+5}$ formülünü m=25 için hesaplayan matlab programını yazınız.

```
clc;
m=25;
toplam =0;
for i=12:m
    sonuc=(5^(3*i))/((7*i)+5);
    toplam=toplam+sonuc;
end
fprintf(' toplam = %g',3*toplam);
```

Örnek : Klavyeden 0 – 4 aralığında girilen bir tamsayının yazı ile kaç olduğunu ekrana yazan, bu değerlerin dışında bir değer girilirse hata veren matlab programını yazınız.

```
clc;
sayi=input('bir sayı giriniz : ');
switch sayi
    case 0
        fprintf('sıfır');
    case 1
        fprintf('bir');
    case 2
        fprintf('iki');
    case 3
        fprintf('üç');
    case 4
        fprintf('dört');
    otherwise
        error(' girdiğiniz sayı 0 ile 4 arasında değil ');
end
```

Örnek : 20 ye kadar olan sayılardan 3 ile bölünenleri ekrana yazdıran programı yazınız.

```
clc;
for i=1:20
    if mod(i,3)==0
        fprintf('%d \n',i);
    end
end
```

Örnek : klavyeden girilen sayıya kadar olan tüm sayıların toplam ve çarpımını ekrana yazdıran matlab programını yazınız.

```
clc;
sayi=input('sayi gir : ');
toplama=0;
carpim =1;

for i=1:sayi;

    toplam =toplam +i;
    carpim =carpim *i;

end

fprintf(' toplam = %d , carpim = %d' ,toplama,carpim);
```

Örnek : 3 santimetreyi, klavyeden girilen *cm*, *inch*, *mm* gibi birimlere göre dönüştürüp o birim türündeki değeri , hatalı giriş yapılmışsa hata mesajını ekrana yazan matlab programını yazınız.

```
clc;
x=3.0;
birim =input('birim giriniz :','s');
switch birim
    case {'in','inch'}
        y = 2.54 * x;
        fprintf('%d %s : %d cm ',x,birim,y);
    case {'m','metre'}
        y=x*100;
        fprintf('%d %s : %d cm ',x,birim,y);
    case {'milimetre','mm'}
        y=x/10;
        fprintf('%d %s : %d cm ',x,birim,y);
    case {'cm','santimetre'}
        y=x;
        fprintf( [num2str(x) ' ' birim ':' num2str(y) 'cm' ]);
    otherwise
        fprintf('bilinmeyen : %s ',birim);
        y=nan;
end
```

Örnek : verilen iki adet matrisin elemanlarını kıyaslayıp büyükse 1 değilse 0 olarak yeni bir matrise değer atayan , boyutlar aynı değilse hata mesajı veren programı *try catch* yapısını kullanarak yazınız.

```
clc;
A=[2 5 43];
B=[2 4 -1 ];
try
    A
    B
    C=A>B
catch
    fprintf(' dizi boyutlari ayni değil ');
end
```

```
A =
     2     5    43

B =
     2     4    -1

C =
     0     1     1

>>
```

Örnek : while ve rand komutlarını kullanarak ve kaç adet zar ikilisi olacağını kullanıcıdan alarak zar atışları yapıp, çıkan sonuçları ekrana yazdıran matlab programını yazınız.

```
clc;
n=1; %sifirdan farklı olsun diye
while n~=0
    ikili_sayisi=input('kaç adet ikili üretmek istiyorsunuz ? : ');
    uretilen =floor(1+6*rand(ikili_sayisi,2));
    fprintf(' - - - - - \n');
    fprintf('zar : %d\n' ,uretilen);
    n=input(' tekrar zar atmak için 1 e , durdurmak için 0 a basın : ');
end
fprintf('program durduruldu...');
```

Not : yukarıdaki örnekte verdiğimiz ;

```
n=input(' tekrar zar atmak için 1 e , durdurmak için 0 a basın : ');
```

satırında, programın devam etmesi için illa 1 sayısı girilecek diye bir kural yoktur. **Sıfırdan farklı olmak koşulu ile** girilen bütün n sayıları için döngü (*program*) devam eder. Bu özellik while döngüsünün özelliğidir.

