ALGORİTMA VE PROGRAMLAMA (Bölüm 1)

Prof. Dr. Hakan Ündil - Elektronik Teknolojisi - 2015-2016 Güz

Bölüm 1 - 2

PROBLEM ÇÖZME ve ALGORİTMALAR

- Bir problemi çözmek için yapılacak **işlemler** **en genel** şekilde **iki adımda** incelenebilir:

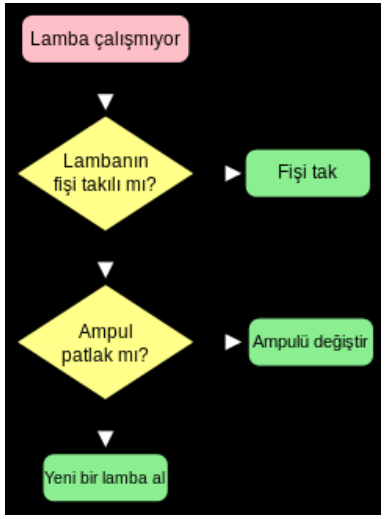
1. **Algoritma** *(Problemi Çözme Yolu)*

2. **Kodlama** *(Gerçekleştirme)*

PROBLEM ÇÖZME ve ALGORİTMALAR *(Devam)*

- **1. Algoritma**

  - Problemin çözümünü tanımlayan **adımların sırasıyla** elde edilmesidir.



PROBLEM ÇÖZME ve ALGORİTMALAR *(Devam)*

- Her **Algoritma** Aşağıdaki **Kriterleri Sağlamalıdır:**

1. Girdi : Dışarıdan Sıfır veya farklı bir değer verilmeli.

2. Çıktı : En azından bir değer üretilmeli.

3. Açıklık : Her işlem (komut) açık olmalı ve farklı anlamlar içermemeli.

4. Sonluluk : Her türlü ihtimal için algoritma sonlu adımda bitmeli.

5. Etkinlik : Her komut, kişinin kalem ve kağıt ile yürütebileceği kadar basit olmalıdır.

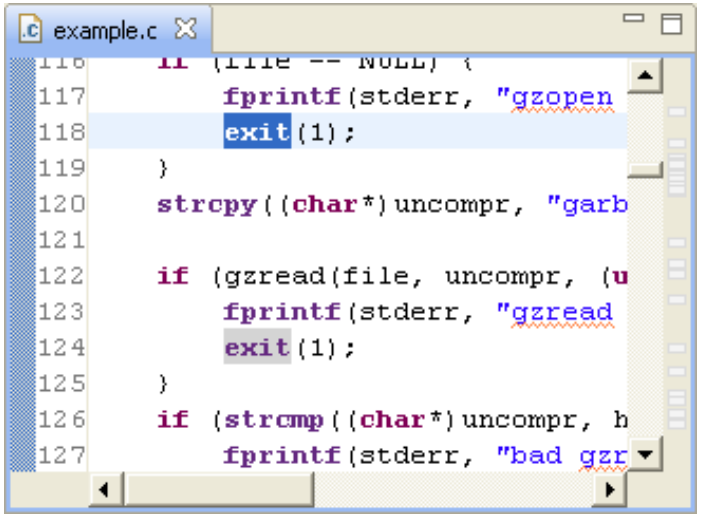
PROBLEM ÇÖZME ve ALGORİTMALAR *(Devam)*

- **Algoritma yapma** Sırasında **Dikkat Edilecek** Hususlar**:**

- Doğruluğu kesin olarak ispatlanmadıkça, hiçbir şeyi doğru olarak kabul etmeyin; **tahmin ve önyargılardan kaçının**.
- Karşılaştığınız her güçlüğü mümkün olduğu kadar **çok parçaya bölün**.
- Anlaşılması en kolay olan (**basit) şeylerle başlayıp** yavaş yavaş daha **zor ve karmaşık olanlara doğru** ilerleyin.
- Önce olaya **bakışınız çok genel**, sonra hazırlayacağınız ayrıntılı liste (detaylı algoritma) ise **hiçbir şeyi dışarıda bırakmayacak kadar kusursuz ve eksiksiz** olsun.

PROBLEM ÇÖZME ve ALGORİTMALAR *(Devam)*

- **2. Kodlama**
  - Bir programlama diliyle bu **Algoritma'nın programa (yazılıma)** dönüştürülmesidir.



PROBLEM ÇÖZME ve ALGORİTMALAR *(Devam)*

- **Kodlama (Programlama) Dili Seçerken;**
  - Dil yapılacak işe uygun özelliklere sahip mi ?
  - Diğer dillerle uyumluluğu ve bu dillere dönüştürme imkanları nedir?
  - Öğrenmesinin kolay/zor olup olmadığı, örnekler var mı?
  - Piyasada, bu dili kullanan, yeteri kadar iş gücü(kişi) olup olmadığı gibi hususlara dikkat edilmelidir.

PROBLEM ÇÖZME ve ALGORİTMALAR *(Devam)*
Çözme Aşamaları

- **Problem Çözme Aşamaları: *(Daha Ayrıntılı)***

1. Sistem İhtiyaçları:

- **Sistem ihtiyaçlarına** göre **problemin tam olarak ne olduğunun** anlaşılmasıdır.
- **Problemin çözümünden neler beklendiği** bilinmelidir.
- **Çözümün girdi ve çıktılarının neler olacağı kesin** olarak belirlenmelidir.

PROBLEM ÇÖZME ve ALGORİTMALAR *(Devam)*
Çözme Aşamaları

2. <u>Analiz:</u>

- Sistem **ihtiyaçları** belirlendikten sonra bunlar **temel parçalara** ayrılarak bu parçalar ayrı ayrı değerlendirilir.
- Bu **parçalar arasındaki ilişkiler** tanımlanır.
- Bu suretle **sonuca gitmeye** çalışılır.

