

Algoritma ve Programlamaya Giriş



mustafacosar@hitit.edu.tr

<http://web.hitit.edu.tr/mustafacosar>

İçerik

- **Algoritma**
- **Akış Diyagramları**
- **Programlamada İşlemler**
 - Matematiksel
 - Karşılaştırma
 - Mantıksal
- **Programlama Yapıları**
 - Değişkenler
 - Döngüler
 - Koşullu İfadeleri
 - Döngü ve Koşullu İfadeleri
 - Sayaç
 - Diziler
 - Matrisler

algoritma

- Adım adım işlem basamaklarının tasarlanmasıdır.
- Algoritmalar bir görevi yerine getirmek için sonlu sayıdaki işlemler dizisidir.
- Her algoritma aşağıdaki kriterleri sağlamak zorundadır.
 1. **Girdi:** Sıfır veya daha fazla değer dışarıdan verilmelidir.
 2. **Çıktı:** En azından bir değer üretmeli.
 3. **Açıklık:** Her işlem açık olmalı ve farklı anlamlar içermelidir.
 4. **Sonluluk:** Her türlü olasılık için algoritma sonlu adımda bitmeli.
 5. **Etkinlik:** Her komut kişinin kalem ve kağıt ile yürütülebileceği kadar basit olmalıdır.

örnek bir algoritma

■ Örnek1: iki sayının toplamını bulan algoritmayı hazırlayınız.

- Adım0:Başla
- Adım1: Toplamı, birinci ve ikinci sayıları tanımla
- Adım2: Birinci sayıyı oku
- Adım3: İkinci sayıyı oku
- Adım4: Sayıları topla ve toplam değişkenine at
- Adım5: Toplamı görüntüle
- Adım6:Dur

- S1 ve S2
- S1=?
- S2=?
- $T=S1+S2$
- T

programlama

- **Program:** belirli bir problemi çözmek için bir bilgisayar dili kullanarak yazılmış deyimler dizisidir.
- Bir problemi bilgisayar ile çözmek için geliştireceğimiz programın yazımında izleyeceğimiz adımlar.
 - Problemin ne olduğunun kavranması
 - Çözüm için gereksinimlerin belirlenmesi
 - Problemin girdi, çıktı ve işlemlerin belirlenmesi
 - Problemin çözümünü veren algoritmayı yaz
 - Algoritmayı bir programlama dili ile yaz
 - Programın doğruluğunu test et.

örnek bir problem çözme işlemi

■ Örnek2: iki sayının toplamının programı

1. Problem matematiksel bir toplama işlemidir.

2. Çözüm için iki sayıya ve bu sayıları saklayacağımız değişkenlere ihtiyacımız var.

3. Sayıların belirlenmesi ve toplama işleminin yapılması.

4. Algoritma

t, s1, s2 değişkenlerini tanımla

s1=?

s2=?

t=s1+s2

t

5. Program

Dim s1, s2, t

input s1, s2

t=s1+s2

Print t

End.

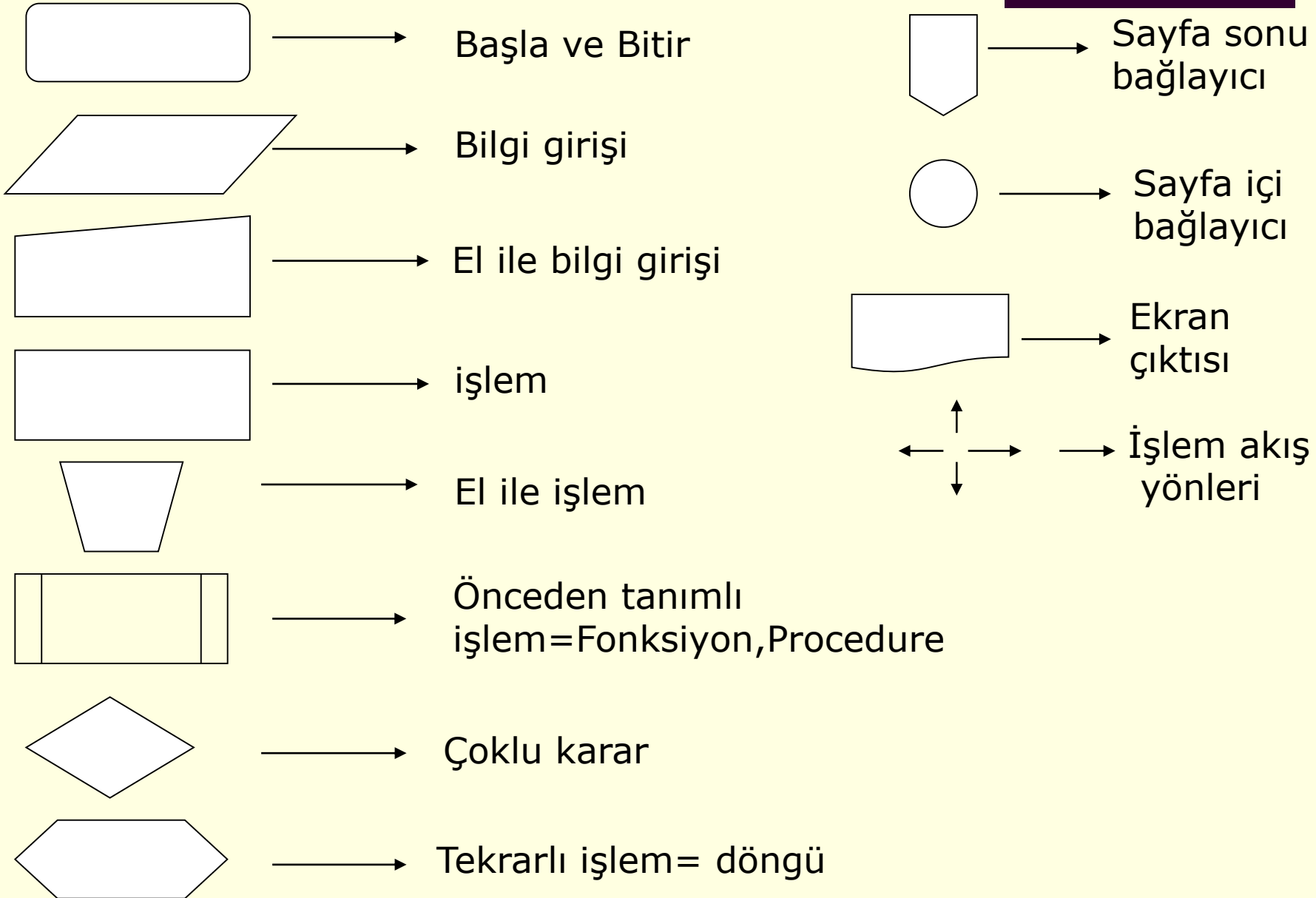
6. Test. s1=2, s2=3, t=s1+s2=2+3=5

- **Örnek3:** kullanıcının girdiği iki sayının karelerinin toplamını görüntüleyen algoritmayı yazınız.
 - Başla,
 - A1: sayac=1, i=1
 - A2: sayı(i) değerini giriniz
 - A3: $t(i)=\text{sayı}(i)*\text{sayı}(i)$
 - A4: sayac=sayac+1
 - A5: sayac<3 ise i=2 olsun ve A2 ye git
 - $T= t1+t2$
 - T yi ekrana yaz
 - Dur.

