

3.2.4 Dizeler üzerinde işlemler

Tüm değişkenleri dize olarak ele alan yapısı, çok çeşitli araç kutuları ve kütüphane fonksiyonları ile MATLAB, mühendislik ve teknik hesaplama problemlerin çözümünde büyük kolaylıklar sağlamaktadır. Aşağıda çok sık başvurulanan dize işlemleri sırasıyla örnekler ile açıklanmaktadır.

Dize boyutlarının bulunması:

MATLAB programları ya da komut satırı işlemlerinde üretilen dizelerin boyutlarının bu dizeler üzerinde yürütülecek işlemler sırasında biliniyor olması gereklidir. Dize boyutlarının ve eleman sayılarının bulunması için kullanılan başlıca MATLAB fonksiyonları **length**, **size** ve **numel** isimleri ile kütüphaneden çağırılmaktadır. Aşağıdaki örnekler fonksiyonların kullanımının anlaşılması için incelenebilir:

```
x=rand(2,7);
length(x) %x bir vektör ise uzunluğunu; dize ise boyutlarından en büyüğünü bulur.
ans =
     7

x=-2*pi:0.01:2*pi;%-2pi,2pi aralığını 0.01 artımla örnekleyen x'in uzunluğu
length(x)
ans =
    1257

x=[1 2 8 -1 6 0];%1x6 boyutlarındaki x dizeyinin uzunluğu
length(x)
ans =
     6

k=[7 0; 6 8; 5 4; 1 -1];%4x2 boyutlarındaki k dizeyinin uzunluğu
length(k)
ans =
     4
```

```
size(D,1) %D dizeyinin satırlarının sayısı
size(D,2) %D dizeyinin sütunlarının sayısı
size(D)   %D dizeyinin tüm yönlerdeki boyutları
```

```
a= rand(4,2,6);
b=size(a) %Üç boyutlu bir dizeyin boyutlarının bulunması
b =
     4     2     6
c=size(b,2) %b dizeyinin sütun sayısı
c =
     2
```

```
g=rand(5,6);
numel(g) %g dizisinin eleman sayısı
ans=
    30
```

Bir dizinin dönüğünün alınması (transpose):

Dizeler (matris) işlemlerinde sıkça gerek duyulan bir işlemdir. Bir dizinin dönüğü MATLAB'da aşağıdaki gibi alınabilir:

```
ct=c';
```

Buradaki tek tırnak işareti dizinin dönüğünü almak için kullanılan işlemdir. Verilen örnekte **c** dizisinin dönüğü alınarak **ct** dizisine atanmaktadır.

Dizinin en büyük ve en küçük elemanlarının bulunması:

Bir dizinin elemanları arasında en büyük ya da en küçük olanların bulunması amacı ile MATLAB kütüphanesinde yer alan **min** ve **max** fonksiyonları kullanılabilir. Her iki fonksiyonu kullanırken dizilerin boyutlarına dikkat edilmelidir. Aşağıda farklı durumlar için fonksiyonların kullanımı verilmiştir.

```
enbuyuk=max(a);
```

a bir boyutlu ise dizideki en büyük elemanı bularak **enbuyuk** isimli değişkene atar, iki boyutlu ise her bir sütundaki en büyük elemanlar bulunur.

```
eB=max(a,b);
```

a ve **b** birer skaler ise sayılardan büyük olanı **eB** isimli değişkene aktarır. **a** ve **b** aynı boyutta iki boyutlu diziler olmak üzere her bir eleman karşılıklı olarak karşılaştırılır, büyük olan **eB** değişkeninde karşılık gelen eleman konumuna yazılır.

```
G=max(a,[],dim)
```

a dizisinin **dim** ile belirtilen boyutundaki en büyük elemanları bularak **G** dizisine aktarır. **dim** 1 ise satır yönünde, 2 ise sütun yönünde bir tarama yapılır. (İçerisi boş köşeli parantezler karşılaştırma seçeneğinin kullanılmayacağını belirtmek için kullanılmıştır.)

```
[deg,yer]=max(X);
```

X' in en büyük elemanını ve yeri bulunur. Diziler bir boyutlu ise **deg** ve **yer** 1x1 boyutundadır. İki boyutlu dizilerde her bir sütundaki en büyük değer ve o sütundaki yeri bulunur.

