



# GÖRÜNTÜ İŞLEME

DERS-3



# Görüntü İşleme (Temel Matlab)



# EKRANA YAZDIRMA

**fprintf** ('çıkışta gösterilmesi istenen ifade', 'ifadenin gösterilme biçimi', değişken listesi)

'Çıkışta gösterilmesi istenen ifade': Tırnak içine hangi ifade yazılırsa ekranda bu ifade gözüktür.

'ifadenin gösterilme biçimi': Burada, ifadenin MATLAB ortamında aldığı değer için hangi formatta ekrana yazılacağını belirten kodlar kullanılır.

Bu kısımda % işaretinin arkasından hassasiyet belirleme ve dönüşüm (c,d,e,E,f,F,g,G, i,o,s,u,x,X) karakterlerinden biri kullanılır.



# EKRANA YAZDIRMA

Örneğin s; deęişken olarak 'karakter' seçildięi zaman kullanılır. Bu karakterlerin bazılarının ne ifade ettięi aőaęıdaki örneklerde gösterilmiştir. Dięer karakterler ile ilgili bilgi ise etkileşimli yardım penceresinden öğrenilebilir. Daha sonra kullanılan \n,\r,\t,\b,\f gibi tanımlar ile de (sırası ile); satırbaşı, yeni satır, sekme, geriye doęru silme ve yeni sayfa işlemleri gerçekleştirilebilir.



# EKRANA YAZDIRMA

## *fprintf*

**%a.bf** : f ifadesi sayının **sabit noktalı** olduğu anlamına gelir (default olarak short e formatındadır). f sayısı gösterim olarak 'a.b' gibi iki sayıdan meydana gelir. 'a' sayısı değişkenin aldığı değerin işareti, tam kısmı ve varsa noktayı da dahil ederek noktadan sonraki rakam sayısını içerir. 'b' sayısı ise sabit noktalı sayının noktadan sonra ekrana yazılması istenen rakam sayısını gösterir. Eğer tamsayı için kullanılacak rakam sayısı 'a', olması gerekenden az olsa bile ekrana yazılan sonuç hatalı olarak ortaya çıkmaz.

```
>> sicaklik=1056.789432;
```

```
>>fprintf('sicaklik=%4.0fderece',sicaklik)
```

```
sicaklik=1057derece
```

```
>>fprintf('sicaklik = %4.1f derece',sicaklik)
```

```
sicaklik = 1056.8 derece
```



# EKRANA YAZDIRMA

## *fprintf*

Örnek olarak 'Ahmet Sadık' adındaki bir öğrencinin adı, soyadı, okul numarası, ilgili dersi, bu dersten aldığı not alta alta ekrana yazdırılsın. Bunun için fprintf komutundan yararlanılabilir;

```
>>Ad='Ahmet';                ('enter')
>>Soyad='Sadık';            ('enter')
>>Numara='307';             ('enter')
>>Ders='Matematik';        ('enter')
>>Not=45.75;                ('enter')
>>fprintf('Ad:%s\nSoyad: %s\nNumara:%s\nDers: %s\nNot:%3.1f',...
          Ad,Soyad,Numara,Ders,Not)      ('enter')
```

Ad:Ahmet

Soyad: Sadık

Numara:307

Ders: Matematik

Not:45.8



# İLİŞKİSEL OPERATÖRLER

## • 8. KONTROL AKIŞ YAPILARI

• MATLAB bir programlama dilidir. Bu nedenle diğer programla dillerindeki temel yapılar benzer şekilde kullanılmaktadır. if-else-end, switch, for, while, continue ve break yapılarını burada da göreceğiz.

## • İlişkisel ve mantıksal işlemler

• MATLAB diğer programlar gibi ilişkisel imleçleri kullanmaktadır.