PROBLEM ÇÖZME ve ALGORİTMALAR(Devam)
Çözme Aşamaları

3. **Tasarım (Algoritma):**

- Kullanılacak **çözüm adımlarını gösteren Algoritma adı verilen** bir liste yapılır…
- Böyle bir **algoritma tasarlanırken ilk önce ana (genel) adımlar** çıkarılır.
- Daha sonra her adım için **daha ayrıntılı bir algoritmaya** geçilir.

PROBLEM ÇÖZME ve ALGORİTMALAR(Devam)
Çözme Aşamaları

4. **Kodlama (Programlama):**

- Geliştirilen **algoritma bir programlama diline** çevrilir.
- Bu dilin, programın yazımını **aşırı zorlaştırmayacak ve karmaşıklaştırmayacak** bir dil olması **tercih** edilmelidir.
- Programı yazacak olanın bilgisi de dikkate alınır.

PROBLEM ÇÖZME ve ALGORİTMALAR(Devam)
Çözme Aşamaları

5. Sistem Testi:

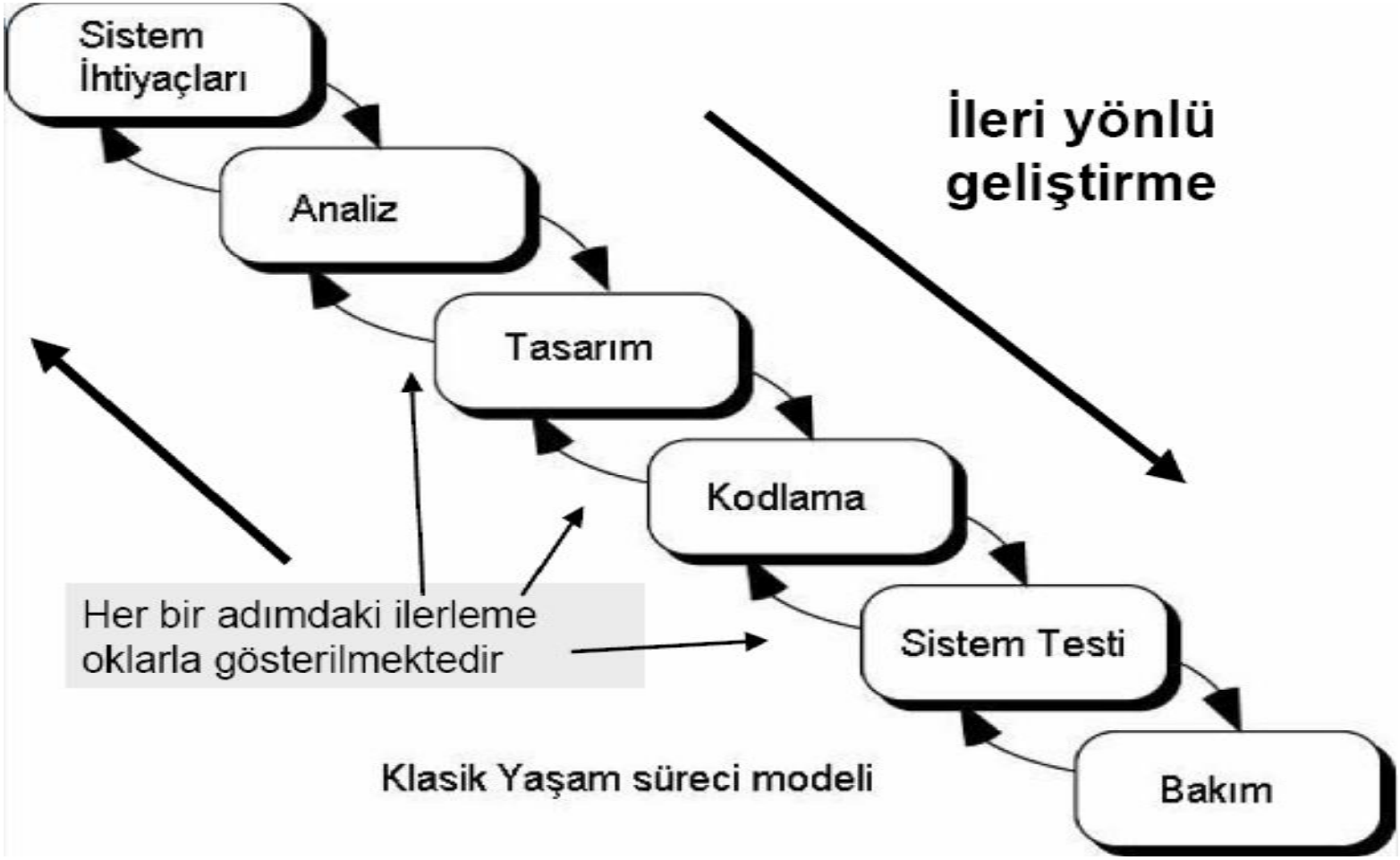
- Program **değişik girdiler ile çalıştırılır…**
- Programın ürettiği **sonuçlar (çıktılar) kontrol edilerek test** işlemi gerçekleştirilir.
- **Sonuçlar beklendiği gibi** ise **programın doğru çalıştığı** ispatlanmış olunur…
- Değilse **doğru çalışmayan kısımları tespit** edilerek düzeltilir.

PROBLEM ÇÖZME ve ALGORİTMALAR *(Devam)*
Çözme Aşamaları

6. <u>Bakım:</u>

- Programın **güncel şartlara göre yeniden düzenlenmesini** içeren bir konudur.
- Oluşan **hataların giderilmesi …**
- Yeni **eklemeler yapılması…**
- Ya da **programın teknolojisinin yenilenmesi** gibi işlemlerdir.