Akış diyagramları

- Algoritmaların özel şekillerle çizilmesine akış diyagramı adı verilir.
- Akış diyagramı çizilirken şu kurallara dikkat edilmesi gerekir.
 - Algoritmanın adımları belirlenir.
 - Algoritmanın adımlarına uygun akış diyagram şekilleri belirlenir.
 - Ayrıntıya inmeden diyagram oluşturulur.

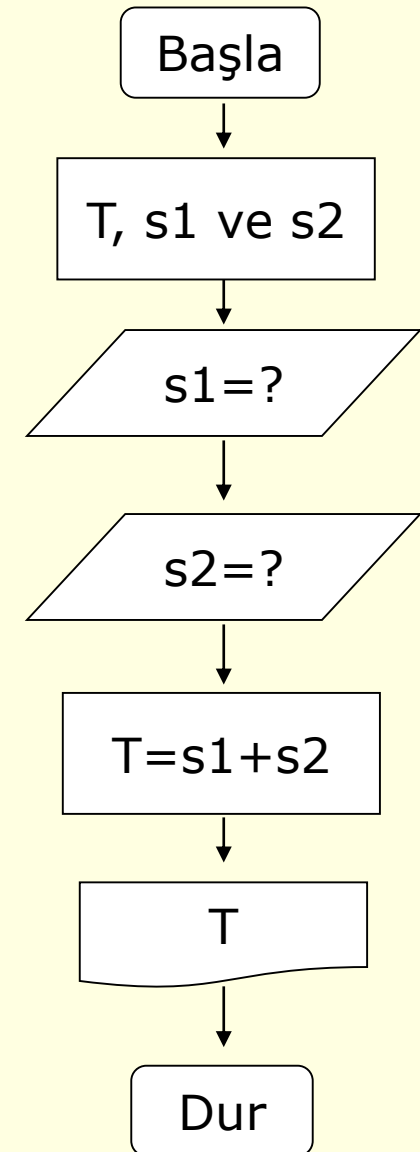
Akış diyagramları



Akış diyagramları

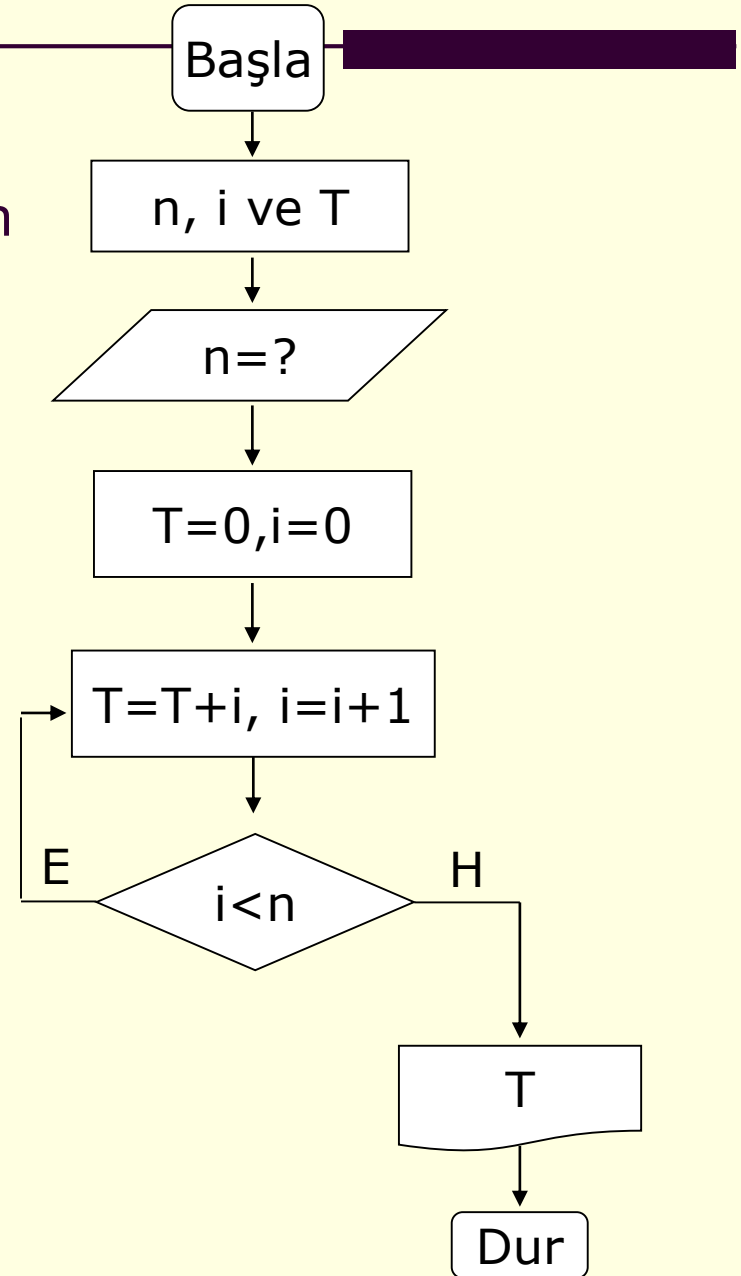
- **Örnek4:** iki sayının toplamını bulan algoritmayı ve akış şemasını hazırlayınız.

- Adım0:Başla
- Adım1: Toplam, s1,s2 yi tanımla
- Adım2: İlk sayıyı öğren
- Adım3: İkinci sayıyı öğren
- Adım4: Sayıları topla
- Adım5: toplamı görüntüle
- Adım6:Dur
 - Toplam, s1, s2
 - S1=?
 - S2=?
 - T=S1+S2
 - T



Örnek5: Birden n ' e kadar olan sayıların toplamlarını bulan algoritmayı ve akış şemasını hazırlayınız

