

*Dr. Yüksel YURTAY*



# Sayısal Analiz

Algoritma & Matlab

İletişim :

[yyurtay@sakarya.edu.tr](mailto:yyurtay@sakarya.edu.tr)

[www.cs.sakarya.edu.tr/yyurtay](http://www.cs.sakarya.edu.tr/yyurtay)

(264) 295 58 99

## Algoritma

Algoritma ; verilerin bilgisayara hangi çevre biriminden girileceğinin, problemin nasıl çözüleceğinin, hangi basamaklardan geçirilerek sonuç alınacağını, sonucun nasıl ve nereye yazılacağını sözel olarak ifade edilmesi biçiminde tanımlanabilir.

### Örnek :

Verilen iki sayının toplamının bulunmasının algoritması aşağıdaki gibi yazılır:

### Algoritma :

- Adım 1-Başla
- Adım 2-Birinci sayıyı oku
- Adım 3-İkinci sayıyı oku
- Adım 4-İki sayıyı topla
- Adım 5-Dur

## PROGRAMLAMA (PROGRAM GELİŞTİRME) NEDİR?

*Programlama, herhangi bir problemin bir programlama dili kullanılarak çözülmesi için yazılan mantıksal kod bloklarına verilen addır.*

Amaç problemin çözümüne uygun şekilde hazırlanan program kodu ile problemi çözmeye çalışmaktır. Bu amaç için araç olarak herhangi bir programlama dilini kullanırız.

Programlama diline ait hazır komutları kullanarak problemi çözmeye çalışırız. Bu komutlar programlama dilleri arasında farklılık göstermesine rağmen programlama mantığı bütün dillerde aynıdır.

Unutulmamalıdır ki hazırlanan bir program, gerektiğinde başkaları tarafından da kullanılacaktır. Bu nedenle hazırlanan programın mümkün olduğunca hatalardan arındırılmış olması gerekmektedir. Beklenen sonuçları verecek şekilde hazırlanmış olması gerekmektedir.

BSM

1.  
Hafta

3.  
Sayfa

## PROGRAMLAMA (PROGRAM GELİŞTİRME) NEDİR?

Bir programlama dilinde hazırlanmış bir program çalıştırılırken genellikle şu iki tür hata ile karşılaşılır:

1. Yazım hataları,
2. Mantıksal hatalar.

**Yazım Hataları**, *programın derlenmesi* sırasında ortaya çıkar ve hata düzeltilmedikçe program çalıştırılamaz.

**Mantıksal Hatalar**, yazım hataları gibi programın yazımından kaynaklanan hatalar değildir. Bunlar *programın çalıştırılması* sırasında ortaya çıkar ve programdan istenen sonucun alınamamasına veya yanlış sonuçlar verilmesine neden olur.

Programın hazırlanmasında dikkat edilmesi gereken en önemli konu, problemin iyi anlaşılması, iyi analiz edilmesidir. Unutulmamalıdır ki bilgisayar sadece programcının vermiş olduğu işlemleri yerine getirir.

## PROGRAMLAMA (PROGRAM GELİŞTİRME) NEDİR?

Programlamanın (Program Geliştirmenin) genel yapısı sırasıyla şu adımları kapsar:

1. Problemin tanımlanması,
2. Problemin çözümlenmesi,
  - 2.1. Çözüm yolunun belirlenmesi,
  - 2.2. Çözüm yoluna uygun algoritmanın belirlenmesi,
  - 2.3. Algoritmaya uygun akış diyagramının çıkarılması,
  - 2.4. Algoritmayı gerçekleştirecek uygun programlama dilinin seçilmesi,
3. Problemin programlama dili komut seti yardımıyla kodlanması,
4. Hazırlanan programın denenmesi ve belgelendirilmesi.



## PROBLEMİN TANIMLANMASI

Bir problemin herhangi bir programlama dilinde kodlanmasına başlanmadan önce problemin tam olarak anlaşılması gerekmektedir. Aksi halde yanlış çözüm kaçınılmazdır.

### Problemin Çözümlemesi

#### Çözüm Yolunun Belirlenmesi

- Giriş verilerinden sonuçta elde edilecek verilere nasıl, hangi yolla ulaşılabileceğinin tespiti gerekir.
- Bu durumun iyi analiz edilmesi gerekir.
- Problemin matematiksel modeli bu aşamada belirlenir.
- Hangi tekniğin en uygun olduğuna programcının bilgisi ve tecrübesi etki eder.

## PROBLEMİN TANIMLANMASI

### *Örnek:*

Aranan bir büyüklüğün herhangi bir  $\{a\}$  kümesi içerisinde olup olmadığının araştırılması.

$\{a\} = \{3, 7, -10, 8, 1, -4, -94, 6, 2, -1, 34, 14, 78, -19, 99\}$  olsun.

$x = -12$  elemanının bu küme içinde yer alıp almadığını arayalım.

### *Çözüm Yolları:*

1. Verilen  $x$  değeri (-12) sırayla  $\{a\}$  kümesinin bütün elemanları ile tek tek karşılaştırılarak arama yapılabilir.

## PROBLEMİN TANIMLANMASI

### *Örnek:*

Aranan bir büyüklüğün herhangi bir  $\{a\}$  kümesi içerisinde olup olmadığının araştırılması.

$\{a\} = \{3, 7, -10, 8, 1, -4, -94, 6, 2, -1, 34, 14, 78, -19, 99\}$  olsun.