PROBLEM ÇÖZME ve ALGORİTMALAR*(Devam)*
Çözme Aşamaları (Adımlar)



- ***Herhangi bir adımda sorun yaşanırsa (gerektiğinde) bir adım geri dönülür…***

Problem Çözmede Algoritma ve Programlama

- Yukarıda açıklanan Problem Çözme Aşamalarından **ikisi** dersimizin konusunu ile **doğrudan ilgilidir** :
  - **Algoritma** (Tasarım)
  - **Programlama** (Kodlama)

Bu başlıkları daha **ayrıntılı** olarak inceleyelim…

PROBLEM ÇÖZME ve ALGORİTMALAR *(Devam)*
Algoritma Türleri

- **Algoritma Türleri:**
  - **İki farklı şekilde ifade edilebilirler:**

    1. Pseudo *(Sözde)* Kod

    2. Akış Diyagramı *(Akış Şeması)*

PROBLEM ÇÖZME ve ALGORİTMALAR *(Devam)*
Psedo (Sözde) Kod

1. <u>Pseudo (Sözde) Kod:</u> Bir algoritmanın;

- Hem **programlama dili kuralları…**
- Hem de **konuşma dili kullanılarak** ortaya koyulması/tanımlanmasıdır.

***Not: İyi bir biçimde** yazılmış, **Psedo Kod'dan, programlama diline** kolaylıkla geçilebilir.*

PROBLEM ÇÖZME ve ALGORİTMALAR *(Devam)*
Psedo Kod *(Devam)*

- **Pseudo Kod…**
  - Doğrudan **konuşma dilinde…**
  - **Programlama mantığı** altında…
  - **Eğer, iken** gibi şart kelimeleri ve **…**
  - **Büyük, Eşit, Küçük ise vb.** ifadeler ile yazılır.

Not: *Aynı Problem için* ***çok farklı Psedo Kod*** *Yazılabilir.*

PROBLEM ÇÖZME ve ALGORİTMALAR *(Devam)*

Psedo Kod *(Devam)*

- **Örnek 1:** Verilen **iki sayının toplamı**nın bulunarak bu **toplamın yazdırılması** algoritmasını **psedo (sözde) kod** ile yazınız.

1. **Başla**
2. **Birinci Sayıyı Oku**
3. **İkinci Sayıyı Oku**
4. **İki Sayıyı Topla**
5. **Toplamı yaz**
6. **Dur**

PROBLEM ÇÖZME ve ALGORİTMALAR *(Devam)*

Psedo Kod *(Devam)*

- **<u>Örnek 2</u>:** Verilen bir sıcaklık derecesine göre suyun durumunu belirten bir algoritmayı psedo (sözde) kod ile yazınız.

  Not 1: *Programın,* ***sıcaklığa göre*** ***Su, Buz, Buhar*** *şeklinde* ***Suyun durumunu*** *göstermesi istenmektedir*.

  Not 2: *Önce programın ne yaptığına dâir* ***açıklama mesajı*** *sonra da* ***sıcaklık değerinin girilmesi için uyarı*** *mesajı yazılsın.*

PROBLEM ÇÖZME ve ALGORİTMALAR *(Devam)*

Psedo Kod *(Devam)*

1. Başla

2. Programın görevini açıklama mesajı yaz

3. Kullanıcının sıcaklığı girmesi için bir uyarı mesajı yaz

4. Girilen **Sıcaklığı Oku**

5. Eğer **Sıcaklık** sıfırdan Küçük ise

Durum=“Buz”

6. Eğer **Sıcaklık 100’**e Eşit yada Büyük ise

Durum= “Buhar”

7. Değilse

Durum =“Su”

8. Durum sonucunu Yaz.

9. Son

PROBLEM ÇÖZME ve ALGORİTMALAR (*Devam*)

Psedo Kod *(Devam)*

- **Programın Çalışması:** *(140 derece örneği için)*

Bu Program, sıcaklığa göre Su, Buz, Buhar şeklinde suyun durumunu gösterir. ← *(Açıklama mesajı)*

Lütfen derece cinsinden sıcaklığı giriniz: **140**

↑ *(Uyarı mesajı)*

↑ *(Girdi)*

BUHAR ← *(Çıktı)*

PROBLEM ÇÖZME ve ALGORİTMALAR *(Devam)*

Psedo Kod *(Devam)*

- **Örnek 3:** Bir Pseudo (sözde) Kod yazarak bir öğrencinin dönem sonu **not ortalamasını hesaplayıp** öğrencinin **dersten geçip geçmeyeceğini** tespit edin.

Not : *Not ortalaması* ***eşit ağırlıklı 4 nottan*** *oluşsun ve* ***Geçme notu 50*** *olsun.*

PROBLEM ÇÖZME ve ALGORİTMALAR *(Devam)*

Psedo Kod *(Devam)*

1. Başla

2. "***Öğrencinin dersten aldığı 4 notu Girin***" uyarı mesajını yaz.

3. Ortalamayı hesaplamak üzere notları topla ve 4'e böl.

4. Eğer Ortalama 50 'nin altında ise
"**Başarısız**" yaz.

5. Değilse *(Aksi halde)*
"**Başarılı**" yaz.

6. Son

PROBLEM ÇÖZME ve ALGORİTMALAR *(Devam)*

Psedo Kod *(Devam)*

- Aynı Psedo Kodu Programlama Diline **daha yakın tarzda farklı şekilde** tekrar yazalım:

1. Başla
2. M1,M2,M3,M4 Öğrencinin Notlarını Girin
3. ORTALAMA = **(M1+M2+M3+M4)/4**
4. Eğer (**ORTALAMA < 50**) ise
"**Başarısız**" yaz
5. Değilse *(Aksi halde)*
"**Başarılı**" yaz
6. Son

PROBLEM ÇÖZME ve ALGORİTMALAR *(Devam)*
Akış Diyagramı

- **2. Akış Diyagramı** *(Akış Şeması):*

- Algoritmanın **görsel/şekilsel** olarak ortaya koyulmasıdır.

- Yapılması gerekenleri **geometrik şekiller**den oluşan **sembollerle** gösterir.

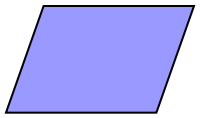
PROBLEM ÇÖZME ve ALGORİTMALAR *(Devam)*
Akış Diyagramı *(Devam)*

- Akış Diyagramı kullanılması ile program **Görsellik kazanacağından** Programın **takip edilmesi** daha da **kolaylaşır**.

