Algoritmalar ve Programlama

Algoritma

Algoritma

- ***Bir sorunu / problemi çözmek veya belirli bir amaca ulaşmak için gerekli olan sıralı mantıksal adımların tümüne algoritma*** *denir.*
- **Algoritma bir sorunun çözümü için izlenecek yolun tanımıdır. Kısaca algoritma mevcut bilgilerden istenilenlere erişme yöntemidir.**
- Programlamanın en önemli kısmı problemin çözümü için **algoritma** hazırlayabilmek / geliştirmektir.
- Hazırlanan algoritmanın bir programlama dili ile kodlanması işin basit kısmıdır.
- Kullanılan dilin basic / pascal / c ya da başka bir programlama dili olması bir şeyi değiştirmez.
- Yani burada Bu yüzden kullanılan programlama dilinin eski yada yeni bir programlama dili olması hiç önemli değildir.

Algoritmadan Beklentiler

- **Etkinlik:** Bilgisayarlar düşünemez. Bu yüzden algoritmanın her adımı anlaşılır, basit ve kesin bir biçimde ifade edilmiş olmalıdır. Yorum gerektirmemeli ve belirsiz ifadelere sahip olmamalıdır. Gereksiz tekrarlarda bulunmayan diğer algoritmalar içerisinde de kullanılabilir olmalıdır.
- **Sonluluk:** Her algoritmanın bir başlangıç noktası, belirli işlem adımı ve bir bitiş noktası içermelidir. Sonsuz döngüye girmemelidir.
- **Kesinlik:** İşlem sonucu kesin olmalı, aynı veri için her yeni çalıştırmada aynı sonucu üretmelidir.
- **Giriş/Çıkış:** Algoritma giriş (üzerinde işlem yapılacak değerler) ve çıkış (yapılan işlemler neticesinde üretilen sonuç değerler) değerlerine sahip olmalıdır.
- **Başarım/Performans:** Amaç donanım gereksinimi (bellek kullanımı gibi), çalışma süresi gibi performans kriterlerini dikkate alarak yüksek başarımlı programlar yazmak olmalıdır.

Algoritma İfade Şekilleri

1. Algoritmanın metin olarak yazılması
   - Çözülecek problem, adım adım metin olarak yazılır.
   - Her satıra numara verilir.
   - ‘Başla’ ile başlayıp ‘son’ ile bitirilir

Algoritma İfade Şekilleri

- **Problem:** Klavyeden girilen sayının karesini hesaplayarak ekrana yazdıran programın algoritmasını yazınız?
    1) Başla
    2) Sayıyı (A) gir
    3) Sayının karesini hesapla (Kare = A*A işlemini yap)
    4) Sonucu (Kare) yaz
    5) Dur

Algoritma İfade Şekilleri

- Pseudo Code (Kaba Kod)
  - Sözde programlar, doğrudan **konuşma dilinde** ve programlama mantığı altında, **eğer, iken** gibi koşul kelimeleri ve **> = <** gibi ifadeler ile beraber yazılır. İyi bir biçimde yazılmış, sözde koddan, programlama diline kolaylıkla geçilebilir.

Algoritma İfade Şekilleri

- Problem: Verilen bir sıcaklık derecesine göre suyun durumunu belirten bir sözde program(pseudo kod) yazınız.
- İstenilen programın Pseudo Kodu:
    1. Program açıklama mesajı yaz.
    2. Kullanıcın sıcaklığı girmesi için bir uyarı mesajı yaz.
    3. Girilen Sıcaklığı Oku.
    5. Eğer Sıcaklık < 0 ise Durum=“Buz”
    6. Eğer Sıcaklık>= 100 ise Durum=“Buhar”
    7. Değilse Durum =“Su”
    8. Sonucu Yaz.

Akış Diyagramları (Şemaları)

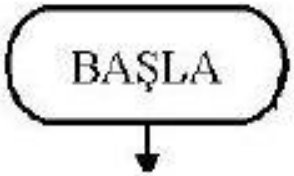
- Algoritmanın, görsel olarak simge ya da sembollerle ifade edilmiş şekline “akış şemaları” veya FLOWCHART adı verilir. Akış şemalarının algoritmadan farkı, adımların simgeler şeklinde kutular içine yazılmış olması ve adımlar arasındaki ilişkilerin ve yönünün oklar ile gösterilmesidir.

Akış Diyagramları (Şemaları)

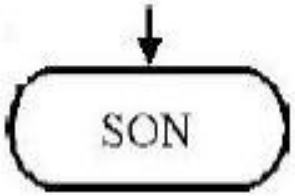
- En basit şekliyle dikdörtgen kutulardan ve oklardan oluşur. Akış şeması sembolleri ANSI (American National Standards Institute) standardı olarak belirlenmiş ve tüm dünyada kullanılmaktadır.

Akış Diyagramları (Şemaları)

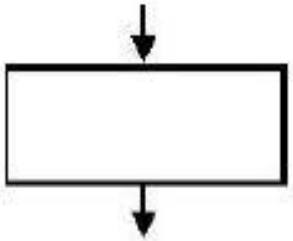
Her simge, yapılacak bir işi veya komutu gösterir. Akış şemalarının hazırlanmasında aşağıda yer alan simgeler kullanılır.



Bir algoritmanın başladığı konumu göstermektedir. Tek çıkışlı bir şekildir.

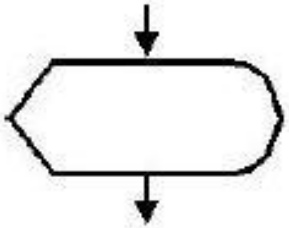


Bir algoritmanın bittiği konumu göstermektedir. Tek girişli bir şekildir.

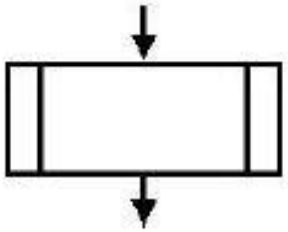


Bir algoritmada aritmetik işlem yapılmasını sağlayan şekildir. Bu dörtgen kutu içerisine yapılmak istenen işlem yazılır. Tek girişli ve tek çıkışlı bir şekildir.

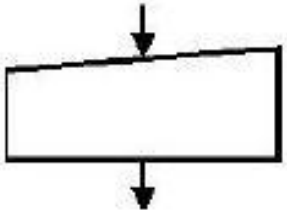
Akış Diyagramları (Şemaları)



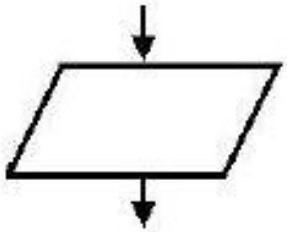
Algoritmada bir bilginin ekrana yazılacağı konumu gösteren şekildir. Ekrana yazılacak ifade ya da değişken bu şekil içerisine yazılır.



Bir algoritmada başka bir yerde tanımlanmış bloğun yerleştiği konumu gösteren şekildir. Kutu içerisine bloğun adı yazılabilir.