yukarıda verilen yazılış ve kurallar aynı şekilde **min** fonksiyonu için de geçerlidir. **max** yerine **min** yazılarak bu kez aynı kurallar çerçevesinde en küçük elemanlar bulunabilir. Aşağıda fonksiyonların kullanımına çeşitli örnekler verilmiştir.

```
>>a=[1 5 -8 0];
>>max(a)
ans =
     5
>>enk=min(a)
enk =
    -8
>>a=3.7; b=4.1;
>>max(a,b)
ans=
    4.1000
```

```
% R dizisinin her sütundaki en büyük elemanı ve bulunduğu satırlar
R=rand(3,4)
R =
    0.4468    0.5108    0.6443    0.5328
    0.3063    0.8176    0.3786    0.3507
    0.5085    0.7948    0.8116    0.9390
[deg,enb]=max(R)
deg =
    0.5085    0.8176    0.8116    0.9390
enb =
     3     2     3     3
% R1 dizisinin her satırdaki en küçük elemanı ve bulunduğu satırlar
R1=rand(4,5)
R1 =
    0.5949    0.2217    0.4242    0.8010    0.4886
    0.2622    0.1174    0.5079    0.0292    0.5785
    0.6028    0.2967    0.0855    0.9289    0.2373
    0.7112    0.3188    0.2625    0.7303    0.4588
[deg,enb]=min(R1,[],2)
deg =
    0.2217
    0.0292
    0.0855
    0.2625
enb =
     2
     4
     3
     3
```

Boyutuna bakılmaksızın dizeyi içindeki en büyük ya da en küçük eleman bulunmak istenirse `max(R1(:))` ya da `min(R1(:))` yazılışları kullanılabilir. Burada indeks hanesindeki `:` işareti dizeyi bir sütun yöneye dönüştürmektedir.

Toplama ve çıkarma

MATLAB'da tüm değişkenlerin birer dizeyi olarak saklandığı bilgisini tekrar hatırlatarak toplama işlemi aşağıdaki şekilde yapılmaktadır. İki dizeyin toplamı

```
T=a1+x_G;
```

yazılışı ile bulunabilir. Eşitliğin sağındaki dizelerin karşılıklı elemanları toplanarak T dizeyinin aynı sıradaki elemanına aktarılır. Burada `a1` ve `x_G` dizelerinin aynı boyutta olması gerektiği açıktır. Boyutları aynı olmayan dizeler toplanamazlar. Bu duruma istisna, bir skaler ile bir dizeyin toplanmasıdır.

```
R=[1 2 7;3 -1 0];
```

```
t=10+R
```

```
t =
```

```
    11    12    17
    13     9    10
```

Çıkarma işlemi için de dizelerin aynı boyutta olması kuralı geçerlidir. İki dizeyin karşılık gelen elemanları birbirinden çıkarılır. Bir skalerin dizeyin tüm elemanlarından çıkarılması da önceki örnekte olduğu gibi gerçekleştirilebilir.

```
F=[1 6 10 -5 2]';
```

```
G=[0 -3 1 0.5 2]';
```

```
>>H=F-G; %F ve G dizelerinin elemanları birbirinden çıkarılıyor
```

```
>>G-5
```

```
ans =
```

```
-5.0000
-8.0000
-4.0000
-4.5000
-3.0000
```

Çarpma ve bölme

Çarpma ve bölme işleminde durum toplama ve çıkarma işlemlerine göre biraz daha farklıdır. Değişik olasılıklar gözetilerek geliştirilmiş çeşitli MATLAB yazılışları bulunmaktadır. İki dizeyin çarpılabilmesi

için birinci dizinin sütun sayısı ikinci dizinin satır sayısına eşit olmalıdır ($n \times m$ ve $m \times k$ gibi). İki dizinin çarpımı $n \times k$ boyutlarında bir dizeyi üretir. Bu şart sağlanıyorsa a ve b gibi iki dize

```
c=a*b;
```

yazılışı ile çarpılarak sonucu c isimli dizeye atanabilir. Bir skaler ile herhangi bir dizi boyutları ile ilgili bir uyum beklenmeden çarpılabilir. Bu durumda dizinin tüm elemanları aynı skaler sayı ile çarpılır.

```
>>F1=2.55*dM;
```

yazılışı buna örnek olarak verilebilir. Dize çarpımından farklı olarak aynı boyutta iki dizinin karşılık gelen elemanları birbirleriyle çarpılmak istenebilir. Örneğin $p \times q$ boyutlarındaki **k** ve **dV** dizilerinin karşılık gelen tüm elemanları birbirleri ile çarpılmak istenirse

```
k=[1.25 2.5 4.0 6];
dV=[0.18 0.09 0.06 0.02];
R=k.*dV
R =
    0.2250    0.2250    0.2400    0.1200
```