$x = -12$  elemanının bu küme içinde yer alıp almadığını arayalım.

### *Çözüm Yolları:*

2. Önce  $\{a\}$  kümesi kendi içerisinde artan sırada (büyükten küçüğe doğru) sıralanır.

$\{a\} = \{-94, -19, -10, -4, -1, 1, 2, 3, 6, 7, 8, 14, 34, 78, 99\}$

Daha sonra verilen  $x$  değeri  $(-12)$  sıralanmış  $\{a\}$  kümesi içerisinde baştan bütün elemanlar ile karşılaştırılarak arama yapılabilir. En son karşılaştırılan değer  $x$  değerinden büyük ise işlem kesilir.



## PROBLEMİN TANIMLANMASI

### *Çözüm Yolları:*

3. Önce  $\{a\}$  kümesi kendi içerisinde artan sırada (büyükten küçüğe doğru) sıralanır.

$$\{a\} = \{-94, -19, -10, -4, -1, 1, 2, 3, 6, 7, 8, 14, 34, 78, 99\}$$

$\{a\}$  kümesindeki eleman sayısı ikiye tam bölünür ve orta eleman bulunur.  $x$  değeri  $(-12)$  orta elemanla karşılaştırılır. Eğer orta eleman  $x$  değerinden büyük ise  $x$  değeri  $\{a\}$  kümesinin ilk yarısında olacaktır. İlk yarıdaki eleman sayısı ikiye tam bölünerek 2. orta eleman bulunur.  $x$  değeri 2. orta elemanla karşılaştırılır.

Eğer 2. orta eleman  $x$  değerinden büyükse  $x$   $\{a\}$  kümesinin 2. yarısının ilk bölümünde olacaktır. Bu işlemler tek eleman kalıncaya kadar sürdürülür. Arama sonlandırılır.

$$\{a\}' = \{-94, -19, -10, -4, -1, 1, 2, 3\}$$

$$\{a\}'' = \{-94, -19, -10, -4\}$$

$$\{a\}''' = \{-10, -4\}$$

BSM

1.  
Hafta

9.  
Sayfa

## PROBLEMİN TANIMLANMASI

### *Çözüm Yolları:*

Yukarıda belirtilen 3 farklı çözüm yolu problemin çözümünü sağlamaktadır.

Bu çözüm yollarından hangisin seçileceği;

- 1. Verinin büyüklüğüne,**
- 2. Amaçlanan işlem hızına,**
- 3. Yapılacak işlemin tekrarlanma sayısına bağlıdır.**

## Akış Şemaları (Diyagramları)

Herhangi bir sorunun çözümü için izlenmesi gerekli olan aritmetik ve mantıksal adımların söz veya yazı ile anlatıldığı algoritmanın, görsel olarak simge ya da sembollerle ifade edilmiş şekline "akış şemaları" veya FLOWCHART adı verilir.

### Örnek :

Adım 1-Başla

Adım 2-A'yı oku

Adım 3-B'yi oku

Adım 4- $D=A+B$

Adım 5- $E=A*B$

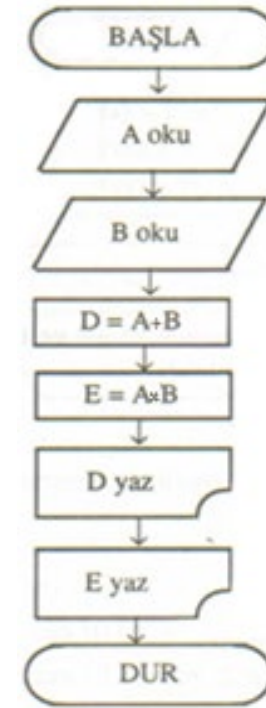
Adım 6-D'yi yaz

Adım 7-E'yi yaz








Adım 8-Dur



### Akış Şeması (FLOWCHART)



Akış şemalarının hazırlanmasında aşağıda yer alan simgeler kullanılır.

	Algoritmanın başladığını ya da sona erdiğini belirtmek için kullanılır.
	Klavye aracılığı ile giriş ya da okuma yapılacağını gösterir.
	Yazıcı(printer) aracılığı ile çıkış yapılacağını gösterir.
	Kart okuyucu aracılığıyla giriş yapılacağını gösterir.
	Araç belirtmeden giriş ya da çıkış yapılacağını gösterir.
	Hesaplama ya da değerlerin değişkenlere aktarımını gösterir.
	Aritmetik ve mantıksal ifadeler için karar verme ya da karşılaştırma durumunu gösterir.

Akış şemaları **içerik** ve **biçimlerine** göre genel olarak üç grupta sınıflandırılabilirler.

1. Doğrusal Akış Şemaları,
2. Mantıksal Akış Şemaları,
3. Döngüsel Akış Şemaları,

## Doğrusal Akış Şemaları

İş akışları, giriş, hesaplama, çıkış biçiminde olan akış şemaları bu grup kapsamına girer.