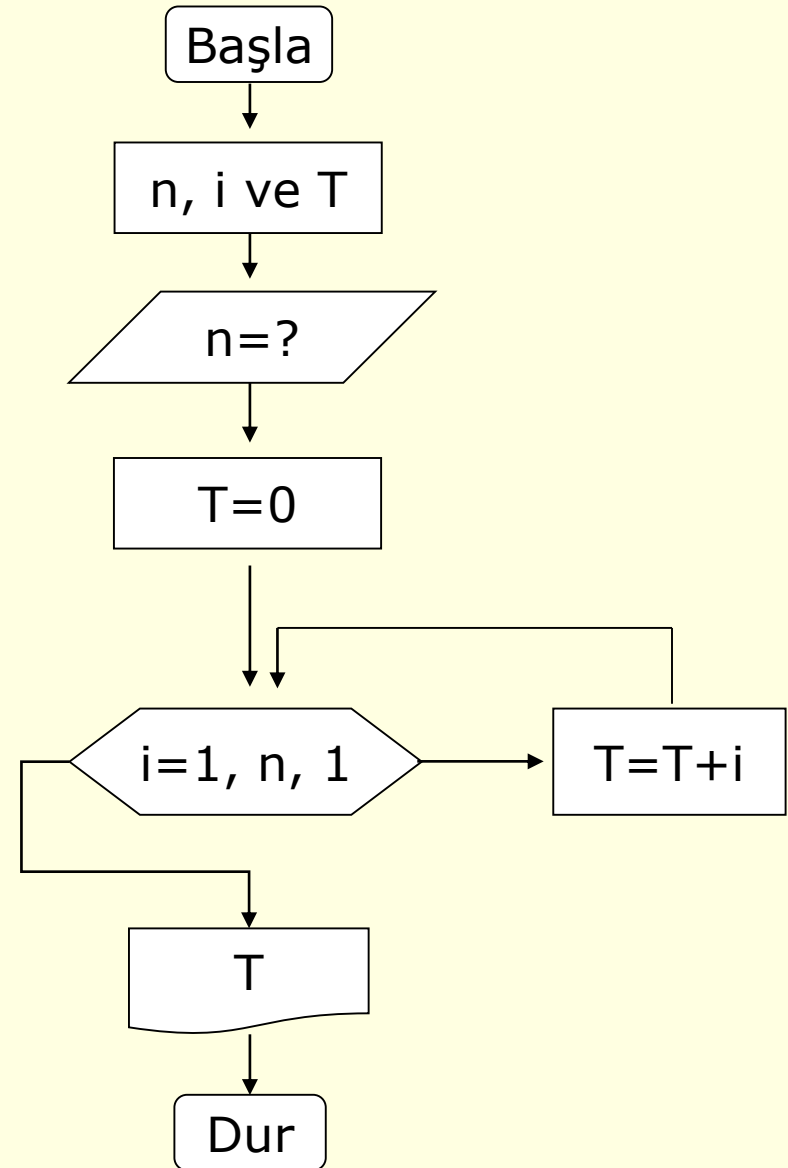
1. Başla
2. n , i ve T değişkenlerini tanımla
3. n sayısını gir, $n=?$
4. T ve i yi sıfırla
5. $T=T+i$, $i=i+1$
6. Eğer $i < n$ ise 5.adıma git
7. T ' yi ekrana yaz
8. Dur



Örnek5 in döngü yöntemiyle yapılması

Örnek6:

- Başla
- n , i ve T yi tanımla
- n sayısını gir, $n=?$
- T ' yi sıfırla
- $i=1$ den n ' ye kadar 1 er 1 er ilerle
 - $T=T+i$
- T ' yi ekrana yaz
- Dur



Programlamaya Giriş

- Bilgisayarlar problemleri çözebilmek için programlara ihtiyaç duyarlar.
- Program belirli bir işlemi gerçekleştirmek için bilgisayara iletilen komutlar dizisidir.

1-Programlama dilleri

- Programlama dili, kişi ile bilgisayar arasındaki iletişimi sağlayan bir araçtır.

Yaz komutu				
Basic	Pascal	C/C++	Matlab	
Print	Writeln	Printf	fprintf	disp

Bilgi al komutu			
Basic	Pascal	C/C++	Matlab
Input	Readln	Scanf	Input

İşlemler

- Bilgisayar programları ile geliştirilen işlemler üç gruba ayrılır:
 1. **Matematiksel (aritmetik) işlemler**
 2. **Karşılaştırma (karar) işlemler**
 3. **Mantıksal (lojik) işlemler.**

1- Matematiksel işlemler

İşlem	Matematik	Bilgisayar
Toplama	$a+b$	$a+b$
Çıkarma	$a-b$	$a-b$
Üs alma	a^b	a^b
Çarpma	$a.b$	$a*b$
Bölme	$a\div b$	a/b

İşlemlerde öncelik sıraları

Sıra	İşlem	Bilgisayar dili
1	Parantez	((.....))
2	Üs alma	a^b
3	Çarpma ve bölme	$a*b$ ve a/b
4	Toplama ve çıkarma	$a+b$ ve $a-b$

Örnek7: $a=9$ ve $b=16$ için aşağıdaki denklemleri çözünüz.

denklem	İşlem	sonuç
$a+b^{1/2}$	$a+b^{1/2}$	$9+16^{1/2}=13$
$a+b^{(1/2)}$	$a+\sqrt{b}$	
$(a+b)^{1/2}$	$(a+b)^{1/2}$	
$(a+b)^{(1/2)}$		

İşlemler- karşılaştırma işlemleri

2-Karşılaştırma işlemleri: iki değer karşılaştırılması durumunda yapılan işlemlerdir. İşlem sembolleri yardımıyla ifadeler karşılaştırılır.

Karşılaştırma sonunda program farklı şekillerde yönlendirilir.

İşlem sembolü	Anlamı
=, (==)	Eşittir
~=	Eşit değildir
>	Büyüktür
<	Küçüktür
>=	Büyük eşittir
<=	Küçük eşittir

■ **Örnek8:** klavyeden girilen iki sayının hangisinin büyük olduğunu bulan algoritma ve akış diyagramını hazırlayınız.

1. Başla
2. a, b değişkenlerini tanımla
3. a sayısını gir
4. b sayısını gir
5. Eğer $a > b$ ise
 “a, b den büyüktür” yaz
6. Eğer $a < b$ ise
 “b, a dan büyüktür” yaz
7. Eğer $a = b$ ise
 “a ile b eşittir.” yaz
8. Dur.

İşlemler- Mantıksal işlemler

■ 3- Mantıksal işlemler

- Aritmetik ve karşılaştırma işlemlerini sonuçlandıran bilgisayar temel mantık işlemleri de yapabilir. Programlardaki karşılaştırma ifadelerinde birden fazla koşulun belirli bir özelliğe sağlanması istenebilir. Bu durumda araya mantıksal işlem operatörleri konulur. Bu operatörler

Mantıksal işlem	Komut	Matematiksel sembol
Ve	And	&
Veya	Or	
Değil	Not	~

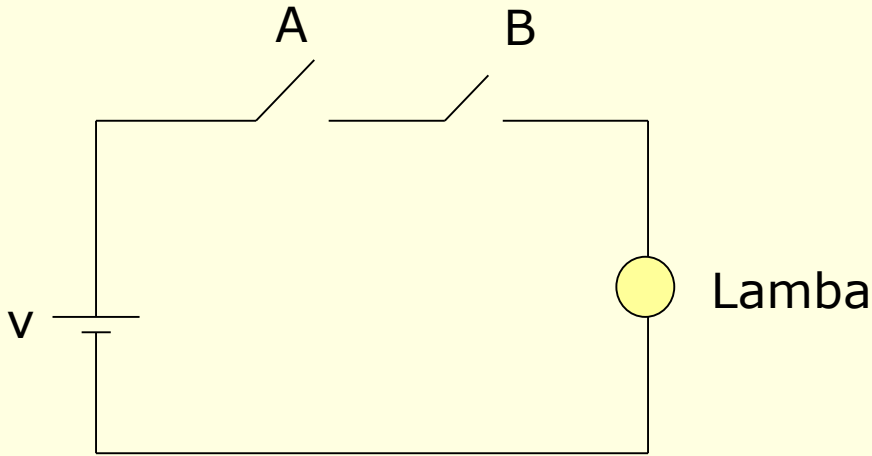
İşlemler- Mantıksal işlemler

- Mantıksal işlem sonuçları farklı ortamlara göre genel anlamları ile tablodaki gibi verilebilir.