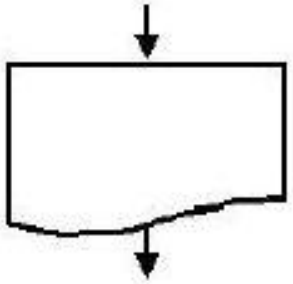


Klavyeden Bilgisayara bilgi girilecek konumu belirten şekildir. Girilecek bilginin hangi değişkene okunacağını kutu içerisine yazabilirsiniz.

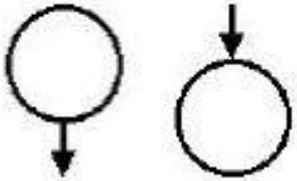


Giriş Çıkış komutunun kullanılacağı yeri belirler ve kutu içerisine hangi değişkeni ve OKUma mı YAZ mı yapılacağını belirtmeniz gerekir.

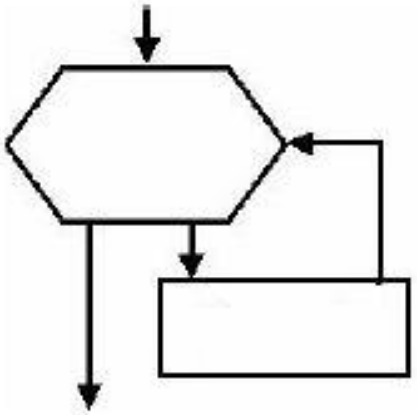
Akış Diyagramları (Şemaları)



Bilginin Yazıcıya yazılacağı konumu gösteren şekildir.

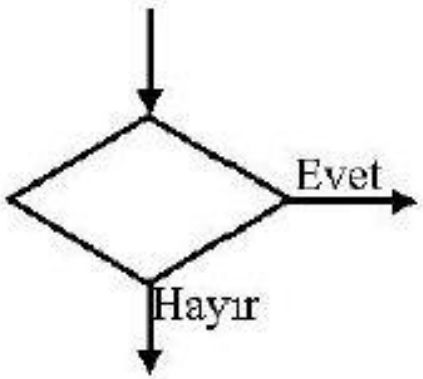


Bir algoritmanın birden fazla alana yayılması durumunda bağlantı noktalarını gösteren şekildir. Tek girişli veya tek çıkışlı olarak kullanılırlar.



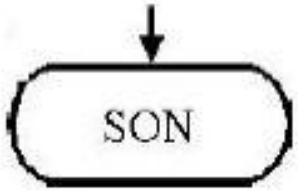
Birçok programda; belirli işlem blokları ardışık değerlerle veya bazı koşullar sağlanıncaya kadar tekrarlanır. Bu tekrarlamalı işlemler döngü olarak isimlendirilir. Döngü şeklinin içine; döngü (çevrim, kontrol) değişkeni, başlangıç değeri, bitiş değeri ve adımı yazılır.

Akış Diyagramları (Şemaları)



Bir algoritmada bir kararın verilmesini ve bu karara göre iki seçenekten birinin uygulanmasını sağlayan şekildir. burada eşkenar dörtgen içerisine kontrol edilecek mantıksal koşul yazılır. Program akışı sırasında koşulun doğru olması durumunda "Evet" yazılan kısma Yanlış olması durumunda "Hayır" yazılan kısma sapılır. Tek girişli ve çift çıkışlı bir şekildir.

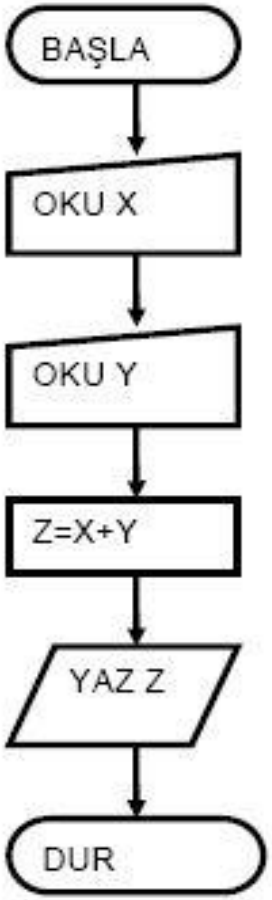
Akış Çubuğu



Programın bittiği yer ya da yerleri gösteren bir şekildir.

Akış Diyagramları (Şemaları)

Klavyeden girilen İki sayının toplamını hesaplayıp yazan pseudo kod ve akış şemasını hazırlayınız.



(X: Birinci sayı, Y: İkinci sayı, Z: toplam)
A1 : Başla
A2 : Klavyeden oku X
A3 : Klavyeden oku Y
A4 : Hesapla Z = X + Y
A5 : Yaz Z
A6 : Dur

Algoritmalarda Kullanılan Operatörler

- İşlemleri belirten sembollere bilgisayar dilinde "operatör" denir.
- Algoritmada kullanılan operatörler Tabloda verilmiştir.

Matematiksel İşlem oper	
Üs alma	^
Çarpma	*
Bölme	/
Toplama	+
Çıkarma	-
Tam ve onda ayırma	.
Mantıksal İşlem Operatörleri	
Değil	‘
Ve	.
Veya	+

Karşılaştırma Operatörleri	
Eşittir	=
Eşit değildir	<>
Küçüktür	<
Büyüktür	>
Büyük eşittir	>=
Küçük eşittir	<=
Genel İşlem Operatörleri	
Aktarma	=
Parantez	()

Algoritmalarda Kullanılan Terimler

1) Tanımlayıcı
2) Değişken
3) Sabit
4) Aktarma
5) Sayaç
6) Döngü
7) Ardışık Toplama
8) Ardışık Çarpma

1) Tanımlayıcı

- Programcı tarafından oluşturulur.
- Programdaki değişkenleri, sabitleri, kayıt alanlarını, özel bilgi tiplerini vb. adlandırmak için kullanılan kelimeler
- Tanımlayıcılar, yerini tuttukları ifadelere çağrışım yapacak şekilde seçilmelidir.
- İngiliz alfabesindeki A-Z veya a-z arası 26 harften
- 0-9 arası rakamlar kullanılabilir
- Sembollerden sadece alt çizgi (_) kullanılabilir.
- Tanımlayıcı isimleri harfle veya alt çizgiyle başlayabilir.
- Tanımlayıcı ismi, rakamla başlayamaz veya sadece rakamlardan oluşamaz.

Algoritmalarda Kullanılan Terimler

2-)Değişken

- Programın her çalıştırılmasında, farklı değerler alan bilgi/bellek alanlarıdır.
- Değişken isimlendirilmeleri, yukarıda sayılan tanımlayıcı kurallarına uygun biçimde yapılmalıdır.

Örneğin ;

- Bir ismin aktarıldığı değişken ; **ad**
- Bir isim ve soy ismin aktarıldığı değişken; **adsoyad**
- Ev telefon no sunun aktarıldığı değişken; **evtel**
- Ev adresinin aktarıldığı değişken; **evadres**
- İş adresinin aktarıldığı değişken; **isadres**

3-) Sabit

Programdaki değeri değişmeyen ifadelere “sabit” denir. “İsimlendirme kuralları”na uygun olarak oluşturulan sabitlere, sayısal veriler doğrudan; alfa sayısal veriler ise tek/çift tırnak içinde aktarılır.