### Değişkenler

Akış Şeması

A: Birinci sayı B: İkinci sayı

D: İki sayının toplamını ( $A+B$ )

E: İki sayının bölümünü ( $A*B$ )

### Algoritma

Adım 1-Başla

Adım 2-A'yı oku

Adım 3-B'yi oku

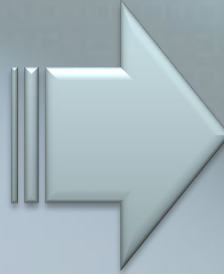
Adım 4- $D=A+B$

Adım 5- $E=A*B$

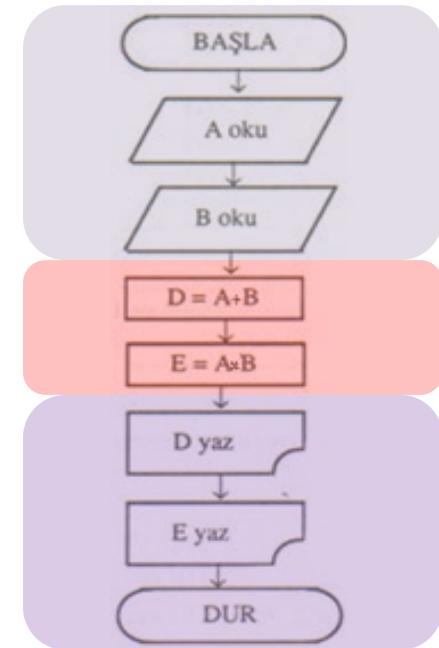
Adım 6-D'yi yaz

Adım 7-E'yi yaz

Adım 8-Dur



Akış Şeması

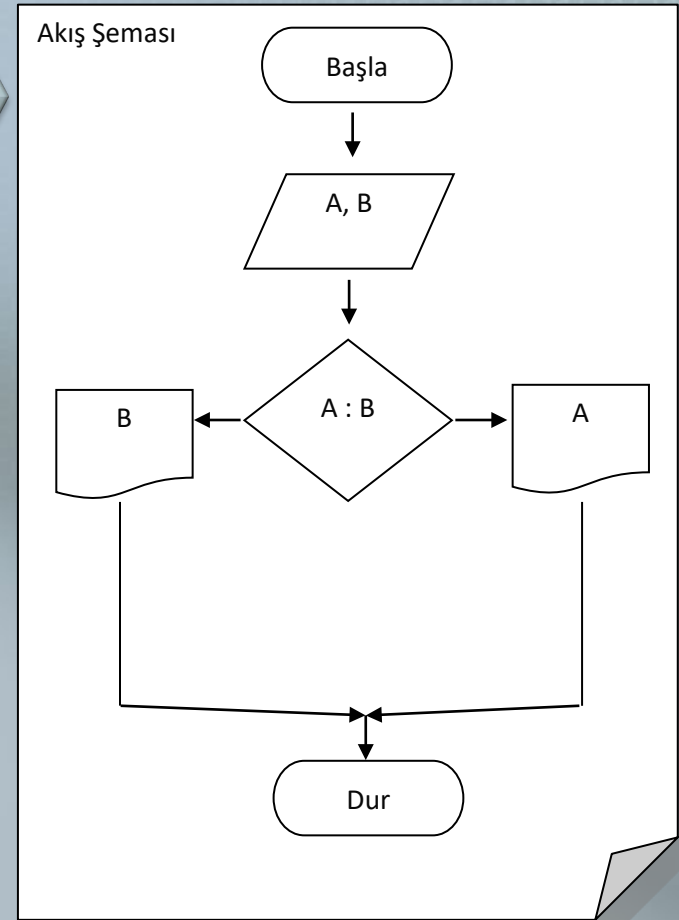


## Mantıksal Akış Diyagramları

Mantıksal kararları içeren akış diyagramlarıdır

### Algoritma

Adım 1-Başla  
Adım 2-A,B'yi oku  
Adım 3- $A > B$  ise Adım 5'ya git  
Adım 4- $A < B$  ise Adım 6ya git  
Adım 5-A'yi yaz Adım 7'e git  
Adım 6-B'yı yaz Adım 7'e git  
Adım 7-Dur



BSM

1.  
Hafta

14.  
Sayfa



## Döngülü Akış Diyagramları

Akış sürecinde yer alan herhangi bir adım ya da aşamanın birden fazla kullanıldığı akış diyagramlarına denir.

### Örnek :

N sayısını ekrandan okutarak faktöriyelini hesaplayıp yazan programın algoritma ve akış diyagramını oluşturalım.

### Değişkenler

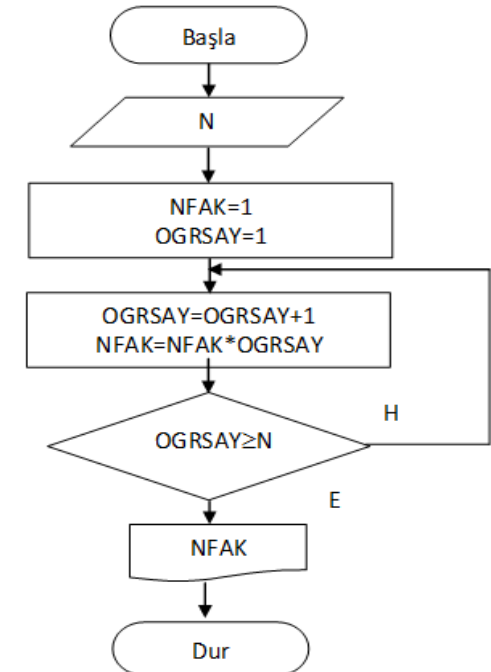
NFAK=N faktöriyel (N!) değerini,  
OGRSAY=1'den N'e kadar sayıları gösterecek,  
NFAK=1\*2\*.....\*N

### Algoritma

Adım 1-Başla  
Adım 2-N'i ekrandan oku  
Adım 3-NFAK=1  
Adım 4-OGRSAY=1  
Adım 5-OGRSAY=OGRSAY+1  
Adım 6-NFAK=NFAK\*OGRSAY  
Adım 7-Eğer OGRSAY ≥ N  
Adım 8-NFAK yaz  
Adım 9-Dur



Akış Diyagramı



## Matlab

MATLAB (**MAT**rix **LAB**oratory);

İlk defa 1985'de C.B Moler tarafından matematik ve özellikle de matris esaslı matematik ortamında kullanılmak üzere geliştirilmiş etkileşimli bir paket programlama dilidir.