PROBLEM ÇÖZME ve ALGORİTMALAR *(Devam)*
Akış Diyagramı *(Devam)*

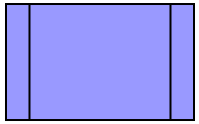
SEMBOLLER (Sık Kullanılanlar) :

Terminal Sembolü:
Akış Diyagramında Başlangıç ya da Bitiş belirtir.

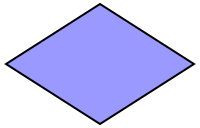


Giriş İşlemi Sembolü:
Bir Giriş işlemini temsil eder.

İşlem Sembolü:
Algoritmada bir işlemi temsil eder



Ön Tanımlı İşlem Sembolü:
Daha önce tanımlı bir işlemi temsil eder



Karar Sembolü:
İki değerin mukayesesi dahil karar işlemini temsil eder

Çıkış İşlemi Sembolü:
Bir Çıkış işlemini temsil eder.

PROBLEM ÇÖZME ve ALGORİTMALAR *(Devam)*
Akış Diyagramı *(Devam)*

- **<u>Örnek 4</u>:** Girilen bir uzunluğa ait inch (inç) değerini santimetreye çeviren hem bir **Psedo kod** yazıp hem de bir **Akış Diyagramı çiziniz.**

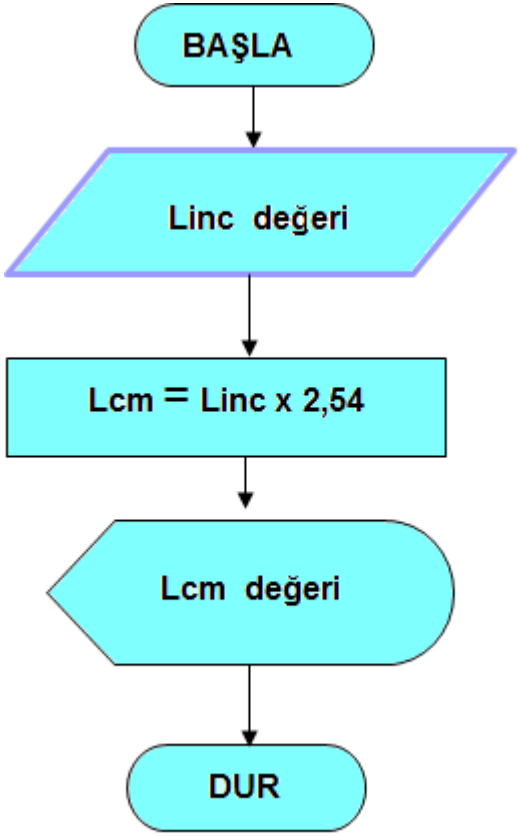
  Not: *1 inch = 2.54 cm dir.*

PROBLEM ÇÖZME ve ALGORİTMALAR *(Devam)*
Akış Diyagramı *(Devam)*

Psedo Kod

1. Başla
2. inch olarak uzunluğu (Linc) girin.
3. inch değerini 2,54 ile çarparak santimetreye çevir.
4. Uzunluğu santimetre olarak (Lcm) yaz.
5. Son

Akış Diyagramı



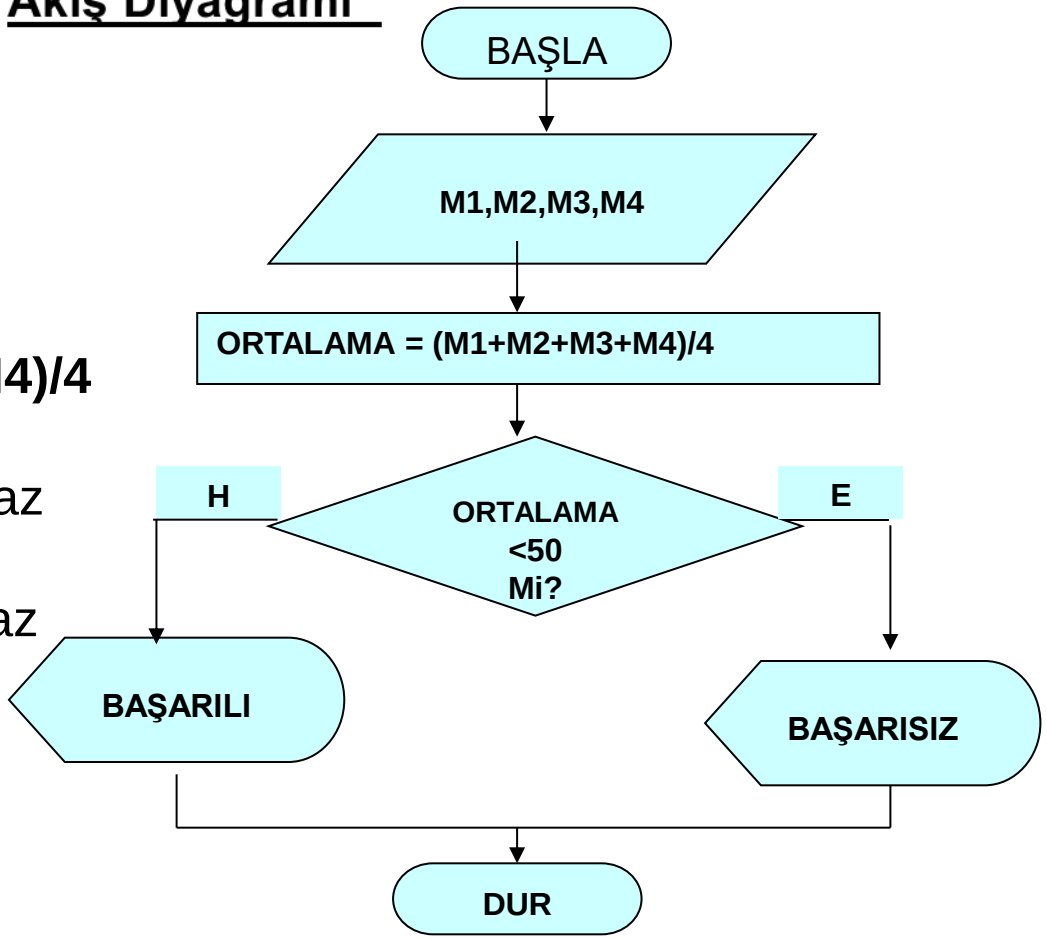
PROBLEM ÇÖZME ve ALGORİTMALAR *(Devam)*
Akış Diyagramı *(Devam)*