Algoritmalarda Kullanılan Terimler

Örnek Algoritma

Başla
Bir isim giriniz(A)
"İlk algoritmama Hoş geldin" mesajı (B)
B VE A yı Yaz
Dur

- Yukarıdaki algoritma, dışarıdan girilen bir A değişkeni ile B sabitini birleştirip ekrana yazar.

A değişkeni	*B sabiti*	*Sonuç*
Onur	İlk Algoritmama Hoş geldin	İlk Algoritmama Hoş geldin Onur
Emre	İlk Algoritmama Hoş geldin	İlk Algoritmama Hoş geldin Emre

Algoritmalarda Kullanılan Terimler

4-) Aktarma

- Herhangi bir bilgi alanına, veri yazma; herhangi bir ifadenin sonucunu başka bir değişkende gösterme vb görevlerde "aktarma" operatörü kullanılır.

değişken=ifade

- Değişken yazan kısım herhangi bir değişken ismidir.
- İfade yazan kısımda ise matematiksel,mantıksal veya alfa sayısal ifade olabilir.
- = sembolü, aktarma operatörüdür ve sağdaki ifadenin/işlemin sonucunu soldaki değişkene aktarır. Bu durumda değişkenin eğer varsa bir önceki değeri silinir.

1.işlem: sağdaki ifadeyi gerçekleştir veya sağdaki işlemi yap

2.işlem: Bulunan sonucu soldaki değişkene aktar.

Algoritmalarda Kullanılan Terimler

5-) Sayaç

Programlarda bazı işlemlerin belirli sayıda yaptırılması veya işlenen/üretilen değerlerin sayılması gerekebilir.

say=say+1

Bu işlemde sağdaki ifadede değişkenin eski değerine 1 eklenmekte; bulunan sonuç yine kendisine yeni değer olarak aktarılmaktadır.

Bu tür sayma işlemlerine algoritmada sayaç adı verilir.

Sayacın genel formülü;

Sayaç değişkeni=sayaç değişkeni+adım

Örnek; X=X+3

Örnek; S=S-5

Algoritmalarda Kullanılan Terimler

Örnek

- Aşağıdaki algoritmada 1-5 arası sayılar, ekrana yazdırılmaktadır. 1-5 arası sayıları oluşturmak için sayaç(s=s+1) kullanılmıştır.

1. **Başla**
2. **S=0**
3. **Eğer s>4 ise git 7**
4. **S=S+1**
5. **Yaz S**
6. **Git 3**
7. **Dur**

Eski S	Yeni S	Ekrana Yazılan
0	0+1=1	1
1	1+1=2	2
2	2+1=3	3
3	3+1=4	4
4	4+1=5	5

Algoritmalarda Kullanılan Terimler

6-) Döngü

- Bir çok programda bazı işlemler, belirli ardışık değerlerle gerçekleştirilmekte veya belirli sayıda yaptırılmaktadır. Programlardaki belirli işlem bloklarını, verilen sayıda gerçekleştiren işlem akış çevrimlerine “döngü” denir.
- Örneğin 1 ile 1000 arasındaki tek sayıların toplamını hesaplayan programda T=1+3+5 …. yerine 1 ile 1000 arasında ikişer artan bir döngü açılır ve döngü değişkeni ardışık toplanır.

Algoritmalarda Kullanılan Terimler

Örnek

Aşağıdaki algoritmada, 1 ile 10 arası tek sayıların toplamı hesaplanmaktadır.

1. **Başla**
2. **T=0**
3. **J=1**
4. **Eğer j>10 ise git 8**
5. **T=T+J**
6. **J=J+2**
7. **Git4**
8. **Dur**

DÖNGÜ

Eski J	Eski T	Yeni T	Yeni J
1	0	0+1=1	3
3	1	1+3=4	5
5	4	4+5=9	7
7	9	9+7=16	9
9	16	16+9=25	11
11	-	-	-

Algoritmalarda Kullanılan Terimler

7-) Ardışık Toplama

Programlarda, aynı değerin üzerine yeni değerler eklemek için kullanılır.

Toplam değişkeni=Toplam değişkeni + Sayı

Örnek: Klavyeden girilen 5 sayısının ortalamasını bulan programın algoritması.

1. Başla
2. T=0
3. S=0
4. Eğer S>4 ise git 9
5. S=S+1
6. Sayıyı (A) gir
7. T=T+A
8. Git 4
9. Ortalama=T/5
10. Yaz Ortalama
11. Dur

Algoritmalarda Kullanılan Terimler

8-) Ardışık Çarpma

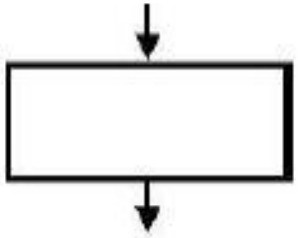
Ardışık veya ardışıl çarpma işleminde; aynı değer, yeni değerlerle çarpılarak eskisinin üzerine aktarılmaktadır.

Çarpım değişkeni=Çarpım değişkeni * Sayı

Örnek: Klavyeden girilen N sayısının faktöriyelini hesaplayan programın algoritmasını yazınız.

1. Başla
2. N sayısını gir
3. Fak=1
4. S=0
5. Eğer S>N-1 ise git 9
6. S=S+1
7. Fak=Fak*S
8. Git 5
9. Yaz Fak
10. Dur

Akış Diyagramlarında Kullanılan Temel Şekiller



Programın çalışması sırasında yapılacak işlemleri ifade etmek için kullanılan şekildir. İçine işlem cümleleri/ifadeleri aynen yazılır. Program akışı buraya geldiğinde, şeklin içerisindeki yazılı işlem gerçekleştirilir. Birden fazla işlem; aynı şekil içinde, aralarına virgül konularak veya alt alta yazılarak gösterilebilir.

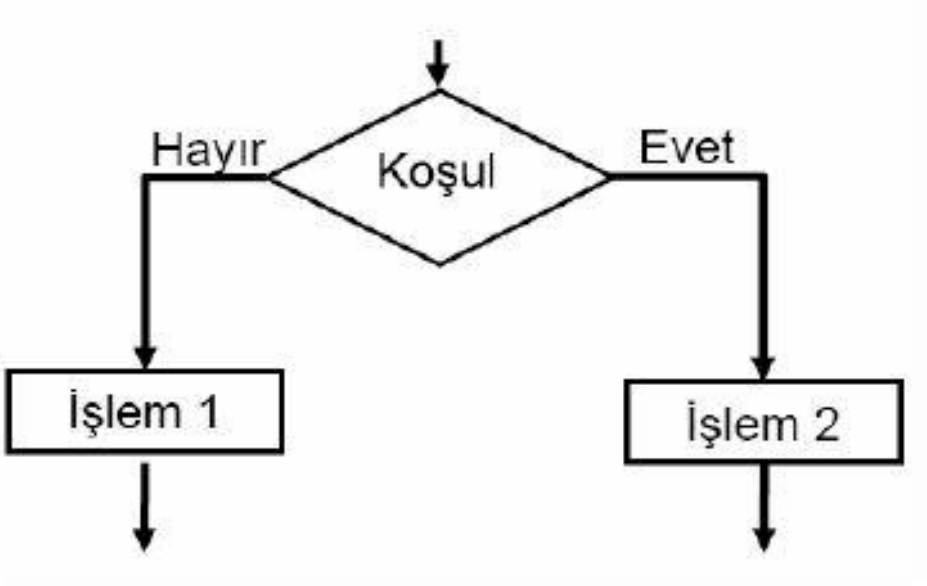
C=(a^2+b^2)^(1/2)

Örnek: İşlem akışı bu şekle gelince, program $c = \sqrt{a^2 + b^2}$ işlemini yapar. İfadedeki 'a' ve 'b' daha önceki adımlarda girilmiş olan değerlerdir.