Örnek : Ekrana matris görüntüsünü çıkaran matlab

1	1	1	0	0
1	1	1	0	0
1	1	1	0	0
0	0	0	5	5
0	0	0	5	5

Örnek : Ekrana

matris görüntüsünü çıkaran matlab

programını yazınız. (Osmangazi Üniversitesi - matematik ve bilgisayar bilimleri – temel bilgi teknolojileri 1 dersi - 2013 vize1 sınav sorusu)

```
clc;
A=ones(3,3);
A
B=5* ones(2,2);
B
A(4:5,4:5)=B;
A
```

A =		B =		
1	1	1	5	5
1	1	1	5	5
1	1	1		

A = (B ye ekledikten sonra)

1	1	1	0	0
1	1	1	0	0
1	1	1	0	0
0	0	0	5	5
0	0	0	5	5

YER DEĞİŞTİRME ALGORİTMASI

x ve y gibi iki değişken verildiğinde bu değişkenlerin değerlerini yer değiştirmek için;

- İlk olarak x değişkeninin değeri geçici bir değişkene atanır.
- Daha sonra y değişkeninin değeri x değişkenine atanır.
- Son olarak da y değişkenine, geçici değişkendeki değer atanarak yer değiştirme işlemi tamamlanmış olur.

<pre>clc; x=26; y=53; gecici=x; x=y; y=gecici; x y</pre>	<p>x =</p> <p>26</p> <p>y =</p> <p>53</p>	<p>x =</p> <p>53</p> <p>y =</p> <p>26</p>
--	---	---

SIRALAMA ALGORİTMASI

Verilen bir A dizisinin (*satır vektörünün*) elemanlarının küçükten büyüye doğru sıralamak için matlabda *sort(A)* fonksiyonu kullanılır. bu dersimizde bu fonksiyonu kullanmadan benzer algoritmalar nasıl yazılabilir bunu inceleyeceğiz.

<pre>clc; A=[44,23,1,78,5,6]; A B=sort(A); B</pre>	<pre>A = 44 23 1 78 5 6 B = 1 5 6 23 44 78</pre>
--	---

Örnek : Elemanları 1 ile 50 arasında değişen 10 elemanlı bir dizinin elemanlarını küçükten büyüğe sıralayınız.

1. Yol

- Dizinin en küçük elemanı belirlenir.
- Daha sonra bu eleman ile ilk sıradaki eleman yer değiştirir.
- daha sonra dizinin sıralanmış kısmındaki en küçük eleman bulunarak 2. Sıradaki eleman ile yer değiştirir.
- Bu böyle devam ederek **n** kez bu işlem yapıldığında , n elemanlı bir dizi sıranmış olur.

```
clc;
A=round(1+49 * rand(1,10));
A
for i=1:length(A)
    en_kucuk = i;
    for j=i+1:length(A)
        if A(j) < A(en_kucuk)
            en_kucuk=j;
        end
    end
    gecici =A(i);
    A(i)=A(en_kucuk);
    A(en_kucuk)=gecici;
end
A
```

2. Yol :

- Sıralanmış diziye, sondan başlanarak elemanların ikişerli kontrol yapılır.
- Eğer sağdaki eleman soldakinden daha büyük ise yer değiştirilir.
- bu işlem yeteri kadar tekrar ederek tüm dizi sıralanmış olur.

```
clc;  
A=round(1+49 * rand(1,10));  
A  
for i=1:length(A)  
  
    for j=length(A):-1:1+i  
  
        if A(j)<A(j-1)  
            gecici =A(j-1);  
            A(j-1)=A(j);  
            A(j)=gecici;  
        End  
  
    end  
end  
A
```