- **Öğrencinin Ortalamasını hesaplayıp durumunu belirleyen Örnek 3** için;

Psedo Kodu *(yazılmıştı)*

1. Başla
2. M1,M2,M3,M4 değerlerini girin
3. ORTALAMA = (M1+M2+M3+M4)/4
4. Eğer (**ORTALAMA <50**) ise
"**BAŞARISIZ**" yaz
5. Değilse*(aksi halde)*
"**BAŞARILI**" yaz
6. Son

Akış Diyagramı



PROBLEM ÇÖZME ve ALGORİTMALAR *(Devam)*
Akış Diyagramı *(Devam)*

- **Örnek 5 :** $ax^2 + bx + c = 0$ şeklindeki **ikinci derece** bir denklemin köklerini bulmak için bir **psedo kod yazarak akış diyagramı** çizin.

➢ **Hatırlatma:** [**d:** diskriminant] olmak üzere

d = sqrt ($b^2 - 4ac$**)** olmak üzere (*sqrt: Karekök)*

Denklemin Kökleri;

x*1 = (–*b* + d)*/*2*a *ve*

x*2 = (–*b* – d)*/*2*a *şeklinde hesaplanır.*

PROBLEM ÇÖZME ve ALGORİTMALAR *(Devam)*
Akış Diyagramı *(Devam)*

- **Psedo Kod**

1. Başla

2. a, b, c katsayılarını girin

3. d =sqrt ($b \times b - 4 \times a \times c$) hesapla

4. x1 = (–b + d)/(2 x a) hesapla

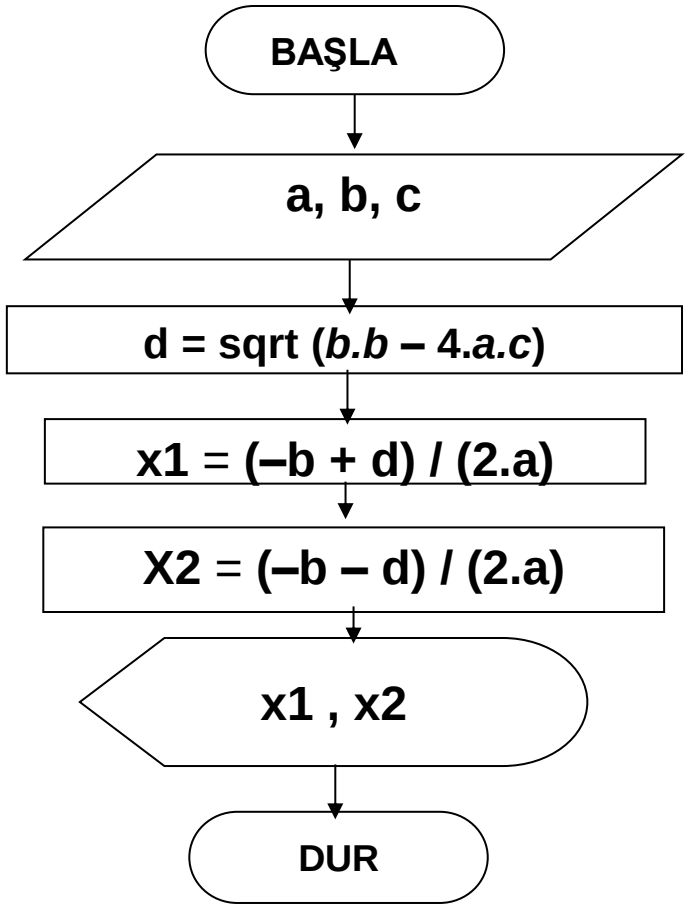
5. x2 = (–b – d)/(2 x a) hesapla

6. x1 ve **x2** yi yaz

7. Son

Not: *sqrt, Karekök anlamındadır.*

- **Akış Diyagramı**



ALGORİTMA VE PROGRAMLAMA (Bölüm 2)

BİLGİSAYARDA İŞLEMLER

- Bilgisayar programları ile gerçekleştirilen işlemlere genel olarak bir göz atalım:

- Bu işlemler temelde **3 Grupta** toplanabilir:

  1. **Matematiksel İşlemler**
  2. **Karşılaştırma(Karar) İşlemleri**
  3. **Mantıksal(Lojik) İşlemler**

Bilgisayarda İşlemler *(Devam)*
Matematiksel İşlemler *(Devam)*

1. **Matematiksel İşlemler:**

- **Temel Aritmetik İşlemler;**

  (Toplama,çıkarma,çarpma,bölme)

- **Matematiksel Fonksiyonlar;**

  (Üstel,logaritmik,trigonometrik,hiperbolik vb.)