Akış Diyagramlarında Kullanılan Temel Şekiller

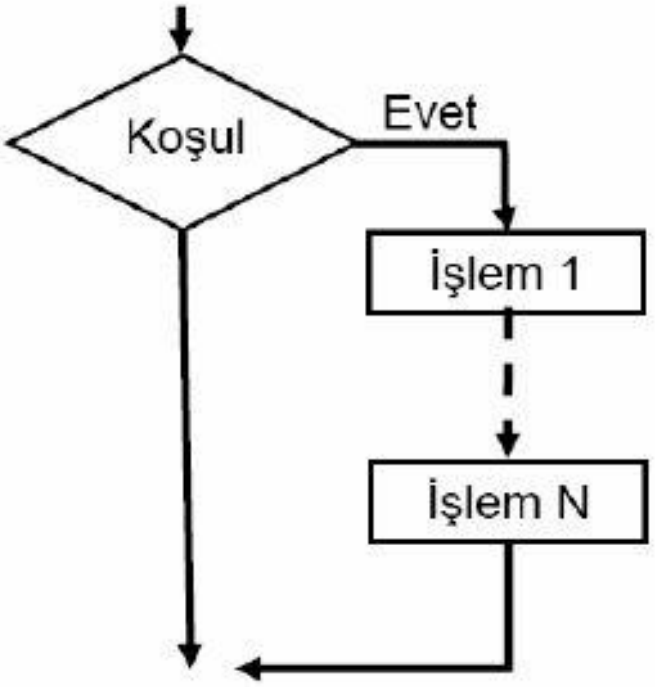
Karar Verme

1. Durum:



a) Koşulun durumuna bağlı olarak 2 farklı işlem vardır.

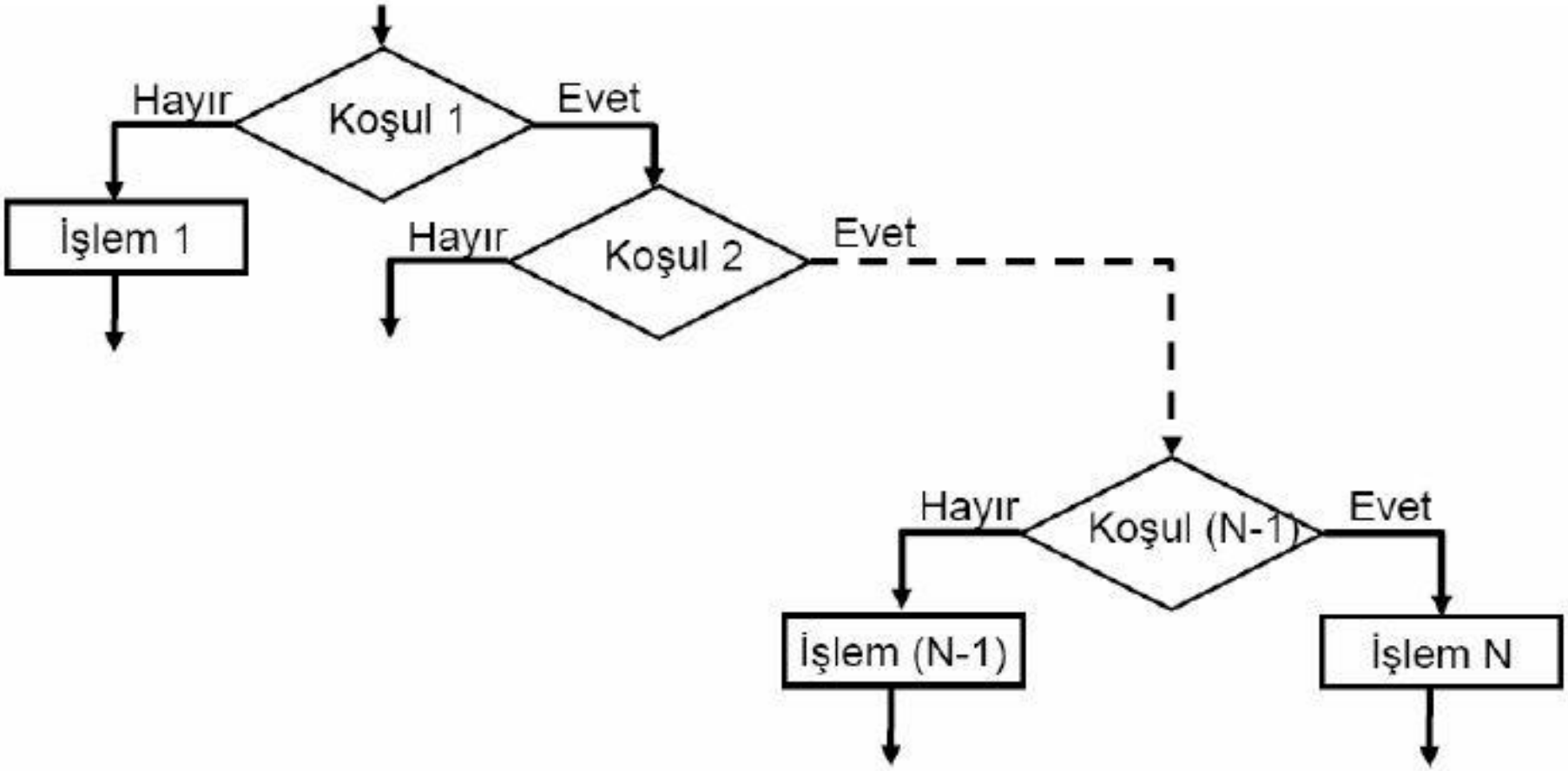
2. Durum:



b) Olumsuz koşulda yapılacak işlem yoktur; olumlu olması durumunda ise N adet işlem yapılacaktır.