Sonuç	Rakam	Bilgisayar	Lamba	Anahtar	Gerilim
Doğru	1	True	Yanıyor	Kapalı	Var
Yanlış	0	False	Yanmıyor	Açık	yok

İşlemler- Mantıksal işlemler

■ Ve (and) doğruluk tablosu

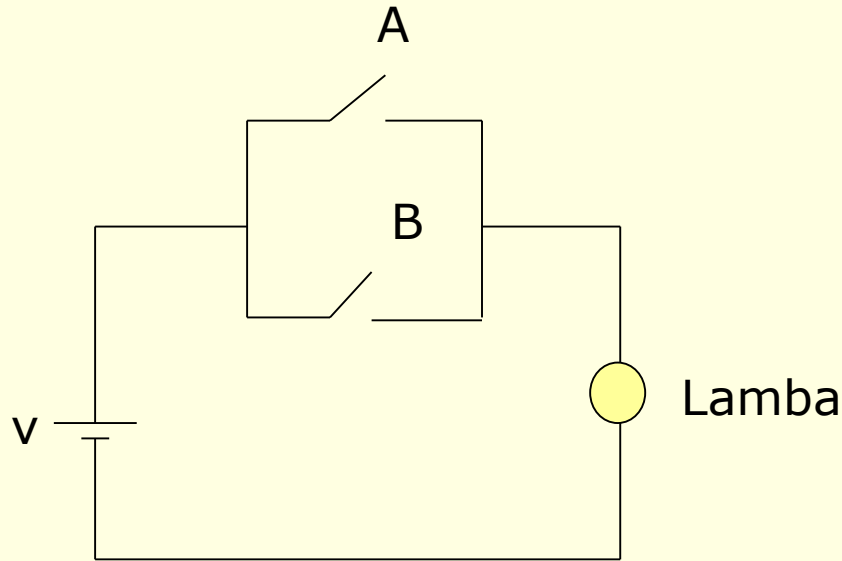


A	B	A ve B
0	0	0
0	1	0
1	0	0
1	1	1

VE (and) işleminde bütün koşullar doğru ise sonuç da doğrudur.

İşlemler- Mantıksal işlemler

■ Veya (or) doğruluk tablosu

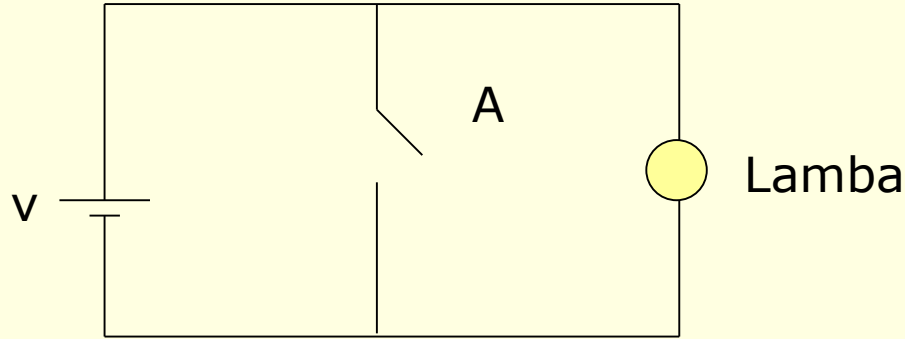


A	B	A veya B
0	0	0
0	1	1
1	0	1
1	1	1

VEYA (or) işleminde ise sonucun doğru olması için koşullardan birinin doğru olması yeterlidir.

İşlemler- Mantıksal işlemler

■ Değil (not) doğruluk tablosu



A	A DEĞİL (A')
0	1
1	0

DEĞİL (not) işleminde ise sonuç daima koşulun tersidir.

İşlemler- Mantıksal işlemler

- Mantıksal işlemlerde işlem sırası

Sıra	İşlem	Bilgisayar dili
1	Parantez	(.....)
2	değil	not
3	ve	and
4	veya	or

İşlemler- Mantıksal işlemler

- **Örnek9:** $a=76$, $b=63$ için $c=a.b$, $d=a+b$, $e=a'$ mantıksal işlem sonuçlarını hesaplayınız.

$$A=76 = 01001100$$

$$B=63 = 00111111$$

$$A.B = 00001100$$

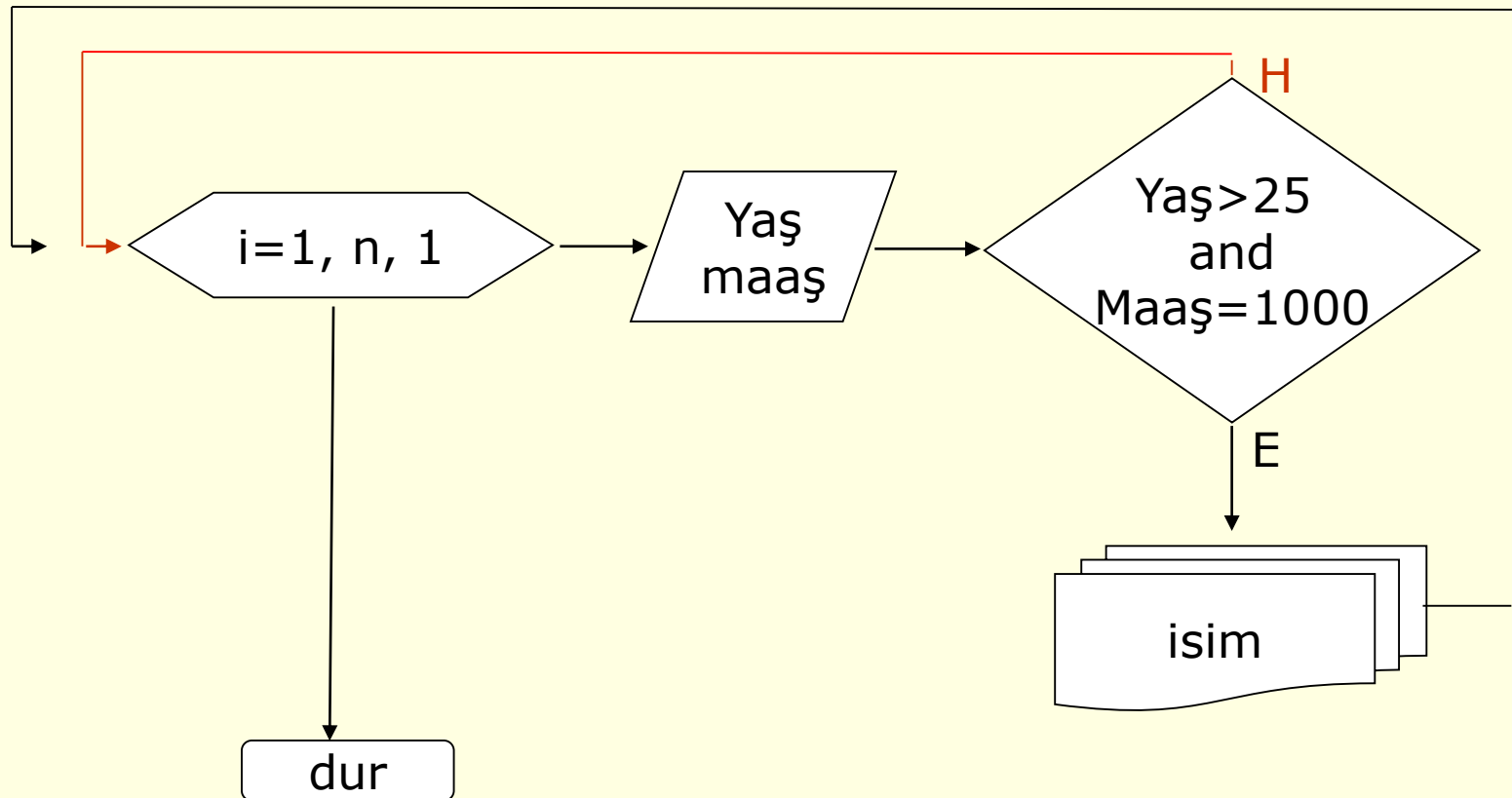
$$A+B = 0\dots\dots\dots$$

$$E = ?$$

İşlemler- Mantıksal işlemler

- **Örnek10:** Bir iş yerindeki işçilerden yaşı 25 den büyük ve maaşı 1000 TL ye eşit olanların listesi nasıl alınabilir.