A =

5 4 27 39 47 7 29 24 2 18

A =

2 4 5 7 18 24 27 29 39 47

MATLABTA FONKSİYON OLUŞTURMAK VE PROGRAMA ÇAĞIRMAK

- Matlabdaki fonksiyonlar görsel veya metin tabanlı olmak üzere 2 şekilde elde edilir.
- Clc, sin, cos, rand gibi bazı fonksiyonlar sistem tarafından yazılmışken bazı fonksiyonları da kullanıcı kendisi yazar.
- Fonksiyonların kendine özgü bir ismi vardır. Yani, clc, sin,rand ... gibi sistemin içinde var olan fonksiyon isimlerini tekrar kullanamazsınız.
- Fonksiyonların ismi verilirken değişken tanımlama kuralları aynen geçerlidir.
(bakınız : sayfa 3)
- Her fonksiyon için bir .m dosyası oluşturulmalıdır.
- m belgesinin ismi ile fonksiyon aynı olmalıdır ve aynı belgeye ancak fonksiyonun alt fonksiyonları yazılabilir.
- Fonksiyon olan m belgesi **function** anahtar kelimesi ile başlar.
- Parametre alan fonksiyon editör üzerindeki çalıştırma butonu ya da f5 ile çalıştırılmaz. Komut satırına ilgili parametre ile fonksiyonun adı yazılmalıdır.
- Fonksiyonlara genellikle bir giriş ve çıkış parametresi alınır.

Fonksiyon olan m dosyasının 3 temel bileşeni vardır;

1. Fonksiyonun imzası ya da prototipi (*ilk satırda tanımlanır*)
2. Fonksiyonun açıklaması (*olmak zorunda değildir*), eğer varsa ikinci satırdan itibaren yorum satırı olarak yazılır.
3. Fonksiyonun tanımı (*açıklama var ise açıklamadan sonra, açıklama yok ise 2. Satırdan başlar.*)

Bir fonksiyonunu imzası aşağıdakiler gibi olabilir ; cp = cikis parametresi ve gp = giriş parametresidir.

- `function [cp1, cp2 ...] = fonksiyonun_adi (gp1, gp2 ...)`
- `function cp1 = fonksiyonun_adi (gp1, gp2, ...)`
- `function [cp1, cp2 ...] = fonksiyonun_adi (gp1)`
- `function cp1 = fonksiyonun_adi ()`
- `function fonksiyonun_adi(gp1)`

Örnek : iki nokta arasındaki uzaklığı bulan matlab fonksiyonunu yazınız.

$$\text{Formülü} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

```
function sonuc = uzaklik_hesapla(x1,y1,x2,y2)
%(x1,y1) ile (x2,y2) notalarını arandaki uzaklığı hesaplar
% u benim ilk fonksiyonum
sonuc = (sqrt(x2-x1)^2 + (y2-y1)^2);
end
```

Örnek : $f(x,y) = x^2y + \sqrt{xy} + \frac{\ln x + 1}{\log_{10} y} + \tan\left(\frac{x}{y}\right)$ fonksiyonunu matlab diline çeviriniz.

```
function sonuc = fonksiyon1(x,y)

sonuc =( x^2+y) + sqrt(x*y) + (log(x) +1) / (log10(y)) +tan(x/y);

end
```

Not : fonksiyonu program içerisinde çağırmak için aşağıdaki yolları kullanabilirsiniz ;

1. yol

```
clc;
x=10;
y=53;
sonuc =fonksiyon1( x,y);
...
```

2. yol

```
clc;
sonuc =fonksiyon1( 10,53);
...
```

Örnek : verilen n sayısının faktöriyelini bulan matlab fonksiyonunu yazınız.

```
function sonuc = faktoriyeli_bul(n)
sonuc =1;
    for i=1:n
        sonuc = sonuc * i;
    end
end
```

Örnek : verilen n sayısından küçük olan tüm asal sayıları listeleyen matlab fonksiyonunu **alt fonksiyon** kullanarak yazınız. (*Osmangazi Üniversitesi - matematik ve bilgisayar bilimleri – temel bilgi teknolojileri 1 dersi - 2012 vize2 sınav sorusu*)

```
function liste = asal_sayilari_bul(n)
    liste=[];
    for i=2:n
        if(asal_mi(i)==1)
            liste=[liste i];
        end
    end
end

function sonuc = asal_mi(sayi)
    bolenler=[];
    for i=1:sayi
        if(mod(sayi,i)==0)
            bolenler=[bolenler i];
        end
    end

    if(length(bolenler)==2)
        sonuc=1;
    else
        sonuc=0;
    end
end
```