sonucu elde edilir. Burada **.*** yazılışı çarpma işleminin karşılıklı elemanlar üzerinde yürütüleceğini belirtmektedir. Bu durumda her iki dizinin aynı boyutta olması gerektiği açıktır. MATLAB ile dize bölme işlemi **C=A/B**; yazımı ile kolayca yapılabilir. Bölme işleminin yapılabilmesi için dize boyutlarının işlem ile uyumlu olması gereklidir. Bölme işleminin elemanlar olarak yapılması için **./** yazımının kullanılabileceği daha önce belirtilmiştir.

Dize elemanlarının toplamı-farkı

Bir dizinin tüm elemanlarının toplamı bir kodlama yapmaya gerek olmadan MATLAB kütüphanesinde bulunan **sum** fonksiyonu ile hesaplanabilir. Fonksiyon farklı boyutlardaki diziler için farklı tepki vermektedir. Bir boyutlu dizilerde tüm elemanların toplamı hesaplanırken iki ve üç boyutlu dizilerde fonksiyon varsayılan olarak sütunlar üzerinde ayrı ayrı işletilmektedir. Aşağıda bu durumlar için örnekler verilmiştir:

```
% 1-100 arasındaki sayıların toplamı
seri=1:100;
toplam=sum(seri)
toplam=
    5050
```

```
ornek=[1 2 3;4 5 6]
ornek =
     1     2     3
     4     5     6
% her bir sütundaki elemanların toplamı
sum(ornek)
ans =
     5     7     9
```

```
% her bir satırdaki elemanların toplamı
>> sum(ornek,2)
ans =
     6
    15
% tüm elemanların toplamı
>>sum(sum(ornek))
ans =
    21
%veya
>>sum(ornek(:))
ans =
    21
```

Son örnekte `(:)` ifadesi iki boyutlu bir dizeyi bir sütun yöneye dönüştürür. Bu işlemi sütunları sırası ile alt alta ekleyerek gerçekleştirir. Bu işlemden sonra toplamı alınan dizeyi bir yöneye dönüştüğünden bir adımda tüm elemanların toplamı alınabilir.

Dizeler elemanların toplamı ile ilgili bir diğer seçenek birikimli (cumulative) toplamadır. Bu işlem **cumsum** MATLAB fonksiyonu ile yürütülür ve ardışık elemanların toplamları birikimli olarak hesaplanır. Örneğin 1-10 arası tam sayıların birikimli toplamı **cumsum** fonksiyonu ile aşağıdaki şekilde hesaplanır:

```
a=1:10
a =
    Columns 1 through 10
     1     2     3     4     5     6     7     8     9    10
cumsum(a)
ans =
    Columns 1 through 10
     1     3     6    10    15    21    28    36    45    55
```

Bu fonksiyon iki boyutlu dizelere uygulandığında yine varsayılan olarak sütun yönünde işlem yapar. İki boyutlu bir dizeyin satırlarının birikimli toplamını hesaplamak için **cumsum(V,2)** şeklinde bir yazılış

tercih edilmelidir. Fonksiyona gönderilen ikinci argüman toplama işleminin hangi boyut için yapıldığını göstermektedir. Varsayılan değeri 1 olduğu için bir şey yazılmadığında sütunlar üzerinde çalışmaktadır.

```
>> V=[1 4 7;2 5 8; 3 6 9]
V =
     1     4     7
     2     5     8
     3     6     9
```

```
>> cumsum(V,1)
ans =
     1     4     7
     3     9    15
     6    15    24

>> cumsum(V,2)
ans =
     1     5    12
     2     7    15
     3     9    18
```

Ardışık dizeler elemanlarının farkının alınmasında ise **diff** fonksiyonu kullanılmaktadır. Ardışık elemanlar arasındaki fark aslında belirli bir adım boyu söz konusu ise türev işlemine karşılık gelir. Buna göre **sin(x)** fonksiyonunun -3 ile 3 arasındaki değerlerinin 0.5 adım boyu için sayısal türevi **diff** fonksiyonu kullanılarak aşağıdaki gibi hesaplanabilir:

```
x=-3:0.5:3;
f=sin (x);
df=diff(f); %f(i+1)-f(i)
dx=diff(x); %x(i+1)-x(i)
t=df./dx;
```