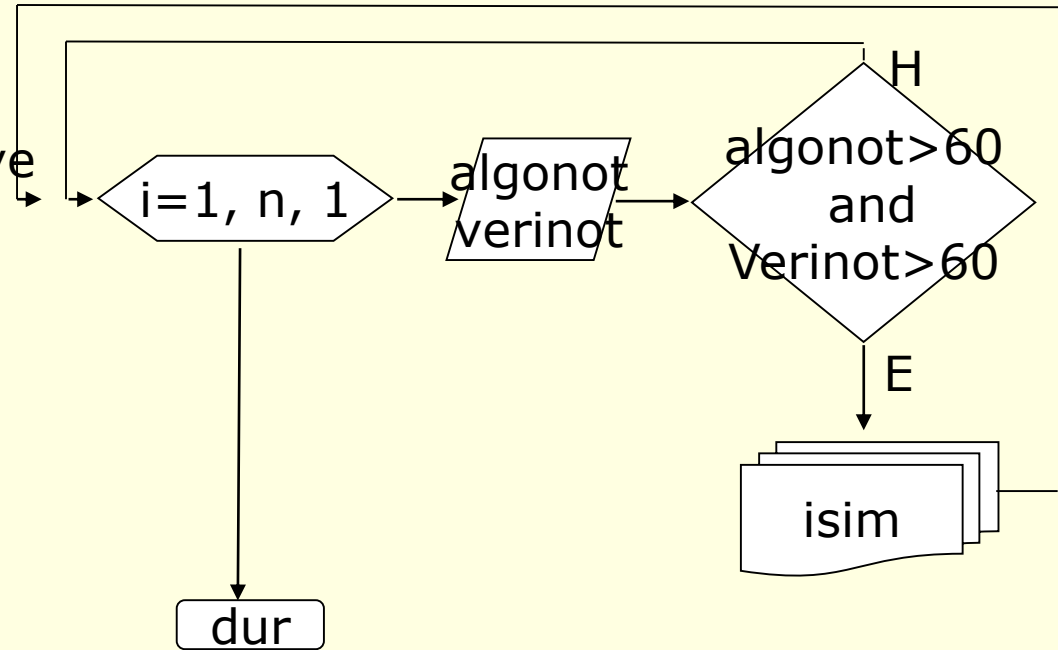
Eğer $yaş > 25$ ve $maaş = 1000$ ise
isimleri ekrana yaz



İşlemler- Mantıksal işlemler

- **Örnek11:** Bilgisayar bölümü öğrencilerinin algoritmadan ve veri tabanı derslerinden 60'ın üzerinde not alanların listesini bulan algoritma ve akış şemasını hazırlayınız.

1. Başla
2. algonot, verinot ve n yi tanımla
3. Bölümdeki öğrenci sayısını giriniz, $n=?$
4. 1 den n kadar her öğrenciye sor ($i>n$ ise 8.adıma git)
5. i inci öğrencinin notlarını giriniz
6. Eğer $algonot>60$ and $verinot>60$ ise öğrencinin adını yaz ve 4 adıma git
7. Değilse 4. adıma git
8. Dur



- **Ödev1:** Bilgisayar bölümünde algoritma ve veritabanından 60 üzeri not alanların sayısını bulan algoritma ve akış şemasını hazırlayınız
- **Ödev2:** Bilgisayar öğrencilerinin algoritma dersinden 70 üzerinde not alıp veri tabanından veya internet derslerinin herhangi birinden 65'in üzerinde not alanların isimlerini bulan algoritma ve akış şemasını hazırlayınız
 - Eğer $algonot > 70$ ve $(verinot > 65$ veya $intnot > 65)$ ise
- **Ödev3:** $a=33$ ve $b=77$ için şu işlemlerin sonuçlarını bulunuz.
 - $c = a \text{ and } b$
 - $d = a \text{ or } b$
 - $e = \text{not}(b)$

Programlamanın yardımcı unsurları

1. Değişkenler/ Bilgi tipleri

- Algoritma ve programlama sırasında bilgisayarda tutulacak her bilgi çeşitli yapılarda olmaktadır. Bu yapı bilgisayarın o bilgiyi anlamasına, işlem yapabilmesine ve saklamasına yardımcı olur.
 1. **Sayısal bilgi tipleri;** İşlemlerde bazı büyüklere temsil etmek amacıyla sayısal bilgi tipleri bulunmaktadır. Sayısal bilgi tipleri **Tamsayı ve Ondalıklı** sayı tipleri olarak iki türdür.

a. Tamsayı bilgi tipleri: sadece tam kısmı bulunan sayısal bilgiler için kullanılan bu bilgi tipleri, bilginin büyüklüğüne göre farklı şekillerde isimlendirilmektedir.

Bilgi tipi	Alabileceği en küçük değer	Alabileceği en büyük değer	Bellekte kapladığı alan (byte)
Byte	0	255	1
Shortint	-128	127	1
Integer	-32.768	32.768	2
Word	0	65.535	2
Longint	-2.147.483.648	2.147.483.647	4

b. Ondalıklı sayı tipleri: hem tam hem de ondalıklı kısmı bulunan sayısal bilgiler için kullanılan bu bilgi tipleri, yine sayısal bilginin büyüklüğüne göre farklı şekillerde isimlendirilirler.

Bilgi tipi	Alt sınır	Üst sınır	Bellekte kapladığı alan (byte)	Duyarlılık (hane)
Real	$\pm 2,9 \cdot 10^{-39}$	$\pm 1,7 \cdot 10^{38}$	6	11-12
Single	$\pm 1,5 \cdot 10^{-45}$	$\pm 3,4 \cdot 10^{38}$	4	7-8
Double	$\pm 5 \cdot 10^{-324}$	$\pm 1,7 \cdot 10^{308}$	8	15-16
Extended	$\pm 3,4 \cdot 10^{-4932}$	$\pm 1,1 \cdot 10^{4932}$	10	19-20
Comp	$\pm -2^{63} + 1$	$\pm 2^{63} - 1$	8	19-20

2- Alfasayısal bilgi tipleri: Alfasayısal yani metinsel bilgi tipleri de 3 farklı isimde kullanılır.

Bilgi tipi	Alabildiği değerler (karakter)	Bellekte kapladığı alan (byte)
Char	1	1
String	1-255 string[n]	n+1
Boolean	True veya False	1

Örnek12: Klavyeden girilen bir sayının Negatif, Pozitif veya Sıfır olduğunu bulan algoritmayı yazınız ve akış diyagramını çiziniz.