Programda çağırmak ve ekrana yazdırmak:

```
clc;
liste=[];
liste=asal_sayilari_bul(58);
liste
```

Örnek : kendine giriş parametresi olarak aldığı x ve y değişkenlerinin değerlerini değiştiren matlab fonksiyonunu yazınız.

```
function [x,y]=yerlerini_degistir(x,y)
gecici=x;
x=y;
y=gecici;
end
```

Program içinde kullanımı:

```
clc;
sayi1=53;
sayi2=1989;
sayi1
sayi2
[sayi1,sayi2]= yerlerini_degistir(sayi1,sayi2);
sayi1
sayi2
```

Ekran çıktısı:

```
sayi1 =          sayi2 =
      53          1989

sayi1 =          sayi2 =
      1989         53
```

Çıkış parametresinin birden çok olduğu durumlarda çıkan sonucu değişkene almak için, aşağıdaki yöntem uygulanır.

[x,y]=degistir(2,3);

Aksi halde editör sadece ilk değeri, **ans** değişkenine atar ve ikincisini atamaz.

KABARCİK SIRALAMA (BUBBLE SORT) ALGORİTMASI

Bilgisayar bilimlerinde kullanılan yalın bir sıralama algoritmasıdır. Sıralanacak dizinin üzerinde sürekli ilerlerken her defasında iki öğenin birbiriyle karşılaştırılıp, karşılaştırılan öğelerin yanlış sırada olmaları durumunda yerlerinin değiştirilmesi mantığına dayanır. Algoritma, herhangi bir değişiklik yapılmıyınca kadar dizinin başına dönerek kendisini yineler. Adına "*Kabarcık*" sıralaması denmesinin nedeni büyük olan sayıların aynı suyun altındaki bir kabarcık gibi dizinin üstüne doğru ilerlemesidir.

Başlangıçta yer yer *değiştirme sıralaması* olarak adlandırılan kabarcık sıralaması, dizi içindeki büyük elemanların algoritmanın her adımında dizinin sonuna doğru doğrusal olarak ilerlemesini sağlar. Bu ilerleme, seçmeli sıralama algoritmasındaki dizideki değeri küçük olan elemanların dizinin başında kümelenmesi yöntemine benzer şekilde gerçekleşir.

Algoritmanın Adım Adım İşleyişi

İçeriği "5 1 4 2 8" olan bir dizi kabarcık sıralaması ile en küçükten en büyüğe doğru aşağıdaki biçimde sıralanır. Her adımda dizinin **kalın** olarak işaretlenmiş elemanları karşılaştırılan elemanlardır.

Birinci Geçiş:

(5 1 4 2 8) → (1 5 4 2 8) Burada algoritma ilk iki elemanı karşılaştırır ve yerlerini değiştirir.
(1 5 4 2 8) → (1 4 5 2 8)
(1 4 5 2 8) → (1 4 2 5 8)
(1 4 2 5 8) → (1 4 2 5 8) Burada elemanlar zaten sıralı olduğu için algoritma yerlerini değiştirmez.

İkinci Geçiş:

(1 4 2 5 8) → (1 4 2 5 8)
(1 4 2 5 8) → (1 2 4 5 8)
(1 2 4 5 8) → (1 2 4 5 8)
(1 2 4 5 8) → (1 2 4 5 8)

Artık dizi sıralıdır ancak algoritma işlemin bittiğini bilmemektedir. Algoritmanın dizinin sıralandığını anlaması için bütün dizinin üzerinden hiçbir değişiklik yapmadan **tam** bir geçiş yapması gerekir.

Üçüncü Geçiş:

(1 2 4 5 8) → (1 2 4 5 8)
(1 2 4 5 8) → (1 2 4 5 8)
(1 2 4 5 8) → (1 2 4 5 8)
(1 2 4 5 8) → (1 2 4 5 8)

Sonuç olarak dizi sıralanmıştır ve algoritma sonlanır.