Verilen örnekte **diff** fonksiyonu birinci dereceden fark işleci olarak çalışmaktadır. Fonksiyonun çıktısı girdisinden 1 eleman daha kısadır. Bunun nedeni ardışık iki elemandan bir fark hesaplanmasıdır. İki ve daha yüksek dereceden farkları hesaplamak için fonksiyon **diff(f,2)** şeklinde kullanılabilir. Son yazılışı ile 2. derece farklar alınmaktadır. Fark işleci 2 ve daha yüksek boyutlu dizeler üzerinde kullanılıyorsa işleç varsayılan olarak sütunlara ayrı ayrı uygulanmaktadır. İstenilen yönde fark almak için üçüncü bir seçenek girilmektedir. Örneğin **diff(g,1,2)** iki boyutlu **g** dizeyinin satırlarındaki elemanların soldan sağa ardışık farklarını alır.

Dizey yapılandırma

Çalışma belleğinde bulunan bir dizeyin elemanlarının yerlerinin değiştirilmesi, içeriğinin boşaltılması, belirli bir yönde döndürülmesi, kopyalanması gibi işlemlere çeşitli amaçlarla gerek duyulmaktadır. MATLAB kütüphanesinde bu amaçla geliştirilmiş çeşitli fonksiyonlardan sık kullanılanlardan bazıları burada örnekler ile anlatılacaktır.

reshape

Dizeleri yeniden şekillendirmek için kullanılan bir fonksiyondur. Yeniden şekillendirme dizey elemanlarının yerleşimlerini değiştirmek anlamındadır. Örneğin tek bir liste olarak verilmiş bir dizi ölçümün sonucu genellikle görüntüleme amacı ile iki boyutlu bir dizeye dönüştürülmek istenebilir. Bu tür durumlar genellikle çeşitli standart biçimlerde verilen verilerin kendi amacınıza uygun düzenlemesi halinde ortaya çıkar. Fonksiyonun genel kullanımı şu şekildedir:

```
R=reshape(G,m,n);
```

Bu yazılış ile fonksiyon **G** dizeyinin içeriğindeki bilgileri **m** satır **n** sütun bir dizey oluşturacak şekilde yeniden düzenleyerek **R** dizeyine aktarır. Bu işlemin gerçekleştirilebilmesi için **G** dizeyindeki toplam eleman sayısının $m \times n$ adet olması gereklidir. Aşağıdaki örneklerde bu durum daha iyi anlaşılabilir.

```
>> a=1:16
a =
    Columns 1 through 9
         1         2         3         4         5         6         7         8         9
    Columns 10 through 16
        10        11        12        13        14        15        16
>> b=reshape(a,4,4)
b =
         1         5         9        13
         2         6        10        14
         3         7        11        15
         4         8        12        16
```

Yeniden düzenleme işleminde giriş dizeyinin elemanları sütun yönünde kullanılmaya başlanır.

```
>> B
B =
         1         3         5         7         9        11
         2         4         6         8        10        12
>> reshape(B,4,3)
ans =
         1         5         9
         2         6        10
```


3	7	11
4	8	12

örneğinde olduğu gibi 2x6 boyutlarındaki **B** dizeyi 4x3 boyutlarına dönüştürülmek istendiğinde yeni dizeyin satır yönündeki boyutu tamamlanıncaya kadar sütunlardaki elemanlar alt alta eklenmekte ve sonraki sütuna geçilmektedir.

fliplr-flipud

Dizeleri düşey ya da yatay eksen yönünde döndürmek için kullanılan fonksiyonlardır. Hatırlaması kolay olması açısından **fliplr** (**f**lip **l**eft-**r**ight) dizeyi sol-sağ yönünde döndürürken, **flipud** (**f**lip **u**p **u**own) dizeyi alt üst etmektedir. Örnekler durumu daha iyi açıklayacaktır:

```
>> x=1:5 % 1x5 boyutlarındaki satır yöney
x =
     1     2     3     4     5
```

```
>> xx=fliplr(x) % satır yöney sağ-sol yönünde döndürülüyor
xx =
     5     4     3     2     1
>> y=[1 4;2 5;3 6] % 3x2 boyutlarındaki dizey
y =
     1     4
     2     5
     3     6
```

```
>> fliplr(y) % sağ-sol
ans =
     4     1
     5     2
     6     3
```

```
>> flipud(y) % yukarı aşağı
ans =
     3     6
     2     5
     1     4
```

repmat

Bir dizeyin satır ya da sütun yönünde belirli sayıda tekrarından oluşan dizeler oluşturmak için kullanılan MATLAB fonksiyonudur. Genel kullanımı

```
C=repmat(t,m,n);
```