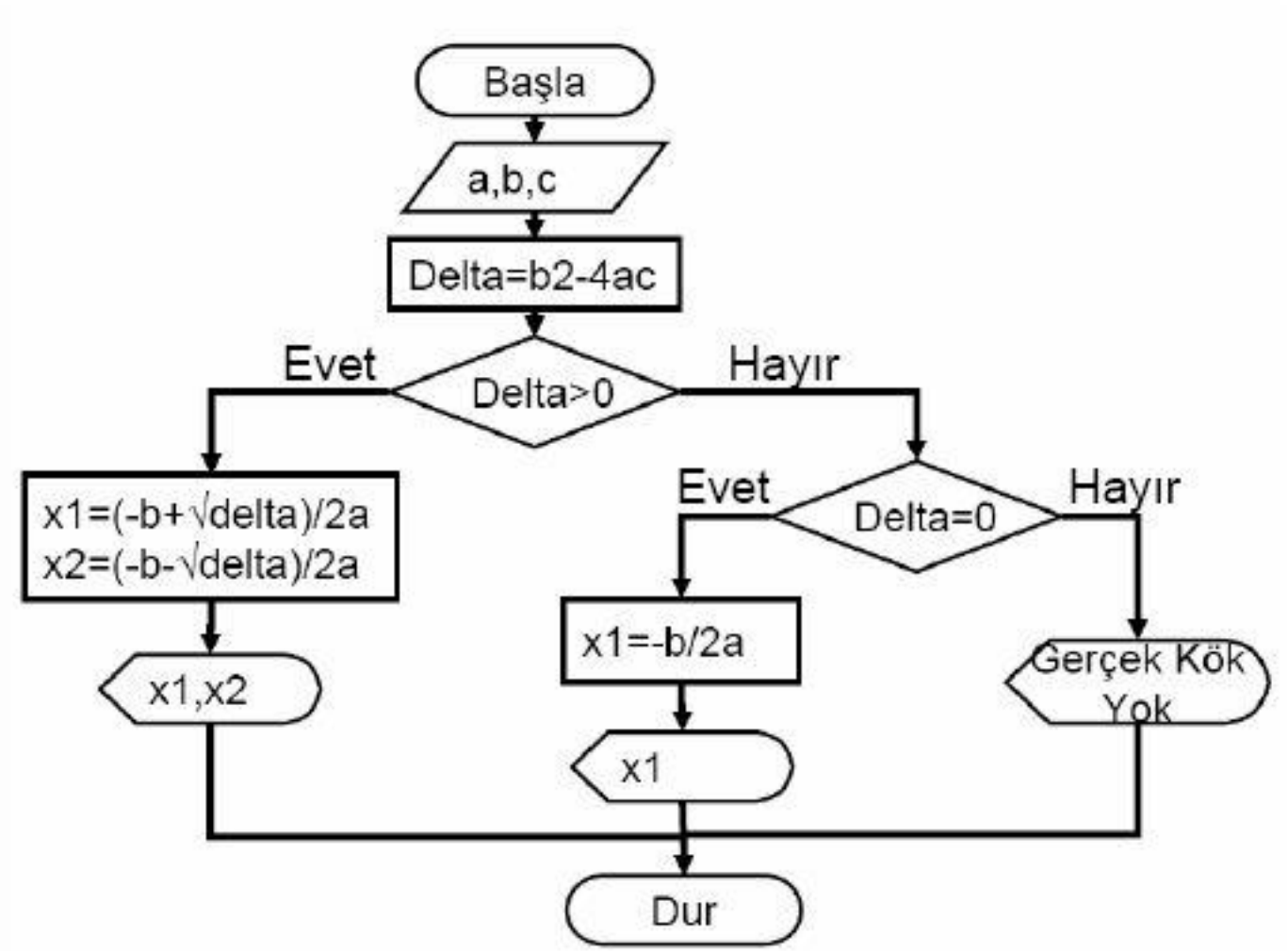
Akış Diyagramlarında Kullanılan Temel Şekiller

Bu yapıyı art arda birden çok kez kullanıp aşağıdaki gibi bir kaşılaştırma dizisi oluşturulabilir.



Akış Diyagramlarında Kullanılan Temel Şekiller

Örnek: ax2+ bx + c = 0 şeklindeki ikinci dereceden bir denklemin köklerini bulan algoritmayı tasarlayıp akış şeması ile gösteriniz.



Akış Diyagramlarında Kullanılan Temel Şekiller

Döngü Yapısı

Bu yapı kullanılırken, döngü sayacı, koşul bilgisi ve sayacın artım bilgisi verilmelidir. Döngü sayacı kullanılmıyorsa sadece döngüye devam edebilmek için gerekli olan koşul bilgisi verilmelidir.

Genel olarak çoğu programlama dilinin döngü deyimleri ;

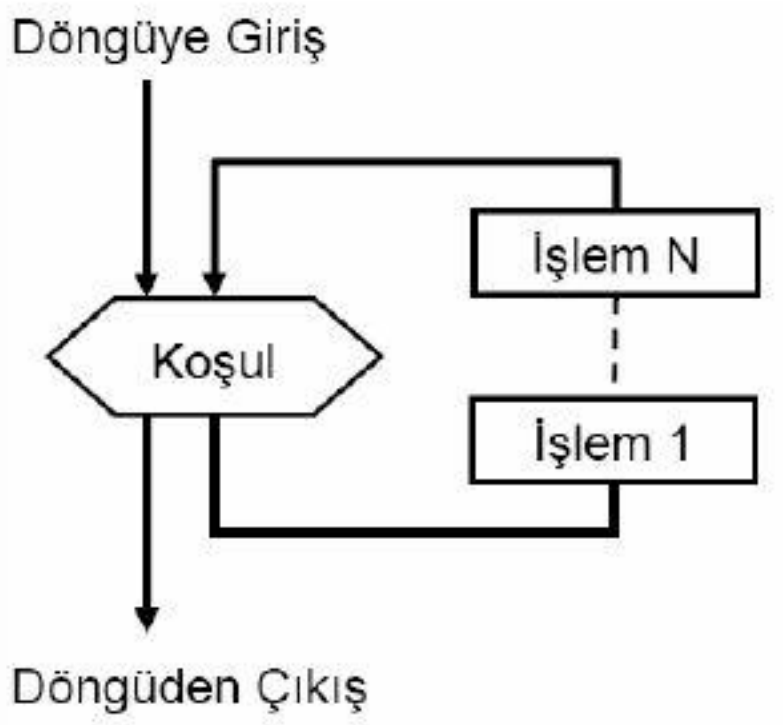
- **While**
- **Do-while**
- **For**

gibi yapılar üzerine kurulmuştur. Farklı dillerde bu yapılara farklı alternatifler olsa da döngülerin çalışma mantığı genel olarak benzerdir.