Örnek : Elemanları 1 ile 100 arasında değişen 20 elemanlı rastgele bir dizi oluşturup, bu diziye kabarcık sıralama algoritması kullanarak sıralayan matlab programını yazınız.
(Osmangazi Üniversitesi - matematik ve bilgisayar bilimleri – temel bilgi teknolojileri 1 dersi - 2013 vize2 sınav sorusu)

1. yol :

```
clc;
A=[1,20];
for t=1:20
    A(1,t)=round(1+99* rand(1,1));
end

for e=1:20
    fprintf(' %2d ',A(1,e));
end

for i=1:20
    for j=20:-1:1+i
        if A(1,j-1)>=A(1,j)
            gecici=A(j-1);
            A(1,j-1)=A(1,j);
            A(1,j)=gecici;
        end
    end
end
fprintf('\n sirali dizi elemanları : \n');

for e=1:20
    fprintf(' %2d ',A(1,e));
end
```

2. yol :

```
clc;
A=round(1+99* rand(1,20));
A
for i=1:20
    for j=20:-1:1+i
        if A(j-1)>=A(j)
            gecici=A(j-1);
            A(j-1)=A(j);
            A(j)=gecici;
        end
    end
end
A
```

MATLABDA KULLANILAN BAZI YUVARLAMA VE STRİNG FONKSİYONLARI

Round :	Ondalıklı kısım 0.5 ten küçükse aşağıdaki tamsayıya , değilse yukarıdaki tamsayıya yuvarlar.
Ceil :	Ondalıklı kısmı $+\infty$ 'a doğru yuvarlar
Fix :	Ondalıklı kısmı sıfıra doğru yuvarlar
Flor :	Ondalıklı kısmı $-\infty$ 'a doğru yuvarlar.

Örnek :

A sayısı	Ceil(a)	Fix(a)	Flor(a)	Round(a)
- 2.5	-2	-2	-3	-3
-1.75	-1	-1	-2	-2
-1.25	-1	-1	-2	-1
-0.5	0	0	-1	-1
0.5	1	0	0	1
1.25	2	1	1	1
1.75	2	1	1	2
2.5	3	2	2	3

Str2num :	Verilen string değeri, eğer mümkünse sayıya çevirir. Not: 123a gibi bir string değer sayıya dönüştürülemez.
Num2str :	Verilen string değeri mümkünse sayıya çevirir
Flplr:	Verilen listeyi terse çevirir. [1,2,3,4] dizisini [4,3,2,1] yapar.
Rem :	Bir sayının başka bir sayı bölümünden kalan verir. (mod ile aynı işi yapar)
Nargin :	Giriş yapılan parametre sayısını ifade eder
Nargout :	Çıkış yapılan parametre sayısını ifade eder.
Varargin :	Tüm giriş parametrelerini ifade eder
Varargout :	Tüm çıkış parametrelerini ifade eder.

Palindrom Kelime : Sağdan ve soldan okunuşu aynı olan kelimeye denir.

Örnek kelimeler : ece, sos, elle, kazak, küllük, nacican...

Örnek : Verilen bir kelimenin palindrom olup olmadığını test eden bir matlab fonksiyonu yazınız. *(Osmangazi Üniversitesi - matematik ve bilgisayar bilimleri – temel bilgi teknolojileri 1 dersi – 2012 vize2 sınav sorusu)*

```
function sonuc = palindrom_mu(kelime)
boyut = length(kelime);
yarisi = fix(boyut/2);
sonuc = 'bu kelime palindromdur';
    for i=0:yarisi
        if kelime(i+1) == kelime(boyut-i)
            continue
        else
            sonuc = 'bu kelime palindrom değildir';
            break
        end
    end
end
```

Fibonacci dizisi : her sayının kendinden öncekiyle toplanması sonucu oluşan bir sayı dizisidir.


1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, şeklinde devam etmektedir.