şeklindedir. Bu yazılış ile **t** dizeyini satır yönünde **m** kez, sütun yönünde ise **n** kez tekrarlayarak daha büyük **c** dizeyini oluşturmaktadır. Örnek:

```
>> r=repmat([1,2,3,4],4,1)
```

```
r =
     1     2     3     4
     1     2     3     4
     1     2     3     4
     1     2     3     4
```

Dizeler tersleme

\mathbf{I} birim dizeyi göstermek üzere $\mathbf{A}\mathbf{A}^{-1}=\mathbf{I}$ eşitliğini sağlayan \mathbf{A}^{-1} dizeyine \mathbf{A} 'nın tersi denir. Dizeler tersleme özellikle doğrusal denklem sistemlerinin çözümünde sık başvurulan bir dizeler işlemidir. Dizeler tersleme sırasında tekil değerlerden kaynaklanan sorunlar yaşanabilmektedir. MATLAB'da bir dizelerin tersi

```
>>T=inv(X);
```

yazılışı ile alınabilmektedir. Dizelerin tekillikten dolayı tersinin alınmasında bir zorluk olduğunda bu bir mesaj ile komut satırında görüntülenmektedir.

Dizeler elemanlarının sıralanması

MATLAB kütüphanesindeki **sort** fonksiyonu kullanılarak bir dizeler içerisinde yer alan elemanlar, artan ya da azalan yönde ve istenilen dizeler boyutunda (satır-sütun-sayfa) sıralanabilmektedir. Fonksiyonun genel kullanımı

```
B=sort(A);
```

şeklindeir. Varsayılan olarak dizeler küçükten büyüğe ve sütun yönünde sıralanır. Sıralama türü ve yönü

```
B=sort(A,dim,mode);
```

yazılışındaki **dim** ve **mode** seçenekleri ile değiştirilebilir. **dim** sıralamanın hangi yönde yapılacağını (1: sütunları kendi içinde sıralar, 2: satırları kendi içinde sıralar) **mode** ise ('ascend' artan 'descend' ise azalan sıralama yapar) sıralama şeklini belirler. Örnek:

```
>> f=[3 6 10 9 8 2 5 7];
>> sort(f)
ans =
     2     3     5     6     7     8     9    10
```

İki boyutlu bir dizeler üzerinde çalışması da aşağıdaki örnek ile açıklanabilir.

```
A = [ 5      7      4
      8      4      3
      9      1      7];
```

olmak üzere dizeyin elemanları ile ilgili çeşitli sıralama seçenekleri aşağıdaki gibi verilebilir:

```
>> sort(A)
ans =
     5     1     3
     8     4     4
     9     7     7
```

```
>> sort(A,2)
ans =
     4     5     7
     3     4     8
     1     7     9
```

```
>> sort(A,2,'descend')
ans =
     7     5     4
     8     4     3
     9     7     1
```

MATLAB kütüphanesindeki birçok fonksiyon çağılırken, parantez içerisine veri ile birlikte çeşitli seçenekler yazılabilmektedir. Bu aynı fonksiyon ile birden çok iş yapabilmemizi sağlar (overloaded functions). Buna benzer şekilde fonksiyonun ürettiği bilgilerin türleri de çıkış bölümüne yazılacak değişken sayısı ile değiştirilebilmektedir. **sort** fonksiyonu üzerinde gösterilmek istenirse bu durum aşağıdaki örnek ile daha iyi anlaşılır:

```
>> vG=[-1 2 0.7 4.2 -3];
>> [deg,yer]=sort(vG);
>> [deg' yer']
ans =
 -3.0000    5.0000
 -1.0000    1.0000
  0.7000    3.0000
  2.0000    2.0000
  4.2000    4.0000
```