Bir sayının pozitif, negatif veya sıfır olduğunu bulmak için sıfır(0) ile test edilmesi gerekir.

Algoritma:

A1: Başla

A2: Sayıyı (X) giriniz

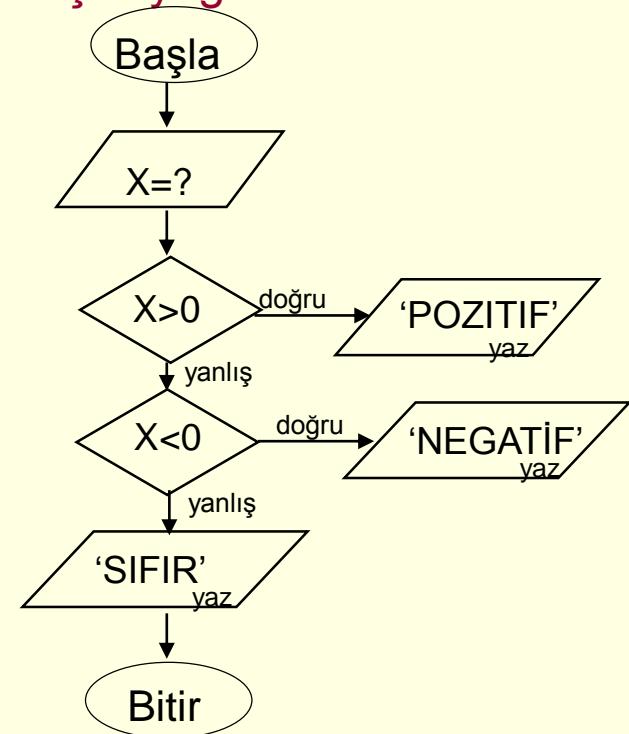
A3: Eğer $X > 0$ ise 'POZITIF' yaz

A4: Eğer $X < 0$ ise 'NEGATİF' yaz

A5: Eğer $X = 0$ ise 'SIFIR' yaz.

A6: Bitir

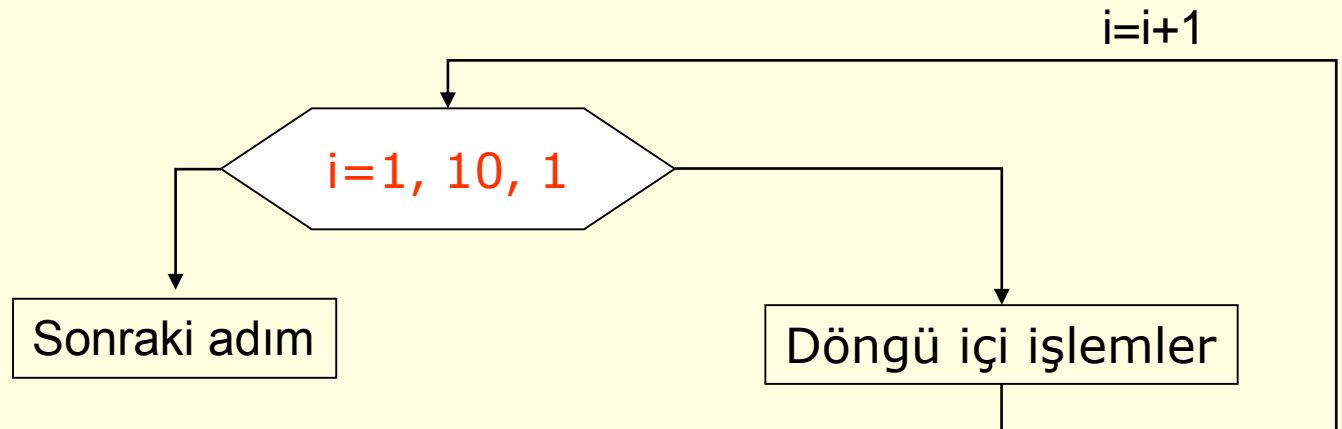
Akış Diyagramı:



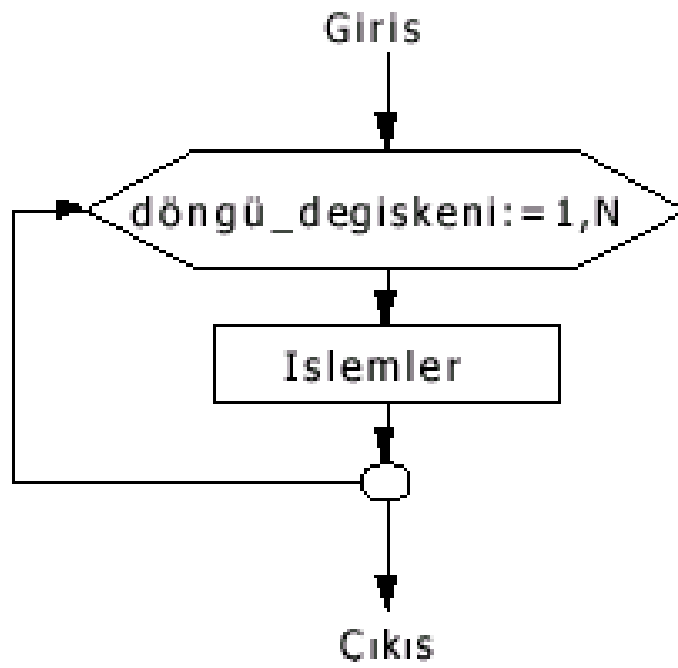
2. Döngüler

Program içerisinde bazı işlemler tekrarlanmak zorundadır bu duruma döngü adı verilir. Her döngünün kontrol değişkeni, başlangıç değeri, bitiş değeri ve artım değeridir.

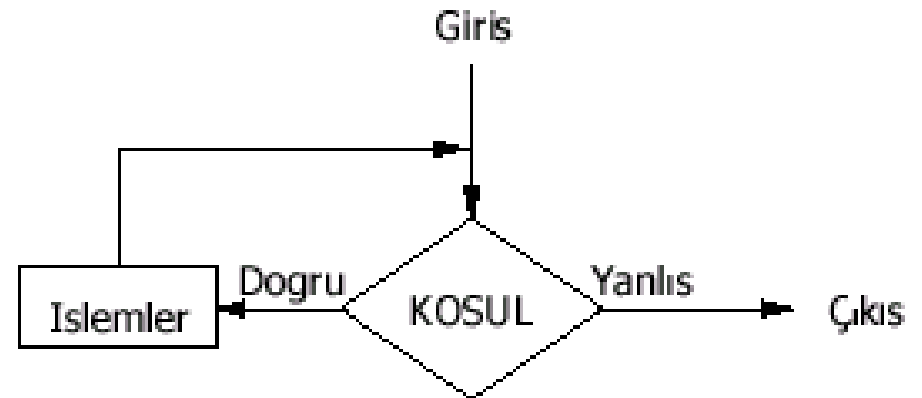
Kontrol değişkeni=başlangıç değeri, bitiş değeri, artım değeri.