•

- < Küçük
- <= Küçük ya da eşit
- > Büyük
- >= Büyük ya da eşit
- == Eşit
- ~= Eşit değil

Operatörler:

- & and operatörü
- | or operatörü
- ~ not operatörü



# SEÇME (SELECTION) TİPİ KONTROL DEYİMLERİ

## IF DEYİMİ

**if deyimi**, Matlab dilinde **sartli dallanma (conditional branching)** adi verilen islemi gerçeklestiren bir deyimdir. Sartli dallanma, herhangi bir programlama dili için temel kontrol yapisidir. Sartli dallanma islemi sayesinde, bir program, kararlar alma imkanina kavusur; bir ifadenin sonucuna göre, bir komutlar dizisinin icra edilip edilmeyecegine karar verebilir. Ifadenin degeri, bir icradan digerine degisebilecegi için, bu özellik bir programa farkli verilere karsi farkli sekillerde davranma imkani saglar. Matlab dilinde sartli dallanma if ve else anahtar sözcükleri ile gerçeklestirilir. if deyiminin en basit sekli için yazilis biçimi asagidaki gibidir:

**if ifade**

*deyim1;*

**end**

*deyim2;*





# SEÇME (SELECTION) TİPİ KONTROL DEYİMLERİ

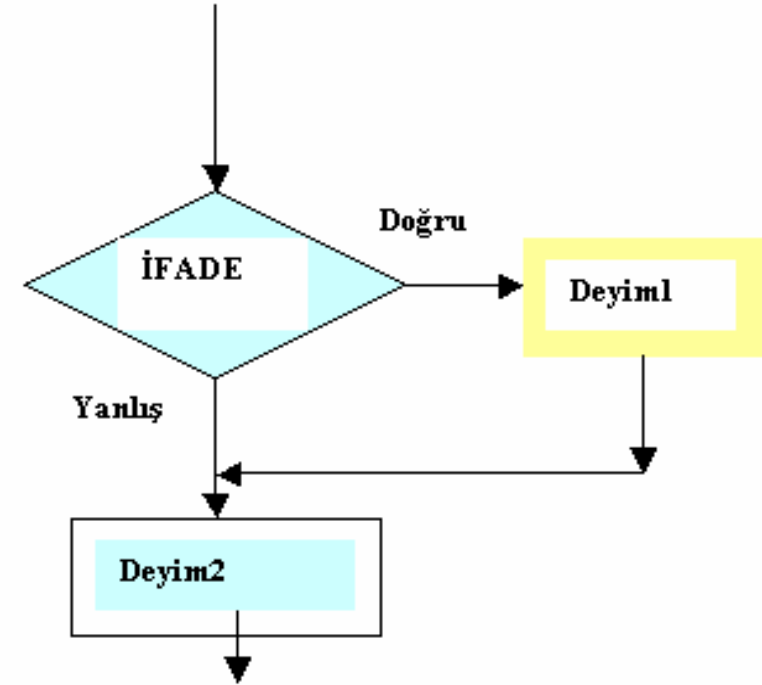
Burada ifadenin degeri dogru(true) ise deyim1 icra edilir sonra icra deyim2'ye geçer.Ifadenin degeri yanlis(false) ise bu durumda da dogrudan deyim2'ye geçilir. Ifadenin degeri yanlis ise deyim1 icra edilmeyecektir.Asagida if yapisini açıklayan bir akis diyagrami verilmistir.