Bilgisayarda İşlemler *(Devam)*
Matematiksel İşlemler *(Devam)*

İşlem	Matematikte	Bilgisayarda
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

Bilgisayarda İşlemler *(Devam)*
Matematiksel İşlemler *(Devam)*

Öncelik Sırası	***İşlem***	***Bilgisayar dili***
1	***Parantezler***	***((…….))***
2	***Üs alma***	***a^b***
3	***Çarpma ve bölme***	***a*b ve a/b***
4	***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.*

•*Mesela ;* ***Y=A*B/C*** *işleminde;*
Önce ***A*B işlemi*** *yapılacak, ardından bulunan* ***sonuç C ye bölünecektir****.*

Bilgisayarda İşlemler *(Devam)*
Matematiksel İşlemler *(Devam)*

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)

Bilgisayarda İşlemler *(Devam)*
(Karar) Karşılaştırma İşlemleri *(Devam)*

2. Karar (Karşılaştırma)İşlemleri

Karşılaştırma Operatörleri	
Operator	**Tanımı**
>	Daha büyük
<	Daha küçük
=	Eşit
≥	Daha büyük veya eşit
≤	Daha küçük veya eşit
≠	Eşit değil

Bilgisayarda İşlemler *(Devam)*
Karar (Karşılaştırma)İşlemleri *(Devam)*

- **A>B** bir **karşılaştırma** (mukayese) ifadesidir.
- Test etmek istediğimiz bir "**şartı**" tanımlar
- Bu şartın **sağlanıp sağlanmama** durumuna göre **farklı işlemler** yapılabilir.
- **A>B ise A nın değerini, aksi halde B değerini yazan psedo kod** (parçası) yazalım:

Bilgisayarda İşlemler *(Devam)*
Karar (Karşılaştırma)İşlemleri *(Devam)*

- Bu durumda **Psedo kod** (parçası) şöyle olacaktır:

- **Eğer A>B doğru ise**

  A ‘nın değerini yaz.

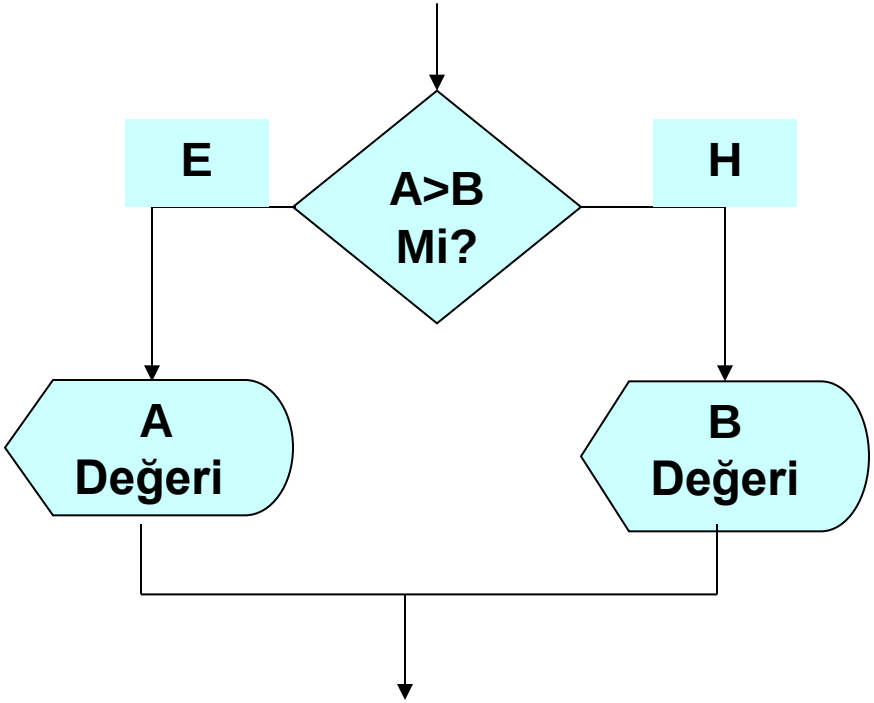
- **Değil ise** *(Aksi halde)*

  B ‘nin değerini yaz.

Not: ***A=B*** *durumu,* ***A<B*** *ile birlikte* ***Değil ise*** *durumuna* ***dahil*** *olacaktır.*

Bilgisayarda İşlemler *(Devam)*
Karar (Karşılaştırma)İşlemleri *(Devam)*

- Bu Karar yapısını **Akış Diyagramı** (parçası) olarak gösterelim;



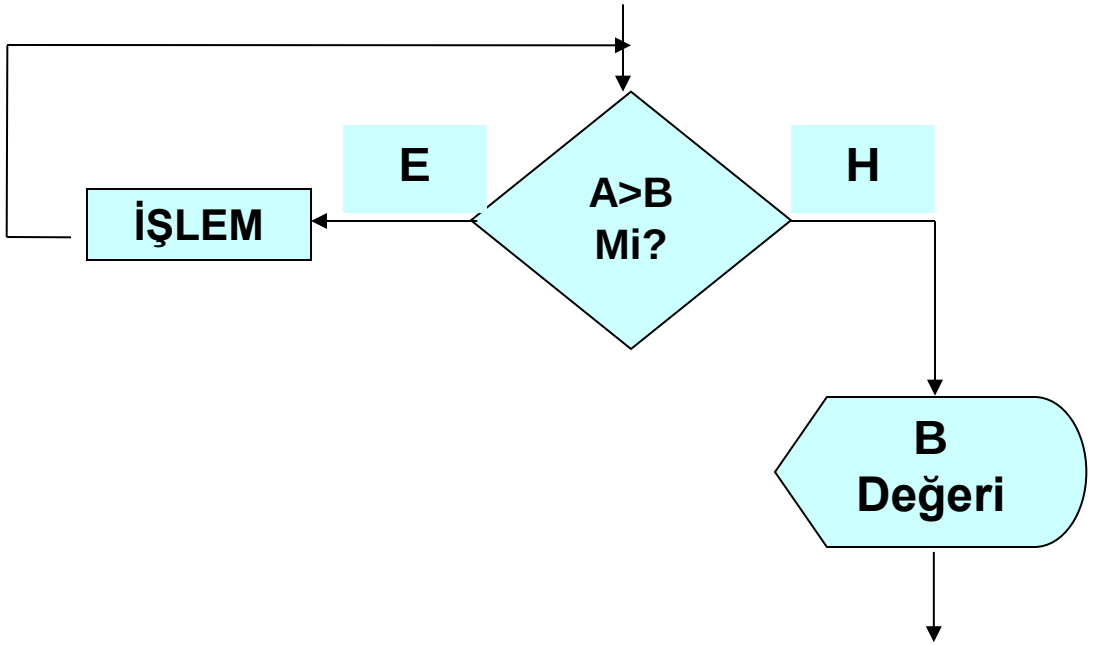
Bilgisayarda İşlemler *(Devam)*
Karar (Karşılaştırma)İşlemleri *(Devam)*

- Bir başka yapıya **örnek** olarak;
  - Eğer **A>B doğru ise** "**İŞLEM**" i yaparak **tekrar test** eden…
  - **A>B** doğru değil ise (aksi halde) **B** **'**nin değerini **yazan** bir karar yapısına ait bir **Akış Diyagramı** (parçasını) gösterelim.