### Kullanımlı Alanları :

- ✓ Sayısal işaret işleme,
- ✓ Kontrol tasarımı,
- ✓ Test ölçüm,
- ✓ Finansal modelleme ve analiz,
- ✓ Haberleşme
- ✓ ...

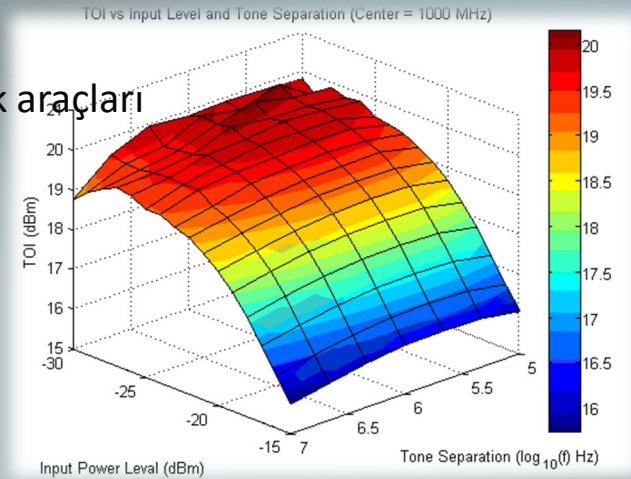
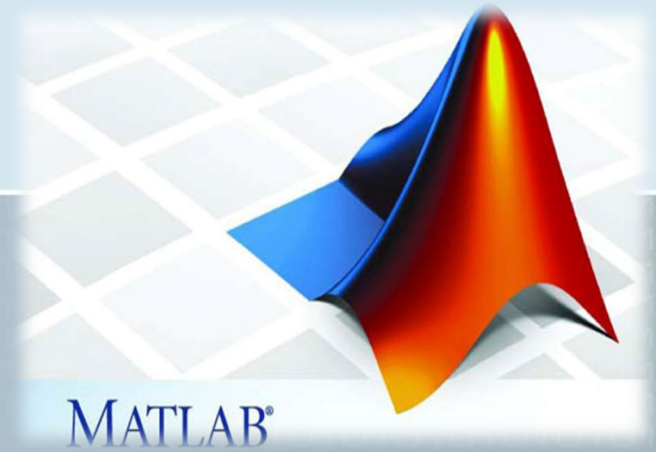
The MATLAB logo is displayed in a blue serif font. To its right is a 3D surface plot with a color gradient from blue to red, set against a background of binary code (0s and 1s).

MATLAB®

## Matlab

### Temel Özellikleri:

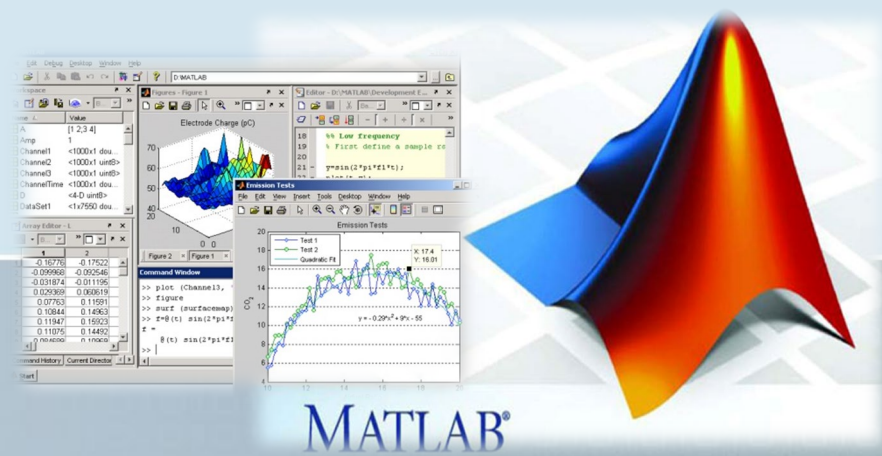
- Teknik hesaplamalar için yüksek seviyeli bir dil
- Kodların , dosyaların ve verilerin düzenlenmesi için bir geliştirme ortamı
- İteratif tasarım ve problem çözme yöntemleri için interaktif araçlar
- Lineer cebir, istatistik, Fourier analizi, filtreleme, optimizasyon ve sayısal integrasyon için matematik fonksiyonlar
- Verilerin görselleştirilmesi için 2 ve 3 boyutlu grafik araçları
- Grafik arayüzler tasarlamak için araçlar



## Matlab

### Kullanım yerleri :

- Denklem takımlarının çözümü, doğrusal ve doğrusal olmayan diferansiyel denklemlerinin çözümü, integral hesabı gibi sayısal hesaplamalar,
- Veri çözümleme işlemleri,
- İstatistiksel hesaplamalar ve çözümlemeler,
- Grafik çizimi ve çözümlemeler,
- Bilgisayar destekli denetim sistemi tasarımı.