Akış Diyagramlarında Kullanılan Temel Şekiller

1. Durum (While)

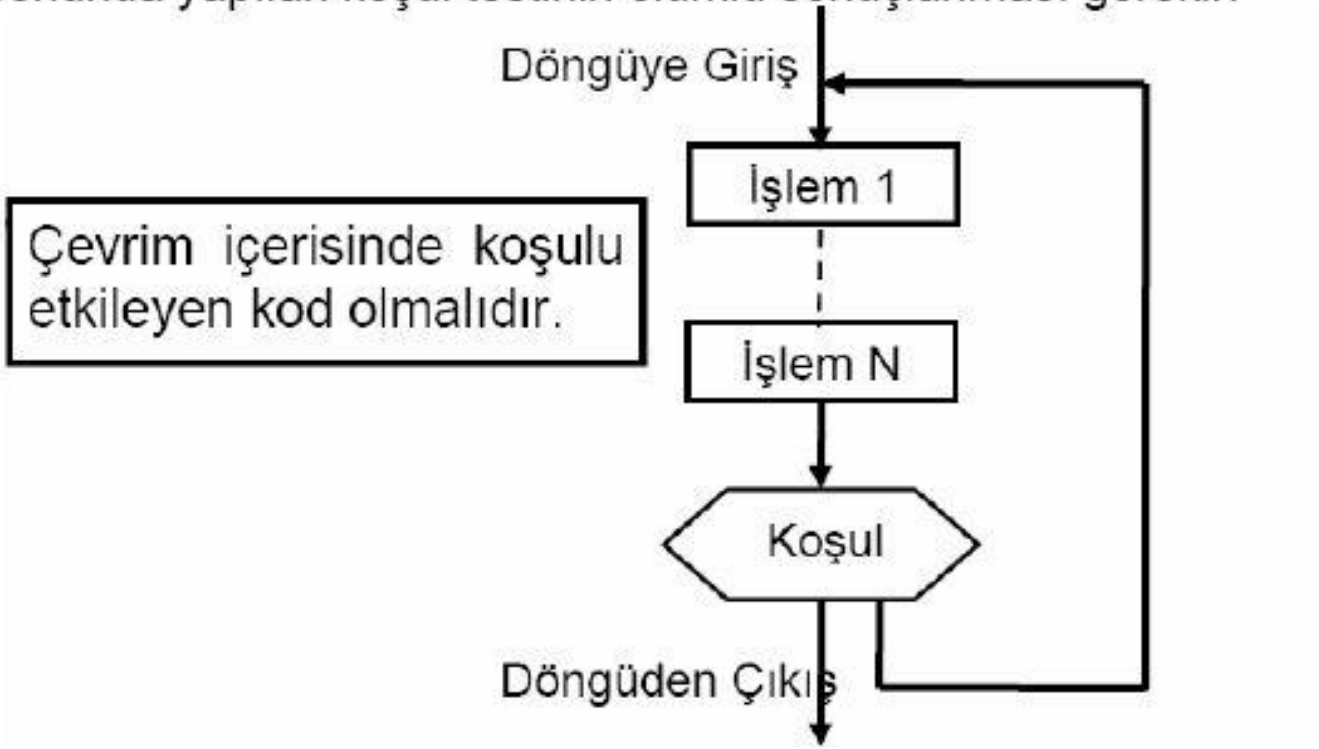
Koşul daha çevrim içerisine girmeden sınanır. Koşul olumsuz olduğunda çerime hiç girilmez ve döngü içerisinde yapılması gerekenler atlanır.



Akış Diyagramlarında Kullanılan Temel Şekiller

2. Durum (Do-While)

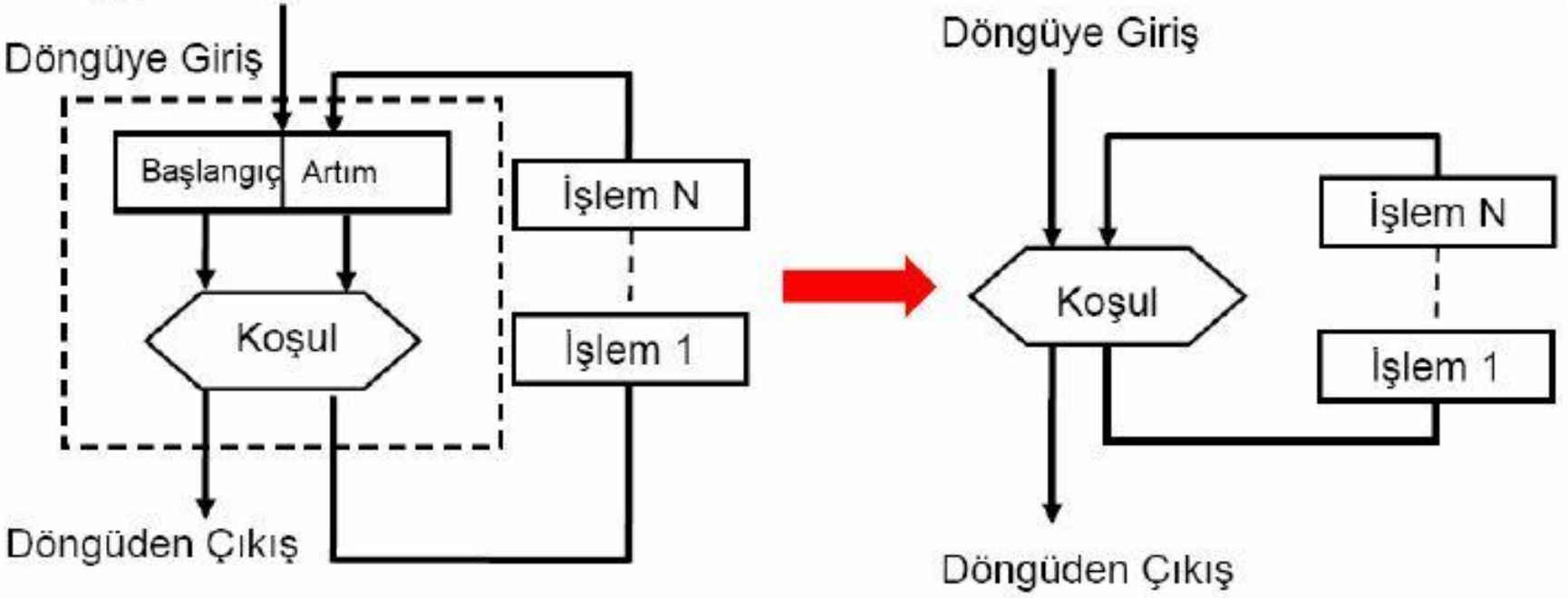
Bu döngü deyiminde, çevrim en az bir defa olmak üzere gerçekleşir. Çünkü koşul sınaması döngü sonunda yapılmaktadır. Eğer koşul sonucu olumsuz ise bir sonraki çevrime geçilmeden döngüden çıkılır. Çevrimin devam edebilmesi için her döngü sonunda yapılan koşul testinin olumlu sonuçlanması gerekir.