Tekrarlı İşlemler (Döngü işlemleri)

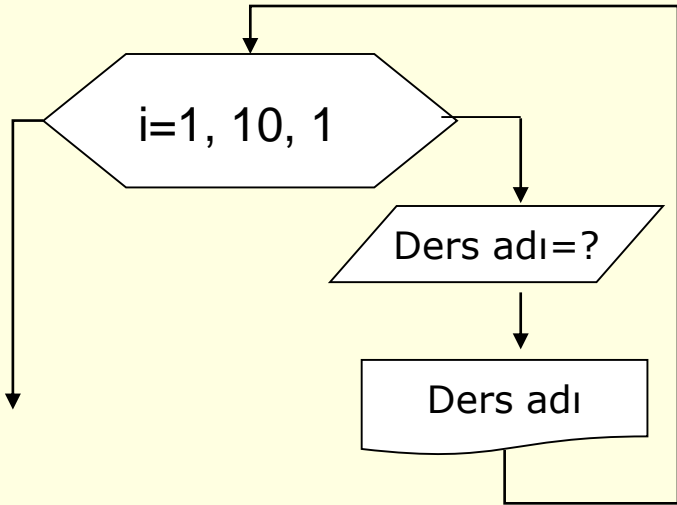
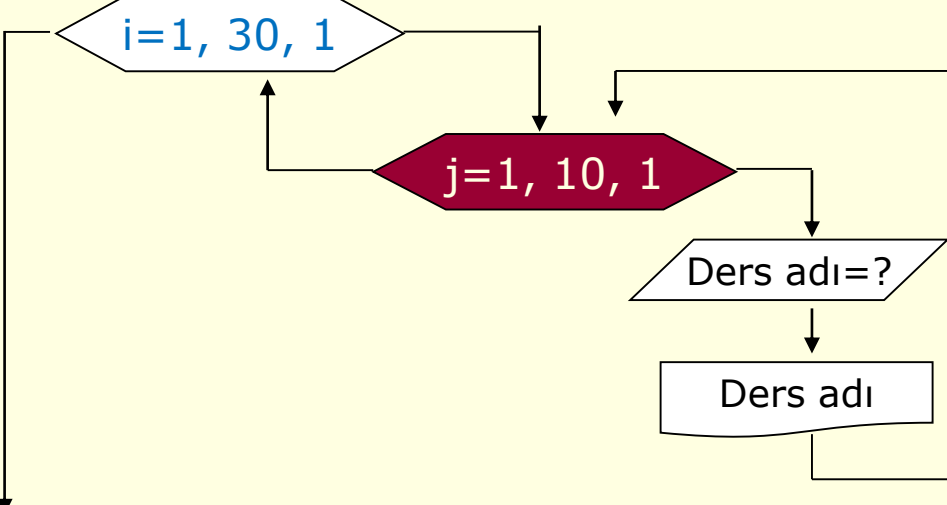


a



b

- **İç içe döngüler:** bir değişkenin birden fazla alt işlemi varsa bu iç döngü içinde verilir.

Tek döngü	İç içe döngü
Bir öğrencinin bir dönemdeki dersleri	30 öğrencinin bir dönemdeki dersleri
 <pre>graph TD; A{i=1, 10, 1} --> B[/Ders adı=?/]; B --> C[Ders adı]; C --> A;</pre>	 <pre>graph TD; A{i=1, 30, 1} --> B{j=1, 10, 1}; B --> C[/Ders adı=?/]; C --> D[Ders adı]; D --> B; B --> A; A --> A;</pre>

Soru

Girilen N deęerinin FAKTÖRİYELİNİ hesaplayan algoritmanın akış diyagramını oluřturunuz.

Adım 1-Başla

Adım 2- n, i ve F yi tanımla

Adım 3- n yi giriniz

Adım 4- $F=1$ ve $i=1$

Adım 5- $F=F*i$

Adım 6- $i=i+1$

Adım 7- Eğer $i \leq n$ ise

5. adıma git

Değilse

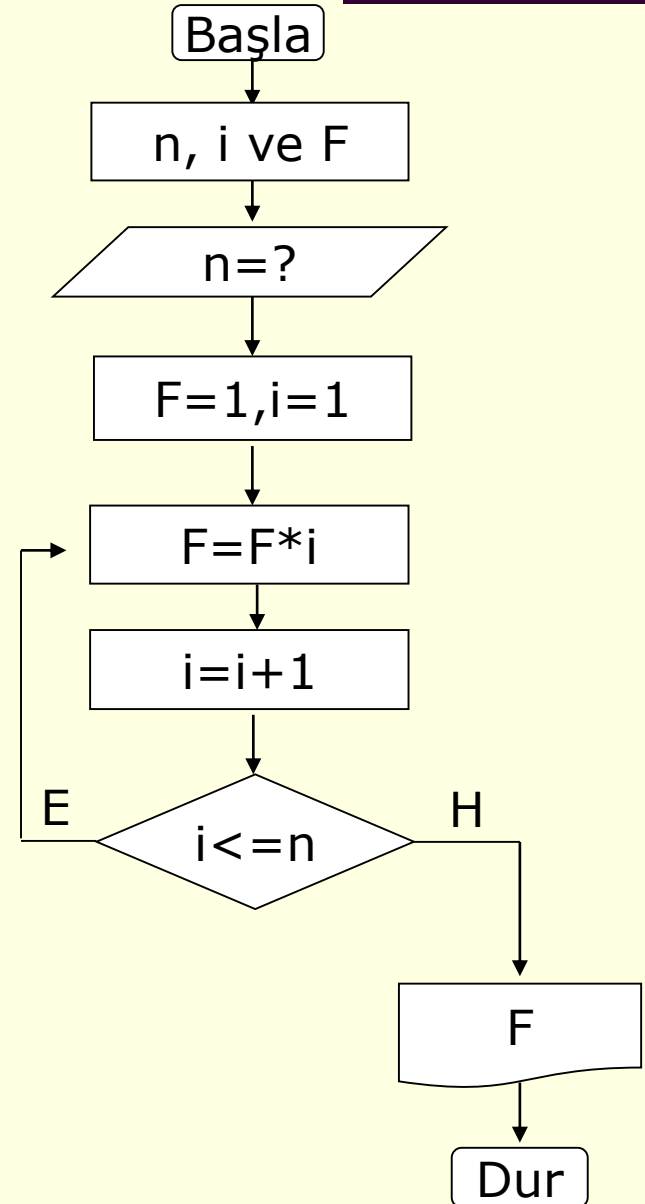
8. adıma git

Adım 8- F yi ekrana yaz

Adım 9-Dur

n=5 olarak girilirse

i	Eski F	Yeni F
1	1	$1*1=1$
2	1	$1*2=2$
3	2	$2*3=6$
4	6	$6*4=24$
5	24	$24*5=120$



Programlamanın yardımcı unsurları

3. Koşullu İfadeler:

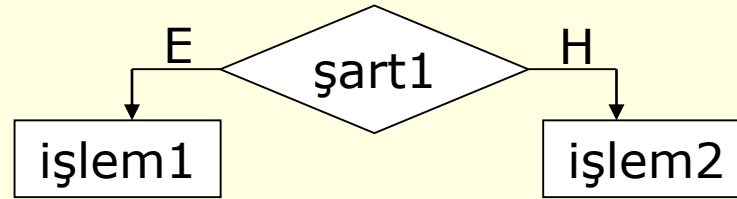
bir işlem birden fazla farklı durum sonucunda çeşitli değerler üretebilir. Bu durumda karar işlemleri kullanılır.