**if ifade**

*deyim1;*

**end**

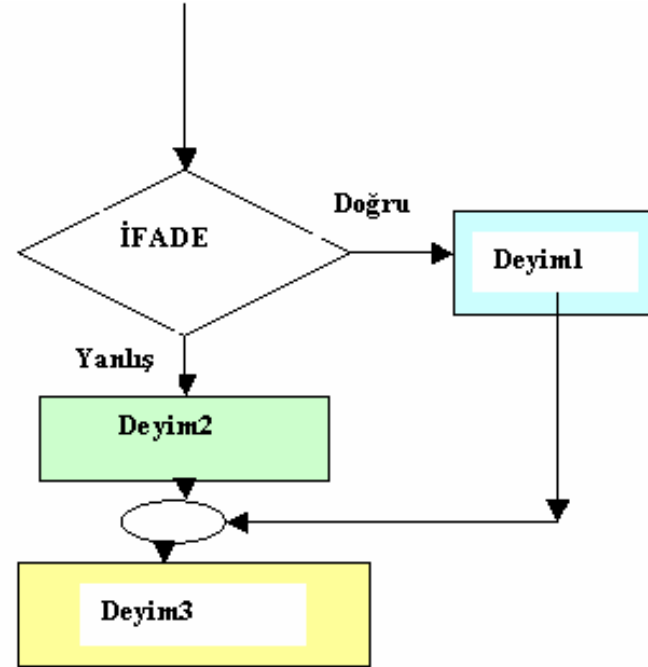
*deyim2;*



# SEÇME (SELECTION) TİPİ KONTROL DEYİMLERİ

If deyimi else sözcüğü ile birlikte kullanılırsa aşağıdaki yazılış biçimi kullanılır:

```
if ifade  
deyim1 ;  
else  
deyim2;  
end  
deyim3;  
...
```



if else deyimi için akış diyagramı (flow chart)



# SEÇME (SELECTION) TİPİ KONTROL DEYİMLERİ

- if-else-end
- 
- En basit gösterim şekli;
- 
- **if expression**
- **commands**
- ...
- **end**
- 
- Örneğin;
- 

```
» A=[1 2; -3 6];
```

```
» if det(A)>0
```

```
    Ainv=inv(A);
```

```
    disp(Ainv)
```

```
end
```

çıktısı aşağıdaki gibidir,çünkü determinant pozitif bir değerdir.

```
0.5000 -0.1667
```

```
0.2500  0.0833
```



# SEÇME (SELECTION) TİPİ KONTROL DEYİMLERİ

```
r = round(-10 + (20+10).*rand(1,8))
```

```
r = [ 5  7 -3  4 19  6  6 -3]
```

Dizinin ortalaması pozitif ise pozitif dizi yazan ifadeyi yazalım.

```
if (mean(r)>0)
    fprintf('Pozitif Dizi')
End
```

Else yapısı ekleyelim.

```
r = round(-10 + (-1+10).*rand(1,8))
```

```
if (mean(r)>0)
    fprintf('Pozitif Dizi')
else
    fprintf('Negatif Dizi')
end
```



# SEÇME (SELECTION) TİPİ KONTROL DEYİMLERİ

- if-else-end yapısının en genel hali aşağıdaki gibidir.
- 
- **if expression-1**
- **commands-1**
- **elseif expression-2**
- **commands-2**
- .
- .
- .
- **elseif expression-(n-1)**
- **commands-(n-1)**
- **else**
- **commands-n**
- **end**



# SEÇME (SELECTION) TİPİ KONTROL DEYİMLERİ

```
a=input('a değerini giriniz:');
```

```
if a == 10
    % IF şartı doğruysa aşağıdakini yaz
    fprintf('Girilen değer= 10\n' );
elseif( a == 20 )
    % ELSEIF şartı doğruysa aşağıdakini yaz
    fprintf('Girilen değer= 20\n' );
elseif a == 30
    % ELSEIF şartı doğruysa aşağıdakini yaz
    fprintf('Girilen değer= 30\n' );
else
    % Hiçbir şart doğru değil ise '
    fprintf('Hiçbir değer eşleşmedi\n');
fprintf('Girilen değer: %d\n', a );
end
```



# SEÇME (SELECTION) TİPİ KONTROL DEYİMLERİ

## switch DEYİMİ

Bir seçim söz konusu olduğunda bu yapı kullanılabilir. Genel gösterimi:

### switch (selector)

case label-1

commands-1

case label-2

commands-2

.

.