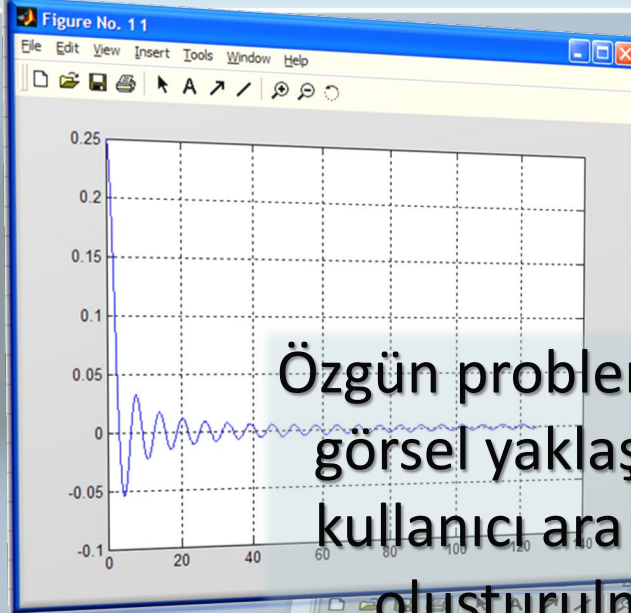
BSM

1.  
Hafta

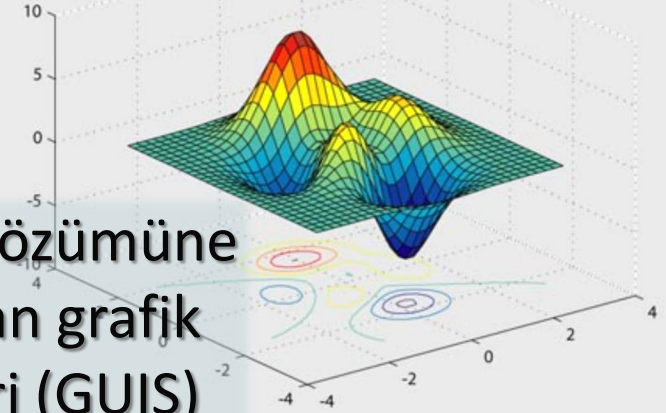
18.  
Sayfa



## Matlab



Özgün problemlerin çözümüne  
görsel yaklaşım sunan grafik  
kullanıcı ara birimleri (GUIs)  
oluşturulmasına olanak  
sağlayan bir uygulama  
geliştirme platformudur.

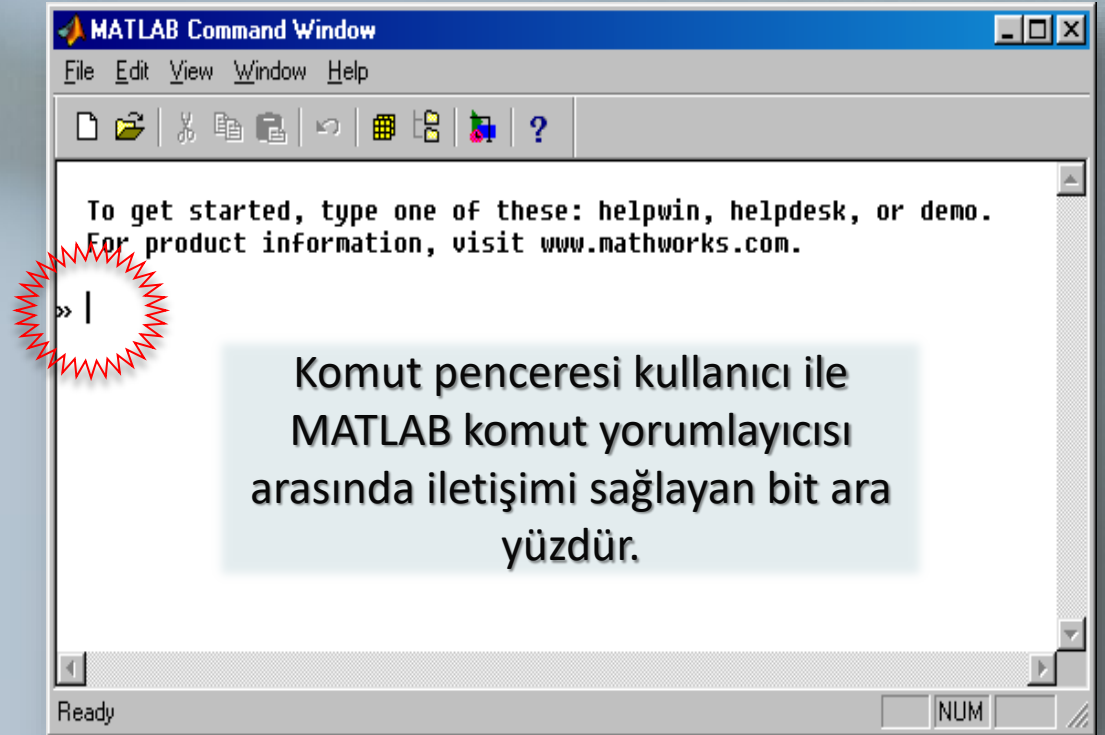
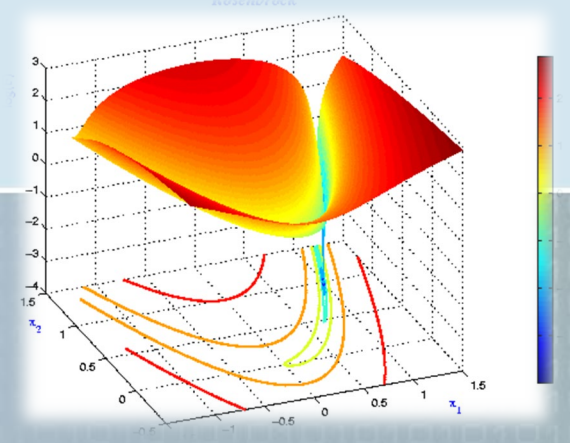