Örnek : 1 ile 1000 arasındaki fibonacci dizi elemanlarını yazdıran matlab programını yazınız. *(Osmangazi Üniversitesi - matematik ve bilgisayar bilimleri – temel bilgi teknolojileri 1 dersi – 2013 vize2 sınav sorusu)*

```
clc;
liste=[];
liste(1)=1;
liste(2)=1;
for i=3:1000
    eleman =liste(i-1) + liste(i-2);
    liste(i)=eleman;
end

for i=1:1000
    fprintf('%d ',liste(i));
end
```

Ekran çıktısı :

 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610

Örnek : a bir kare matris ve n bir pozitif tam sayı olmak üzere

$$A^n = (A^{n-1}) * A$$

Olarak tanımlanır. Buna göre herhangi bir matris ve n sayısı verildiğinde A^n değerini hesaplayacak matlab programını alt fonksiyon kullanarak yazınız. *(Osmangazi Üniversitesi - matematik ve bilgisayar bilimleri – temel bilgi teknolojileri 1 dersi – 2013 vize2 sınav sorusu)*

```
function sonuc = matrisi_hesapla(A,n)

[x,y]=size(A);
if(x~=y)
    error(' malesef matrisiniz kare matris degil ');
else
    sonuc=A;
    for i=1:n-1
        sonuc=carp(sonuc,A);
    end
end

function C=carp(M1,M2)
x=size(M1);
C=zeros(x);
for i=1:x
    for j=1:x
        for k=1:x
            C(i,j) = C(i,j) + M1(i,k) * M2(k,j);
        end
    end
end
end
```

Örnek : Girilen n sayısına karşılık, n satır ve n sütundan oluşan, * lar kullanarak dik üçgen çizdiren matlab fonksiyonunu yazınız. *(Osmangazi Üniversitesi - matematik ve bilgisayar bilimleri – temel bilgi teknolojileri 1 dersi – 2012 vize2 sınav sorusu)*

```
function ucgen_ciz(n)
for i=1:n
    for j=1:i
        fprintf('* ');
    end
    fprintf('\n');
end
```

Ekran çıktısı :

```
*
**
***
****
*****
fx >>
```


Armstrong Sayılar : 3 basamaklı sayılardan, basamak değerlerinin küpleri toplamı sayıya eşit ise bu sayıya Armstrong sayı denir.

Örneğin : 153 bir Armstrong sayısıdır, çünkü : $1^3 + 5^3 + 3^3 = 153$ tür.

Örnek : Tüm Armstrong sayıları bulan bir matlab fonksiyonu yazınız.

```
function liste = armstrong_bul()

liste=[];
for i=100:999
    s=num2str(i);
    s1=str2num(s(1));
    s2=str2num(s(2));
    s3=str2num(s(3));
    toplam=s1^3 + s2 ^3 + s3^3;
    if i==toplam
        liste=[liste i];
    end
end
end
```

Ekran çıktısı :

```
ans =
    153    370    371    407
fx >>
```

Örnek : İç içe fonksiyon kullanarak faktöriyel hesabı yapan matlab fonksiyonu yazınız.

```
function sonuc = faktoriyel(n)
    if(n<0)
        error('sayı 0 dan küçük olamaz');
    end
    if n==0
        sonuc =1;
    else
        sonuc = sonuc * faktoriyel(n-1);
    end
end
```

MATLABDA TABAN ARİTMETİĞİ İŞLEMLERİ

Decimal Sistem (onluk sayı sistemi) : Günlük hayatımızda decimal sistemi kullanmaktayız ve tabanı10'dur. **Örneğin :** $365 = (3 * 10^2) + (6 * 10^1) + (5 * 10^0)$

İkili SayıSistemi (Binary System) :

- Sistemin Tabanı2'dir.
- Sadece "0"ve "1"kullanır.
- Her sayı"dijit"olarak ifade edilir.
- Basamaklar 2'nin kuvveti olarak yazılır. **Örnek :** 101011 , 1010, 1111, 100

Binary-Decimal Çevrimi : Her dijit, 2'nin kuvveti ile çarpılarak decimal sayı sistemine çevrilir.