Akış Diyagramlarında Kullanılan Temel Şekiller

3. Durum (For)

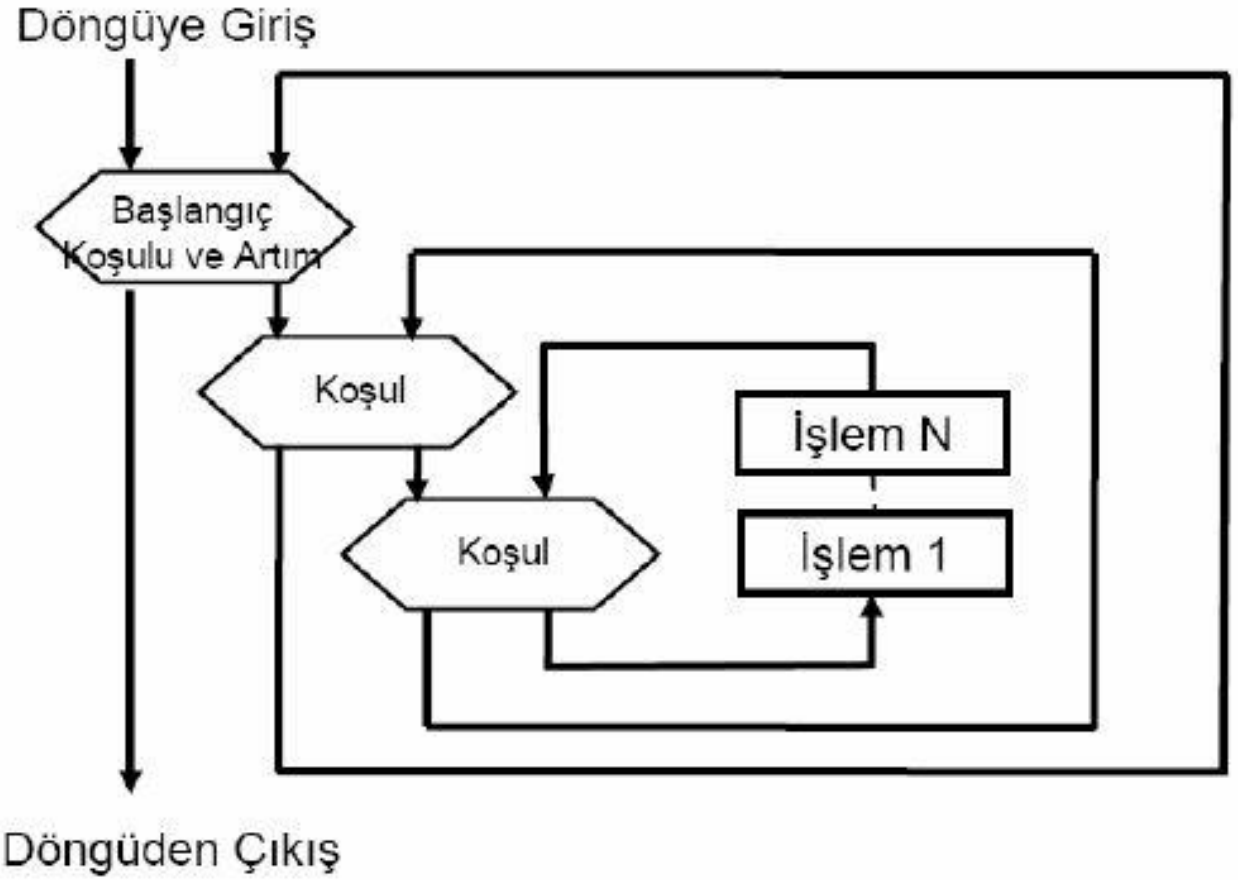
Diğer deyimlerden farklı olarak, döngü sayacı doğrudan koşul parametreleri düzeyinde verilir.
Döngü girmeden önce sayaç değişkenine başlangıç değeri atanmakta ve daha sonra koşula bakılmaktadır. Döngü içerisinde belirtilen işlemler yapıldıktan sonra sayaç değişkeni arttırılmaktadır.



Akış Diyagramlarında Kullanılan Temel Şekiller

İç içe Döngülerin Kullanılması

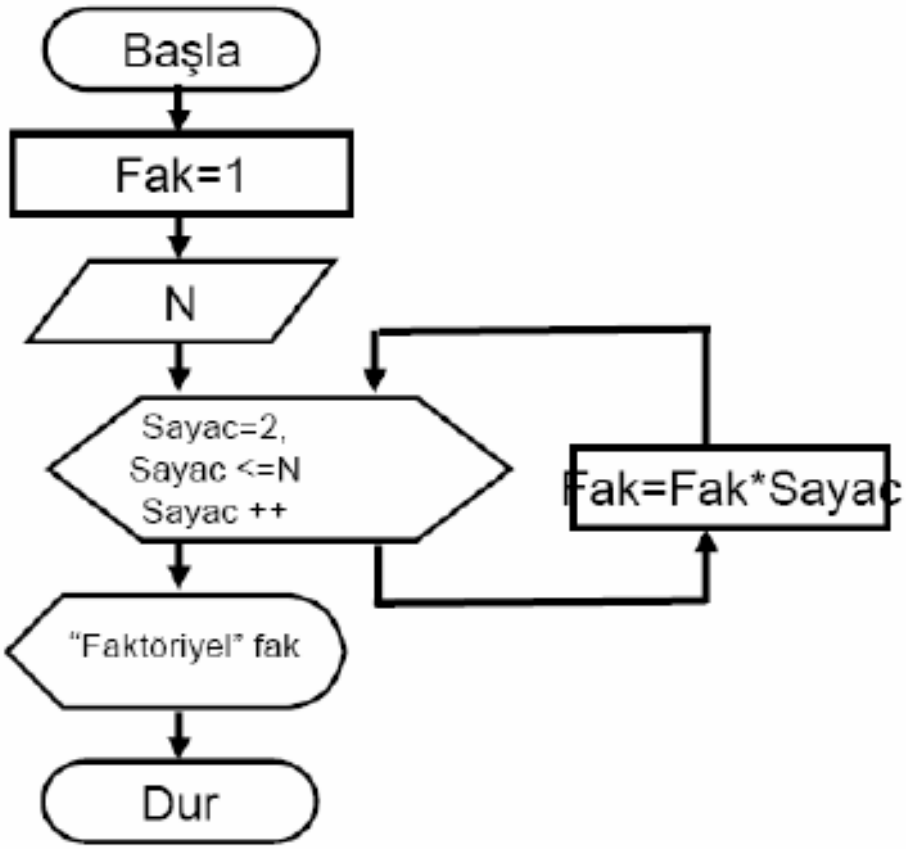
İçiçe döngü kurulurken en önemli unsur, içteki döngü sonlanmadan bir dıştaki döngüye geçilmemesidir. Diğer bir deyişle döngüler birbirlerini kesmemelidir.



En içteki döngü bir dıştaki döngünün her adımında N kez tekrarlanır.

Akış Diyagramlarında Kullanılan Temel Şekiller

Örnek: Klavyeden girilen N sayısının faktöriyelini alan algoritmanın akış diyagramını çiziniz.



- N ile hangi sayının faktöriyelin hesaplanacağı belirlenir ve N çevrimlik bir döngü kurulur.
- İlk çevrimde 1!, ikinci çevrimde 2! ve sırayla N'inci çevrim sonucunda da N! değeri hesaplanmış olur.
- Sayac>N koşulu oluştuğunda döngüden çıkılır ve elde edilen sonuç dış ortama aktarılır.