BSM

1.  
Hafta

19.  
Sayfa

## MATLAB komut penceresi



BSM

1.  
Hafta

20.  
Sayfa



## Matlab / Örnek

```
» 3+7  
ans = 10  
» disp(3+7)  
10  
»
```

a

```
» [ 5 , 2 ]  
ans = 5 2  
» disp( [ 3 , 8 ] )  
3 8  
» disp( [3 8 ] )  
3 8  
»
```

b

```
» a=[2 1 -1]  
a = 2 1 -1  
» b=[4 -2 3]  
b = 4 -2 3  
» a+b  
ans = 6 -1 2  
» a-b  
ans = -2 3 -4  
»
```

c

BSM

1.  
Hafta

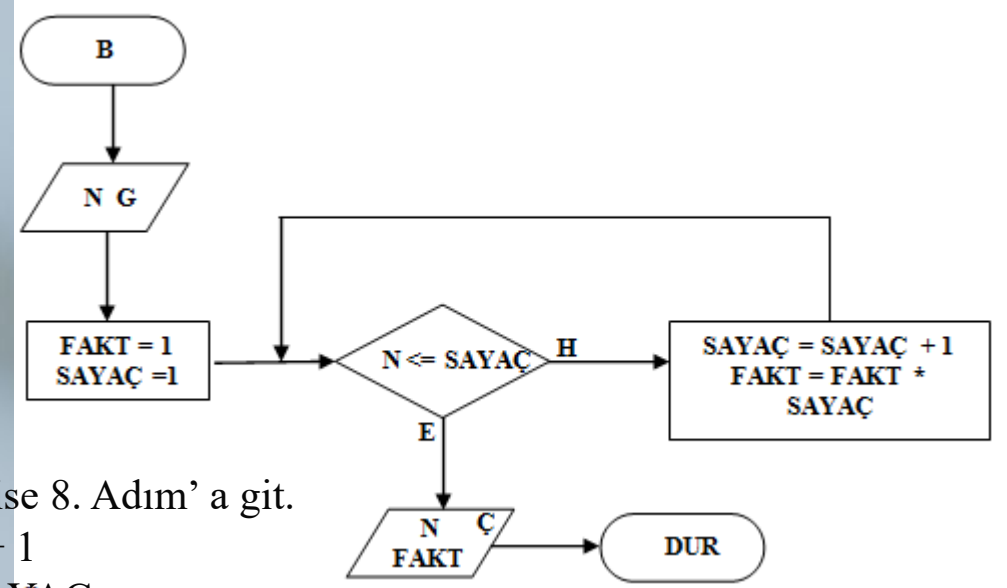
21.  
Sayfa

## Uygulama 1

Girilen bir sayının faktöriyelini hesaplayan algoritmayı yazarak ve akış diyagramını çiziniz.

### Algoritma :

1. Adım: N sayısını al.
2. Adım: FAKT = 1 olsun.
3. Adım: SAYAÇ = 1 olsun.
4. Adım: Eğer  $N \leq \text{SAYAÇ}$  ise 8. Adım' a git.
5. Adım:  $\text{SAYAÇ} = \text{SAYAÇ} + 1$
6. Adım:  $\text{FAKT} = \text{FAKT} * \text{SAYAÇ}$
7. Adım: 4. Adım' a git.
8. Adım: N yaz.
9. Adım: FAKT yaz.
10. Adım: DUR.



BSM

1.  
Hafta

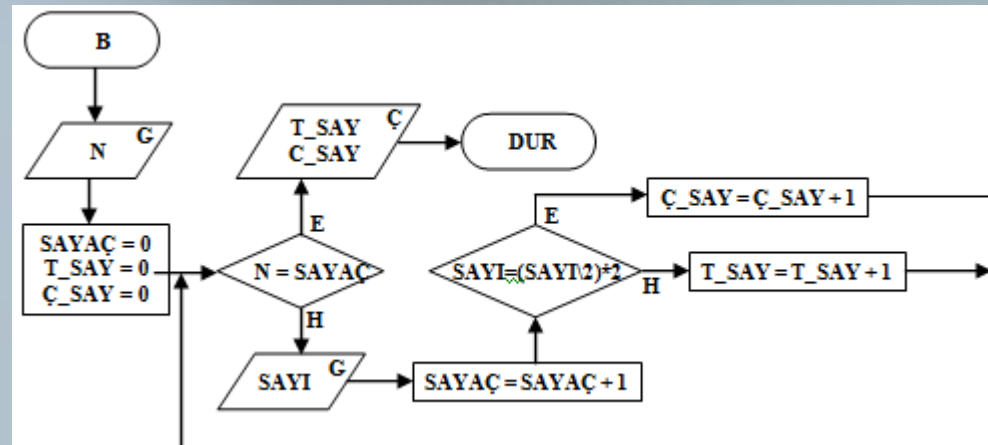
22.  
Sayfa

## Uygulama 2

İsteğe bağlı sayıda girilen sayıların içinden tek ve çift olanların sayısını bulduran algoritmayı yazarak ve akış diyagramını çiziniz.