Not: *Programın tamamına ait (Başla'dan Son'a kadar) A. Diyag. gösterilmezse «parçası» ifadesi kullanılır.*

Bilgisayarda İşlemler *(Devam)*
Karar (Karşılaştırma)İşlemleri *(Devam)*

Akış Diyagramı (Parçası)

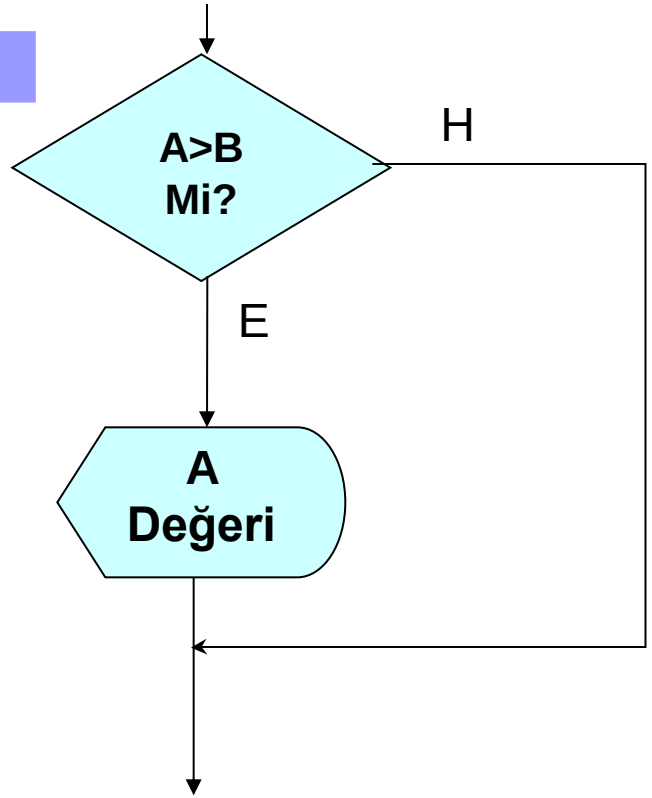


Bilgisayarda İşlemler *(Devam)*
Karar (Karşılaştırma)İşlemleri *(Devam)*

- Bir başka yapıya **örnek** olarak;
  - Eğer **A>B doğru** ise **A değerini yazdıktan sonra** Devam eden,
  - **A>B doğru değil** ise *(aksi halde)* **A değerini yazmadan** Devam eden Karar yapısına ait bir **Akış Diyagramı** (Parçasını) çizelim.

Bilgisayarda İşlemler *(Devam)*
Karar (Karşılaştırma)İşlemleri *(Devam)*

Akış Diyagramı (Parçası)



Bilgisayarda İşlemler *(Devam)*
Karar (Karşılaştırma)İşlemleri *(Devam)*

- Daha farklı Karar Yapıları bulunmakla birlikte bunları ilerdeki uygulamalarımıza

  bırakarak birkaç örneği (tam program için);

  - Hem **psedo…**
  - hem de **Akış Diyagramı** olarak inceleyelim:

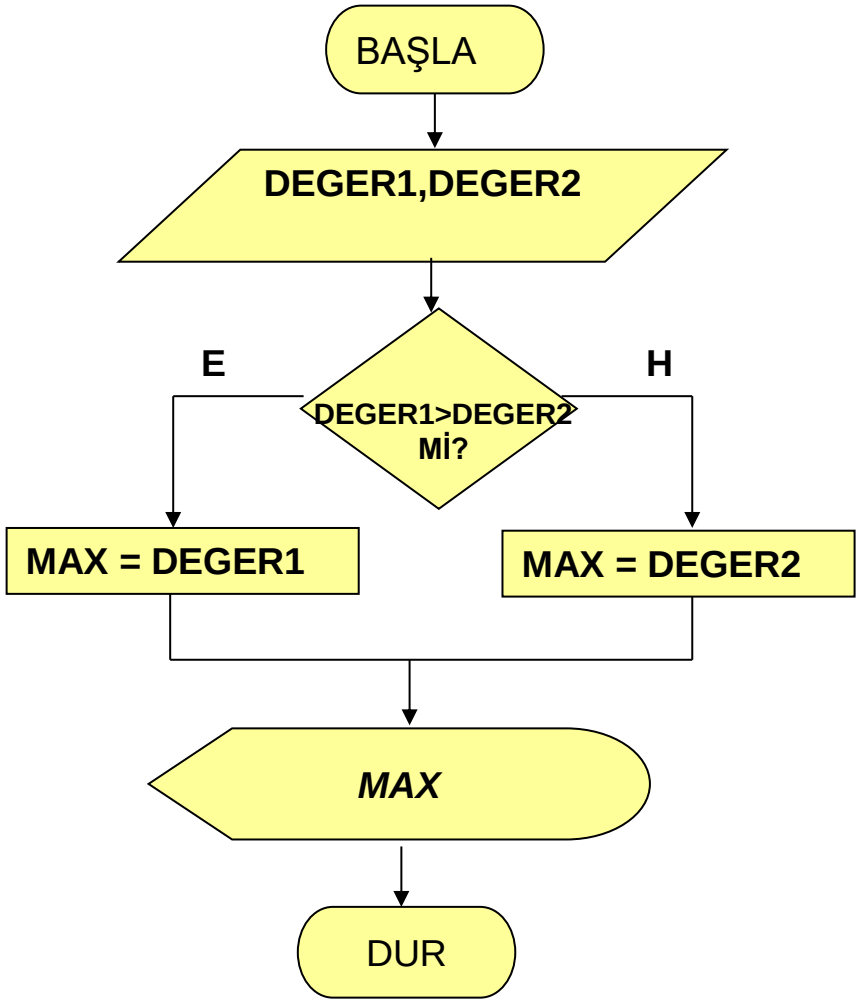
Bilgisayarda İşlemler *(Devam)*
Karar (Karşılaştırma)İşlemleri *(Devam)*

- **Örnek 6 :**Girilen iki değeri mukayese ettikten sonra **büyük olanı yazan** bir **psedo (sözde) kod** ve **akış diyagramı** veriniz.