.

case label-n

commands-n

otherwise

commands-m

end



# SEÇME (SELECTION) TİPİ KONTROL DEYİMLERİ

- Örnek:
- 
- `>> a=[4 5;2 3];`
- `>> switch (det(a))`
- `case 1`
- `b=a';`
- `disp(b)`
- `case 2`
- `b=a*a;`
- `disp(b)`
- `end`
- 
- a matrisi determinantı 2 ye eşit olduğu için çıktısı aşağıdaki gibi olacaktır.
- 
- 26   35
- 14   19





# DÖNGÜLER

## 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;

- `k=1;`
- `for num=[6 37 23 -1]`
- `fprintf('%d nci eleman:%d\n', k, num)`
- `k=k+1;`
- `end`

Buraya Dikkat!

çıktısı;

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



# DÖNGÜLER

**Rastgele bir dizideki asal sayıları yazdırınız.**

```
clc; clear;
for r = round(0 + (200+0).*rand(1,10));
    if isprime(r)
        fprintf('%d asal sayıdır.\n',r );
    else
        fprintf('%d asal sayı değildir!\n',r );
    end
end
```

Bu şekilde indis bilgilerini alamayız!



# DÖNGÜLER

**Rastgele bir dizideki asal sayıları yazdırınız.**

```
clc; clear;
r = round(10 + (200+10).*rand(1,10));
i=1;
for i=1:length(r)
    if isprime(r(i))
        fprintf('%d nci eleman %d asal sayidir.\n',i,r(i) );
    else
        fprintf('%d nci eleman %d asal sayi degildir.\n',i,r(i) );
    end
end
end
```

İndis bilgileri alınabilir.



# Görüntü İşleme (Temel Matlab)

## For ve if örnekleri

$x^4 - 0,5x^3 - 8x^2 - 1,5x + 9$  fonksiyonunun köklerini bulunuz.

### 1. yol:

```
clc;clear;
for x=-5:0.1:5
    t=x^4 - x^3/2 - 8*x^2 - (3*x)/2 + 9;

    if (t==0)
        fprintf('Kök=%2.5f',t);
        fprintf('Değerler=%2.5f\n',x);
    end
end
```



# Görüntü İşleme (Temel Matlab)

2. yol:

```
clc;clear;
x=-5:0.1:5;
t=x.^4 - x.^3/2 - 8*x.^2 - (3*x)/2 + 9;
for i=1:length(t)
    if (t(i)==0)
        fprintf('Kök=%2.5f ',t(i));
        fprintf('değer=%2.5f\n',x(i));
    end
end
```



# Görüntü İşleme (Temel Matlab)

$$f(x) = 2x - e^{x/2}$$

Fonksiyonun köklerini döngüler kullanarak bulunuz. Bu fonksiyonu çiziniz.



# 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 resmin siyah ve beyaz piksellerini saydırma

```
b=imread('bayrak_BW.jpg');  
imshow(b);  
beyazlar=0;  
for i=1:size(b,1)  
    for j=1:size(b,2)  
        if b(i,j)>0  
            beyazlar=beyazlar+1;  
        end  
    end  
end  
fprintf('Beyaz Piksel Sayısı:%d\n',beyazlar);
```





# DÖNGÜLER

## while döngüleri

Genel gösterimi;  
**while expression  
commands**

....

**end**

şeklindedir.



# DÖNGÜLER

Bir resim dosyasında değeri 150 ile 180 arasında olan ve arka arkaya sıralı 5 pikselin koordinatlarını bulan programı while döngüsü ile yazınız.