### Algoritma :

1. Adım: N sayısını al.
2. Adım: SAYAÇ = 0 olsun.
3. Adım: T\_SAY = 0 olsun.
4. Adım: Ç\_SAY = 0 olsun.
5. Adım: Eğer  $N = \text{SAYAÇ}$  ise 11. Adım' a git.
6. Adım: Bir SAYI al.
7. Adım:  $\text{SAYAÇ} = \text{SAYAÇ} + 1$
8. Adım: Eğer  $\text{SAYI} = (\text{SAYI} \setminus 2) * 2$  ise  $\text{Ç\_SAY} = \text{Ç\_SAY} + 1$
9. Adım:  $\text{T\_SAY} = \text{T\_SAY} + 1$
10. Adım: 5. Adım' a git.
11. Adım: T\_SAY yaz.
12. Adım: Ç\_SAY yaz.
13. Adım: DUR.



BSM

1.  
Hafta

23.  
Sayfa

## Uygulama 3

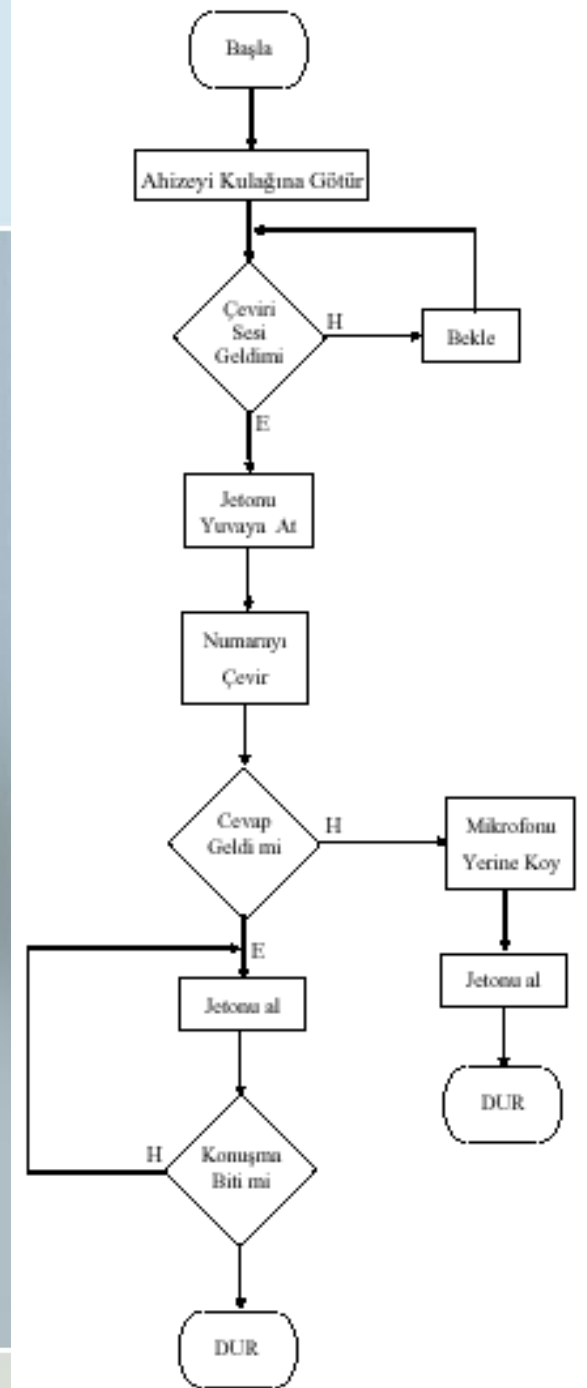
### Telefonla Görüşmenin Algoritması

1. Ahizeyi kulagınıza götürünüz
2. Çevir sesinin gelmesini bekle
3. Ses gelmisse (E) jetonu ilgili yuvaya at
4. Ses gelmemisse bekle
5. Ses gelirken jeton yerine yerlesince numarayı çevir
6. Cevap geldi mi?
7. Hayır ise ahizeyi yerine koy
8. Jetonu iade çıkısından al
9. Cevap gelmsse konuş
10. Konuşma biitt mi?
11. Evet ise Ahizeyi yerine koy
12. Hayır ise konuşmaya devam et.