Matematiksel işlemler

- Temel aritmetik işlemler
  - toplama,çıkarma,çarpma,bölme
- Matematiksel fonksiyonlar
  - Üstel,logaritmik,trigonometrik, hiperbolik vb,

İşlem	Matematik	Bilgisayar
Toplama	a + b	a + b
Çıkarma	a - b	a - b
Çarpma	a . b	a * b
Bölme	a ÷ b	a / b
Üs Alma	a^b	a ^ b

Matematiksel işlemler

❖ Matematiksel İşlemlerde Öncelik Sıraları

Sıra	İşlem	Bilgisayar
1	Sayıların Negatifliği	-…
2	Parantezler	(…..)
3	Matematiksel Fonksiyonlar	cos, sin, log, …
4	Üs alma	a ^ b, pow, …
5	Çarpma ve Bölme	a * b ve a/b
6	Toplama ve Çıkarma	a + b ve a - b

- NOT: Bilgisayar diline kodlanmış bir matematiksel ifadede, aynı önceliğe sahip işlemler mevcut ise bilgisayarın bu işlemleri gerçekleştirme sırası soldan sağa(baştan sona) doğrudur.
- Örneğin ; Y=A*B/C

Önce A*B işlemi yapılacak, ardından bulunan sonuç C ye bölünecektir.

Matematiksel işlemler

Matematiksel Yazılım	Bilgisayara Kodlanması
a+b-c+2abc-7	a+b-c+2*a*b*c-7
$a+b^2-c^3$	a+b^2-c^3
$a-\frac{b}{c}+2ac-\frac{2}{a+b}$	a-b/c+2*a*c-2/(a+b)
$\sqrt{a+b}-\frac{2ab}{b^2-4ac}$	(a+b)^(1/2)-2*a*b/(b^2-4*a*c)
$\frac{a^2+b^2}{2ab}$	(a^2+b^2)/(2*a*b)

Matematiksel işlemler

- **Örnek:**
  - **Matematiksel ifade :**

$$x = a \cdot b / c + d \cdot e^{f} - g$$

  - **Bilgisayar ifadesi:**

x = a * b / c + d * e ^ f - g

2 3 4 2 1 5

- **Not:** İşlem önceliklerine dikkat edilmez ise aynı gibi görünen ifadeler, farklı sonuçlar verebilir.

Karşılaştırma

- İki büyüklükten hangisinin büyük veya küçük olduğu,
- İki değişkenin birbirine eşit olup olmadığı gibi konularda karar verebilir.

Karşılaştırma İşlemlerinin Bilgisayar Dilindeki Karşılıkları	
Sembol	Anlamı
= veya ==	Eşittir
<> Veya !=	Eşit Değildir
>	Büyüktür
<	Küçüktür
>= veya =>	Büyük eşittir
<= veya =<	Küçük eşittir

- **Örnek:** **Eğer A > B ise Yaz " A sayısı B'den büyüktür "**

Karşılaştırma

- Sayısal karşılaştırmalarda doğrudan değerler karşılaştırılır.
  - Örnek: 25 > 15 "25 sayısı 15 ten büyüktür"

- Karakter olarak karşılaştırmalarda ise karşılaştırma işlemine ilk karakterlerden başlanılarak sıra ile karşılaştırılır.
  - Örnek: a > c "1. karakter alfabetik olarak daha önde"

❖ **Not:** Karakter karşılaştırma işlemlerinde, karşılaştırma karakterler arasında değil, karakterlerin ASCII kodları arasında yapılır. Örneğin, A nın ASCII kod karşılığı 65 tir. a nın ise ASCII kod karşılığı 97 tir. Büyük ve küçük harfler arasındaki ASCII kod farkı ise 32'dir.

Mantıksal İşlemler

Temel Mantıksal İşlem Karşılıkları

İşlem	Komut	Matematiksel Sembol	Anlamı
VE	AND	.	Koşulların hepsi doğru ise sonuç doğrudur
VEYA	OR	+	Koşullardan en az biri doğru ise sonuç doğrudur
DEĞİL	NOT	’	Sonuç koşulun tersidir. 1 ise 0 dır

Mantıksal İşlemlerde Öncelik Sıraları

Sıra	İşlem	Komut
1	Parantez İçindeki İşlemler	(…..)
2	DEĞİL	NOT
3	VE	AND
4	VEYA	OR

Mantıksal İşlemler

- "ve,veya,değil " operatörleri hem matematiksel işlemlerde hem de karar ifadelerinde kullanılırlar.

Mantıksal işlem	Matematiksel sembol	Komut
Ve	.	And
Veya	+	Or
değil	'	Not

- Bütün şartların sağlatılması isteniyorsa koşullar arasına VE
- Herhangi birinin sağlatılması isteniyorsa koşullar arasına VEYA
- Koşulu sağlamayanlar isteniyorsa DEĞİL mantıksal operatörü kullanılır.

Mantıksal İşlemler

- VE Bağlacı: Ve bağlacı ile söylenmek istenen her iki koşulun da sağlanmasıdır. VE bağlacı ile bağlanmış önermelerden en az birinin yanlış olması sonucu yanlış yapar.
- VEYA Bağlacı: VEYA bağlacı ile bağlanan koşullardan en az birisi doğru ise sonuç doğrudur. İkiden fazla önermeler için, önermelerden en az birinin doğru olması sonucu doğru yapar.
- DEĞİL Bağlacı: DEĞİL bağlacı doğruyu yanlış, yanlışı doğru yapar. DEĞİL tek bir önerme veya koşul üzerinde uygulanır. VE, VEYA ise iki önerme veya koşul üzerinde uygulanır. Doğru=1 ve Yanlış=0 tanımıyla, aşağıdaki tabloda bağlaçların x ve y'nin alacağı değerlere göre sonuçları gösterilmiştir.

Mantıksal İşlemler

- Örnek; Bir işyerinde çalışan işçiler arasından yalnızca yaşı 23 üzerinde olup, maaş olarak asgari ücret alanların isimleri istenebilir.

  Burada iki koşul vardır ve bu iki koşulun da doğru olması gerekir. Yani;

Eğer Yaş>23 VE maaş=asgari ücret ise ismi Yaz

1.KOŞUL *2.KOŞUL*

- *Yaz komutu 1. ve 2.koşulun her ikisi de sağlanıyorsa çalışır.*

Mantıksal İşlemler

- Örnek; Bir sınıfta Bilgisayar dersinden 65 in üzerinde not alıp, Türk Dili veya Yabancı Dil derslerinin herhangi birinden 65 in üzerinde not alanların isimleri istenmektedir.

  Burada 3 koşul vardır ve Bilgisayar dersinden 65 in üzerinde not almış olmak temel koşuldur. Diğer iki dersin notlarının herhangi birinin 65 in üzerinde olması gerekir.

Eğer ;

Bilg.Not>65 ve (TDili not>65 veya YDil not>65) ismi Yaz

Mantıksal İşlemler

X	Y	X VE Y	X VEYA Z	Z	DEĞİL Z
0	0	0	0	0	1
0	1	0	1	1	0
1	0	0	1		
1	1	1	1		

İyi Çalışmalar…