|     |     |     |     |     |     |     |     |
|-----|-----|-----|-----|-----|-----|-----|-----|
| 17  | 143 | 192 | 161 | 101 | 115 | 52  | 116 |
| 39  | 188 | 97  | 74  | 70  | 246 | 104 | 138 |
| 231 | 151 | 113 | 226 | 151 | 0   | 228 | 93  |
| 80  | 3   | 164 | 214 | 226 | 222 | 102 | 176 |
| 57  | 68  | 71  | 245 | 30  | 193 | 70  | 253 |
| 179 | 92  | 230 | 47  | 172 | 156 | 56  | 174 |
| 131 | 164 | 87  | 25  | 90  | 7   | 170 | 220 |
| 224 | 30  | 64  | 148 | 109 | 16  | 205 | 227 |
| 48  | 70  | 67  | 106 | 212 | 60  | 208 | 217 |
| 220 | 181 | 94  | 18  | 246 | 200 | 132 | 28  |
| 217 | 35  | 169 | 178 | 180 | 121 | 244 | 194 |
| 10  | 55  | 255 | 169 | 156 | 185 | 63  | 131 |
| 63  | 221 | 133 | 152 | 160 | 102 | 165 | 226 |
| 8   | 97  | 53  | 163 | 138 | 201 | 95  | 119 |
| 244 | 60  | 85  | 178 | 57  | 50  | 138 | 11  |
| 191 | 204 | 79  | 88  | 248 | 79  | 125 | 29  |
| 43  | 209 | 23  | 69  | 114 | 200 | 5   | 57  |
| 162 | 26  | 88  | 59  | 117 | 27  | 121 | 200 |
| 171 | 129 | 237 | 239 | 171 | 100 | 75  | 40  |
| 249 | 183 | 42  | 214 | 19  | 53  | 250 | 249 |



# DÖNGÜLER

```
clc;clear;
a=imread('rastgeleResim.jpg');
i=1;
sayac=0;
while (i<=size(a,1)*size(a,2))
    if (a(i)>=150 & a(i)<=180)
        sayac=sayac+1;
    else sayac=0;
    end
    if sayac==5
        sutun=ceil(i/size(a,2)); %bir fazlasına yuvarlar.
        if (rem(i,size(a,1))~=0)
            satir=rem(i,size(a,1));
        else
            satir=size(a,1);
        end
        fprintf('Oruntu bulundu! Satir=%d Sütun=%d i=%d\n',satir,sutun,i);
    end
    i=i+1;
end
```



# DÖNGÜLER

## break komutu

break yapısı döngünün dışına ulaşmak için kullanılır.

```
» y=zeros(1,2);  
» x=[2 3];  
» while x>=0  
    y=y+x;  
    x=x-1;  
    if y>=3  
        break  
    end  
    disp(y)  
end
```

çıktısı;

2 3

(3, 5) vektörü görünmedi çünkü  $y \geq 3$  ifadesi doğru bir ifade.



# DÖNGÜLER

continue

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

```
>> y=zeros(1,2);  
>> x=[2 3];  
>> while x>=0  
    y=y+x;  
    x=x-1;  
    if y<=3  
        continue  
    end  
    disp(y)  
end
```

çıktısı;

3 5

3 6

(2, 3) vektörü görüntülenmedi.  $y \leq 3$  ifadesi doğru olduğu için sıradaki iterasyonu atladı.



# 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)

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)

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

```
sayi=input('Bir sayı giriniz: ');  
fact=1;  
if sayi<0  
    fprintf('\nNegatif sayı girdiniz!\n\n');  
else  
    for i=1:sayi  
        fact=fact*i;  
    end  
fact  
end
```





# Görüntü İşleme (Temel Matlab)

Bir resmin tek sayı olan sütunlarını for döngüsü ile siyah yapınız.

```
clc;clear;
resim=imread('cameraman.tif');
imshow(resim)
for i=1:size(resim,1)
    for j=1:size(resim,2)
        if mod(j,2)~=0
            resim(i,j)=0;
        end
    end
end
figure
imshow(resim);
```



# 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
```



# 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 edilen 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 arasindaki 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 uzaklik=%f', uzaklik);
```

## ANA PROGRAM:

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



# Fonksiyon yapıları

Alt sınır ve Üst sınırları verilerek Rastgele sayı üreten bir fonksiyon yazınız.

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

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

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

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