1- Eğer (şart1) ise

işlem1

Değilse

işlem2

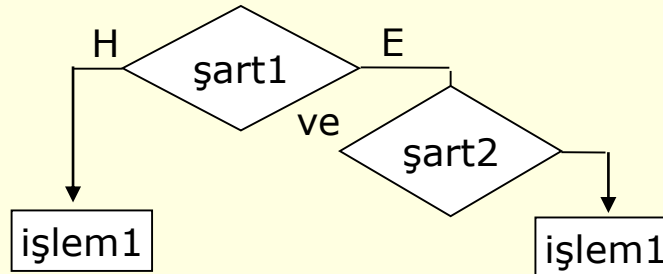


2- Eğer (şart1) **ve/veya** (şart2) ise

işlem1

Değilse

işlem2



Eğer yapısının Üç şekli vardır:

Eğer Şart
1. işlem
2. işlem
3. işlem
:
Eğer Sonu

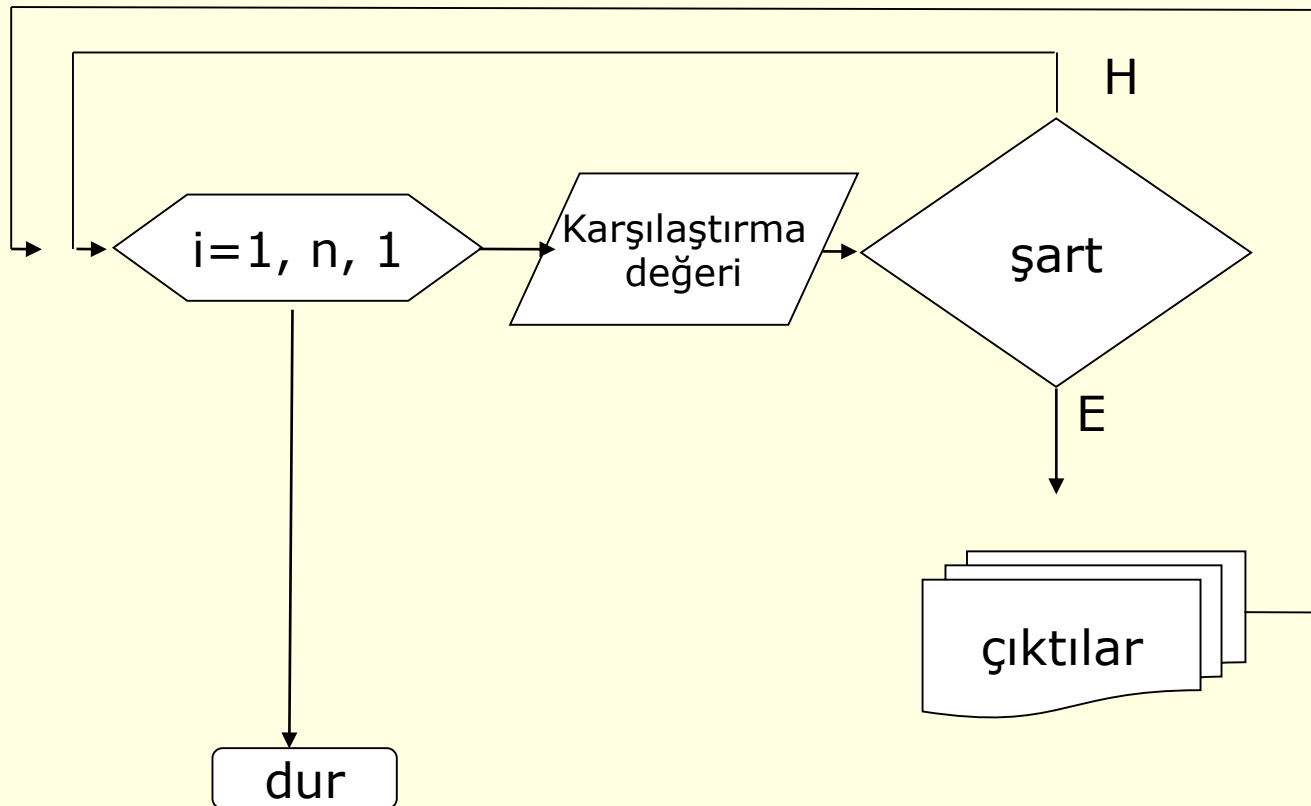
Eğer Şart
1.işlem;
Değilse
2. işlem
Eğer Sonu

Eğer Şart
1.işlem
Değilse eğer ...
2.işlem
Değilse
3.işlem
Eğer Sonu

Programlamanın yardımcı unsurları

4. Döngü ve Koşullu işlemleri

- Bazı durumlarda döngü içerisinde karar sonucunda işlem yaptırılmak istenebilir.



Programlamanın yardımcı unsurları

5.Sayaç: Programlarda; bazı işlemlerin belirli sayılarda yaptırılması veya üretilen değerlerin sayılması gerekebilir. Örneğin klavyeden girilen bir cümlede kaç sesli harf olduğunu bulan programda, cümlenin her harfi sırayla çağırılır ve sesli harfler kümesine ait olup olmadığı araştırılır. Eğer sıradaki çağırılan harf bu kümeye ait ise bunları sayacak olan değişkenin değeri bir arttırılır.

$$\text{Say} = \text{Say} + 1$$

şeklindeki işlemde sağdaki ifadede değişkenin eski(önceki) değerine '1' eklenmekte; bulunan sonuç yine kendisine, yeni değer olarak aktarılmaktadır. Bu tür sayma işlemlerine, algoritmada “**sayaç**” veya “**sayıcı**” adı verilir. Yani “sayaç”; işlem akışı kendisine her geldiğinde, belirtilen adım değeri kadar artan/azalan değişkendir.

Örnek13: Aşağıdaki algoritmada 1-5 arası sayılar, sayaç kullanılarak ekrana yazdırılmaktadır.

A1:Başla

A2: S yi tanımla

A3. S=0

A4. S=S+1

A5: Eğer S < 5 ise

S' yi ekrana yaz

4. adıma git

Değilse

6. Adıma git

A6:Bitir

Burada, S: sayaç değişkenidir

Programlamanın yardımcı unsurları

6. Dizi

Bir değişken içerisinde birden çok verinin numaralandırılarak saklanmasına DİZİ denir. Dizi içinde aynı isim altında çok sayıda veri olduğu için bunları birbirinden ayırt etmek için İNDİS adı verilen bir bilgiye ihtiyaç vardır.

Örnek14: A değişkeni A(5) ise dizi değişkenidir ve aşağıdaki gibi simgelenir

A Dizisi	(23,	45,	-1,	1.24,	78)
	A(1)	A(2)	A(3)	A(4)	A(5)

Programlamanın yardımcı unsurları

7. İki Boyutlu Diziler (Matrisler)

$$F = \begin{bmatrix} F_{11} & F_{12} & \dots & F_{1n} \\ F_{21} & F_{22} & \dots & F_{2n} \\ \dots & \cdot & \cdot & \dots \\ \dots & \cdot & \cdot & \dots \\ F_{m1} & F_{m2} & \cdot & F_{mn} \end{bmatrix}_{m \times n}$$

Böyle dikdörtgen bir tabloya, $m \times n$ matrisidir denir. Burada, m sayısına matrisin satır sayısı, n sayısına matrisin sütun sayısı denir. Eğer, $m=n$ ise matris kare matris olarak adlandırılır.