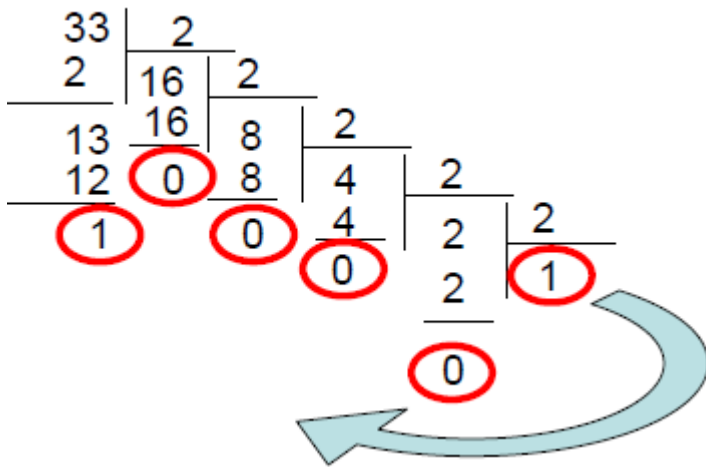
Örneğin :

$$(101011)_2 = 1.2^5 + 0.2^4 + 1.2^3 + 0.2^2 + 1.2^1 + 1.2^0 = 43$$

Decimal-Binary Çevrimi : Decimal sayıları binary sayılara çevirirken, her sayı bölüm 2 ye eşit oluncaya kadar ikiye bölünür. kalanlar tersten yazılır.

Örneğin :

$$(33)_{10} = (?)_2$$



$$(33)_{10} = (100001)_2$$

Örnek : İkilik sistemde verilen bir sayıyı onluk sisteme çeviren matlab fonksiyonunu yazınız.

```
function sonuc=ikiligi_onluk_yap(ikilik_sayi)

    sonuc=0;
    ikilik_yazi = num2str(ikilik_sayi);
    uzunluk=length(ikilik_yazi);
    kuvvet=uzunluk-1;

    for i=1:uzunluk
        if str2num(ikilik_yazi(i)) > 1
            error('sayi hatali formatta');
        else
            sonuc=sonuc + str2num(ikilik_yazi(i)) * 2^kuvvet;
        end
    end
end
```

Ekran çıktısı :

onluk sayi giriniz : 53 sonuc = 110101

Örnek : Onluk tabandan iklik tabana dönmeyi sağlayacak bir matlab fonksiyonu yazınız.

```
function sonuc = onlugu_ikilik_yap(sayi)
i=1;
bolum =fix(sayi/2); %floor da olabilir
kalan= rem(sayi,2);
sonuc(i) = num2str(kalan); %listeye ekledik kalanı

    while ( 2 <= bolum )
        sayi =bolum;
        bolum =fix(sayi/2); %floor da olabilir
        kalan= rem(sayi,2);
        i=i+1;
        sonuc(i) =num2str(kalan);
    end
    sonuc(i+1) = num2str(bolum);
    sonuc =fliplr(sonuc);
end
```

Ekran çıktısı :

iklik sayi giriniz : 110101110 sonuc = 1536

Örnek : Eğer sadece 2 adet giriş parametresi verilmişse, sayılar arası uzaklığı bulan matlab fonksiyonunu yazınız.

```
function uzaklik = uzaklik_bul(x,y)
    if nargin == 2
        fprintf('iki deger arası uzaklık : ');
        uzaklik=abs(x-y);
    end
end
```

Şimdi de aynı örneği giriş yapılan ve çıkış yapılan parametre sayılarına sınır koymadan tekrar kodlayalım.

```
function [varargout] =uzaklik_bul(varargin)
x = varargin{1};
y = varargin{2};
    if nargin == 2
        fprintf('iki deger arası uzaklık : ');
        uzaklik=abs(x-y);
        varargout{1} = uzaklik;
    end
end
```

Örnek : x ve y birbirinden farklı sayılar olmak üzere,

$|x| + |y| \leq 3$ şartını sağlayan tüm x,y ikililerini ekrana yazan matlab programını yazınız.

```
clc;
for x=-3:3
    for y=-3:4
        islem=abs(x)+abs(y);
        if islem<=3
            fprintf('x = %d , y = %d \n',x,y);
        end
    end
end
```

```
x = -3 , y = 0
x = -2 , y = -1
x = -2 , y = 0
x = -2 , y = 1
x = -1 , y = -2
x = -1 , y = -1
x = -1 , y = 0
x = -1 , y = 1
x = -1 , y = 2
x = 0 , y = -3
x = 0 , y = -2
x = 0 , y = -1
x = 0 , y = 0
...
```