Yukarıdaki örnekte fonksiyondan elde edilen çıktılar eşittir işaretinin solundaki köşeli parantezler içerisine yazılmıştır. Bu yazılış ile fonksiyondan iki adet çıkış beklendiği belirtilmektedir. Değişik fonksiyonlar çok daha fazla çıktı üretebilmektedir. Bu durumda hangi sırada üretildiklerini bilmek kaydıyla olası bilgiler burada olduğu gibi çıktı bölümüne sırası ile yazıldığında bu sonuçlar üretilerek çalışma belleğine aktarılır. **sort** fonksiyonu en fazla iki çıktı üretmektedir. Verilen örnekte **deg** değişkeni giriş değerlerinin sıralanmış halini, **yer** değişkeni ise sıralanmış değerlerin **vG** isimli giriş değişkeninde hangi indiste bulunduğunu göstermektedir. Burada sayılanlar dışında, dizeler üzerinde benzer matematik ya da yapılandırma işlemleri yürüten başka fonksiyonlar da bulunmaktadır. Bunlar ayrı ayrı incelenmek yerine aşağıda bir listesi ve kullanım amaçları verilmiştir. Gerek duyulduğunda MATLAB yardım belgeleri veya internet sayfası aracılığı ile fonksiyonların kullanımları ile ilgili açıklamalara bakılabilir.

Çizelge 3.1 Çeşitli MATLAB fonksiyonları ve kullanım amaçları

Fonksiyon adı	Kullanım amacı
unique	Dizeler içinde tekrarsız elemanları listeler
diag	Köşegen dizeler üretmek ya da bir dizelerin köşegen elemanlarını almak için kullanılır.
std	Standart sapma
mean	Ortalama ya da ortanca bulma
var	Varyans
det	Dizeler determinanı
mod	Bölümden kalan

Alıştırmalar

1. $[-2\pi, 2\pi]$ aralığı $\pi/32$ artımlarla örneklenerek oluşturulan dizelerin kaç elemanı vardır?
2. **randn** MATLAB fonksiyonu ile üreteceğiniz 5×7 boyutlarındaki dizelerin en küçük ve en büyük elemanlarını ve bunların bulunduğu satır sütun numaralarını bulunuz.
3. $p_3x^2 + p_2x + p_1 = 0$ polinomunun katsayıları $p = (G^T G)^{-1} G^T d$ şeklinde verilen dizeler işlemleri ile çözülebilir. T dizelerin dönüğünü, -1 ise tersini göstermektedir. Burada G veri çekirdek dizeyi olarak adlandırılır ve içeriği aşağıdaki gibidir:

$$G = \begin{bmatrix} 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \\ \vdots & \vdots & \vdots \\ 1 & x_{n-1} & x_{n-1}^2 \\ 1 & x_n & x_n^2 \end{bmatrix}_{n \times 3}$$

Bu bilgilere göre aşağıda verilen yatay eksen (x) değerleri için hesaplanmış verileri (d) kullanarak polinom katsayılarını hesaplayınız.

x	-3	-2.5	-2	-1.5	-1	-0.5	0	0.5	1	1.5	2	2.5	3
d	3.4	1.5	0.1	-0.8	-1.2	-1.1	-0.5	0.6	2.2	4.3	6.9	10	13.6

4. $T = \sum x_i^2$ toplamını x 'in $[0 \ 20]$ aralığındaki değerleri için 0.5 artımla hesaplayınız.
5. Üst üste konulmuş küp şeklindeki kutuların kenar uzunlukları aşağıdan yukarıya doğru

$dx=[5\ 4\ 3\ 2\ 2.5\ 2.0\ 1.6\ 1.3\ 1.0\ 0.8\ 0.6];$

şeklinde sıralanmaktadır. Her bir kutunun üst kenarının yerden yüksekliğini bulan MATLAB ifadesini yazınız.

6. $R=10 \times \text{randn}(20,1)$; ifadesi ile üretilen R dizeyinin en büyük 5 elemanını bulan MATLAB ifadelerini yazınız.
7. $f(x)=e^{-0.5x^2}$ fonksiyonunun $[-3\ 3]$ aralığındaki sayısal türevini $dx=0.125$ adım boyu için hesaplayınız.
8. X ve Y koordinatları ile verilen konumlarda yapılmış ölçümler aşağıdaki çizelgede verilmiştir. Son sütunda yer alan ölçümleri her bir sütunda bir Y konumuna ait veriler olacak şekilde yeniden yapılandırınız.

x	y	ölçüm
0	0	0.54
5	0	1.83
10	0	-2.26
15	0	0.86
20	0	0.32
0	2	-1.31
5	2	-0.43
10	2	0.34
15	2	3.58
20	2	2.77
0	4	-1.35
5	4	3.03
10	4	0.73
15	4	-0.06
20	4	0.71

9. Önceki örnekte verilen ölçüm değerlerini büyükten küçüğe doğru sıralayınız.