BSM

1.  
Hafta

24.  
Sayfa



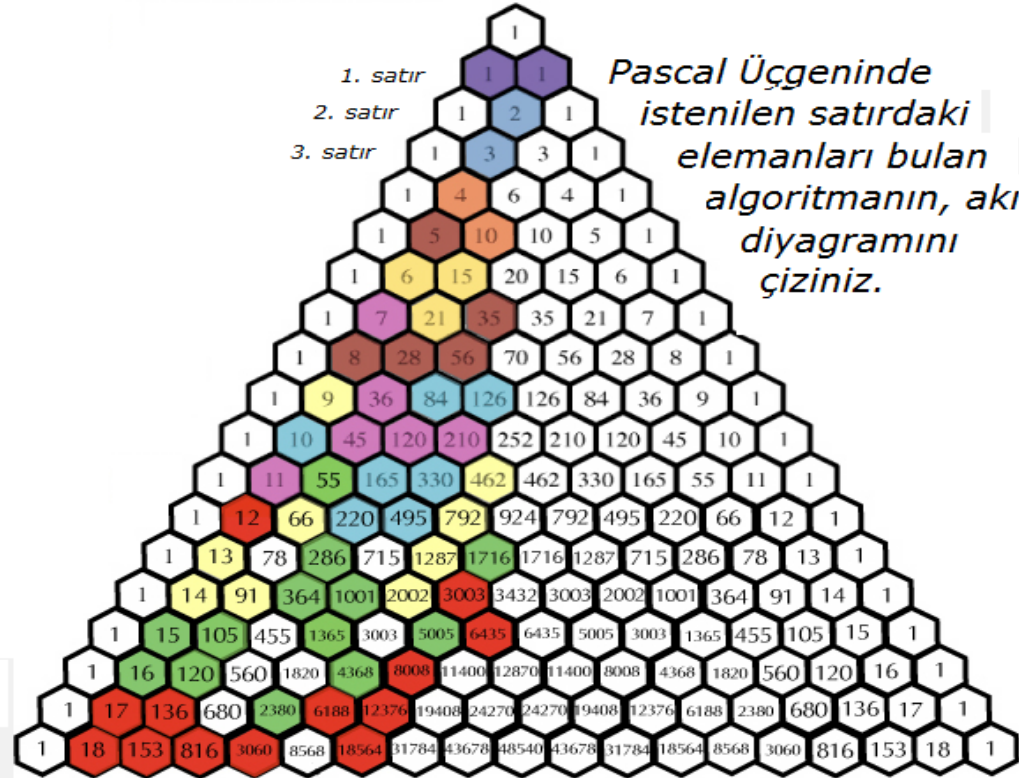
## Uygulama Ödevi

BSM

1.  
Hafta

25.  
Sayfa

RELATION PASCAL TRIANGLE

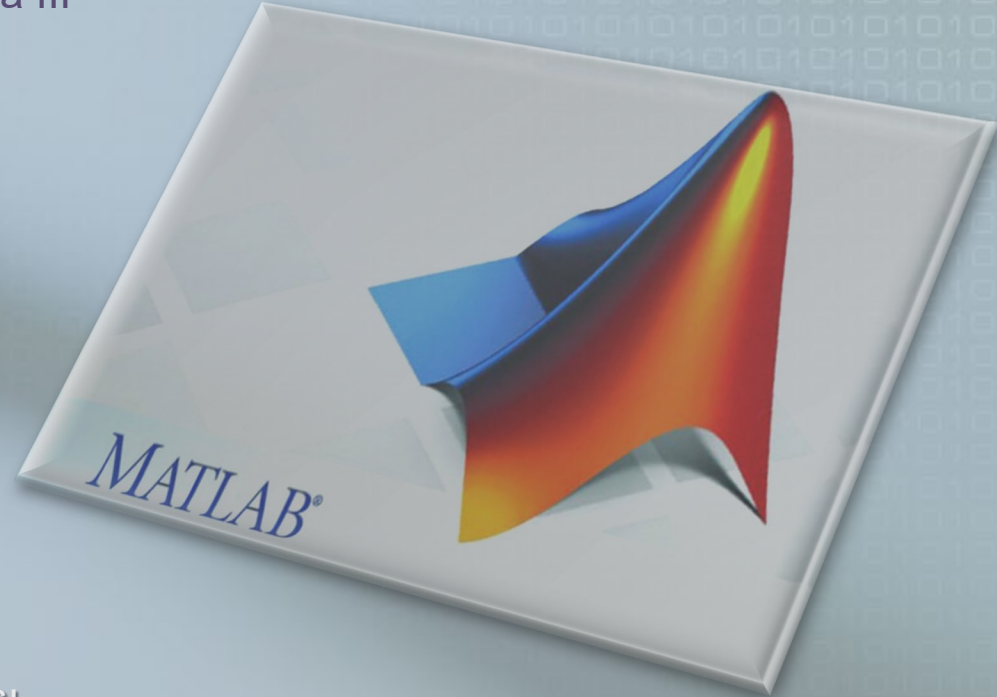


*Pascal Üçgeninde  
istenilen satırdaki  
elemanları bulan  
algoritmanın, akış  
diyagramını  
çiziniz.*

Pascal üçgeni, binom açılımındaki katsayıları bulmaya yarar. Pascal'ın bu üçgeni, olasılıklar kuramında da ustalıkla kullanılır. Bu üçgen, biyolojideki uygulamalar, matematik, istatistik ve pek çok modern fizik konularında uygulama alanı bulur. (Bazı kaynaklara göre eski Çinliler de üçgeni tanımışlar; bazıları Pascal üçgeni diye aslında bir Hayyam üçgenidir söylemişler.)

## Kaynaklar

Sayısal Analiz ve Programlama III  
Dr.Ü.Dikmen



Sonraki Hafta :

Algoritma Kurulması ...

BSM

1.  
Hafta

26.  
Sayfa