1. Başla
2. DEGER1 ve **DEGER2** yi Girin
3. Eğer (**DEGER1** > **DEGER2**) ise
MAX'a **DEGER1 '**i aktar
4. Değilse *(Aksi halde)*
MAX'a **DEGER2'**yi aktar
5. MAX değerini yaz
6. Son

Bilgisayarda İşlemler *(Devam)*
Karar (Karşılaştırma)İşlemleri *(Devam)*

- **Akış Diyagramı**



Bilgisayarda İşlemler *(Devam)*
Karar (Karşılaştırma)İşlemleri *(Devam)*

- **Örnek 7 :** 3 sayıyı (**N1,N2,N3**) **okuyan** ve **en büyük değeri** yazan bir program için bir **psedo kod** yazarak **Akış Diyagramı** çiziniz.

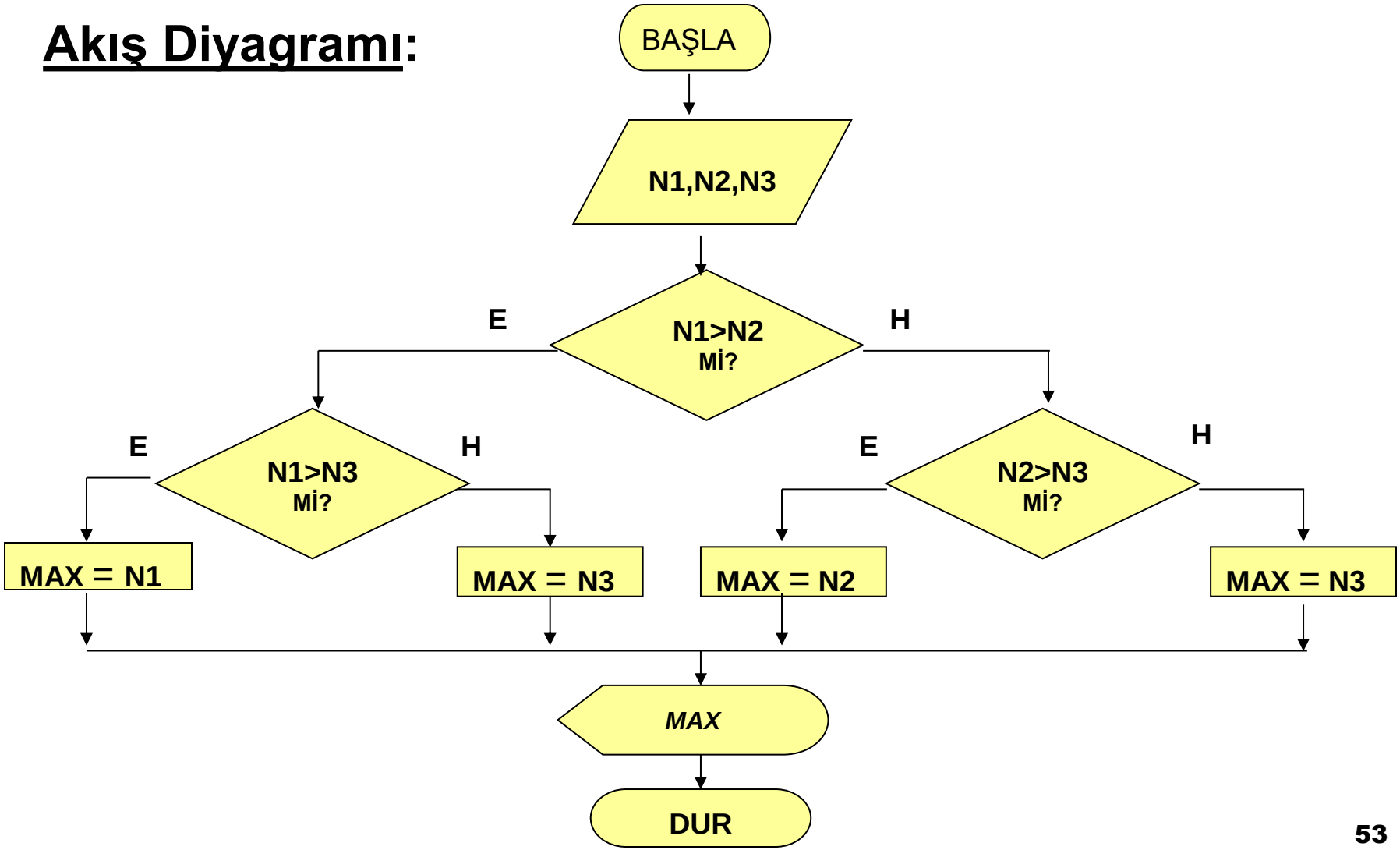
Bilgisayarda İşlemler *(Devam)*
Karar (Karşılaştırma)İşlemleri *(Devam)*

1. Başla
2. N1, N2, N3 değerlerini girin
3. Eğer *(*N1>N2) ise
Eğer (N1>N3) ise
MAX'a N1'i aktar *[N1>N2 ve N1>N3] demektir.*
Değilse
MAX'a N3'ü aktar *[N1>N2 ve N3>N1] demektir.*
4. **Değilse;**
Eğer (N2>N3) ise
MAX'a N2'yi aktar *[N2>N1 ve N2>N3] demektir. .*
Değilse
MAX'a N3'ü aktar *[N2>N1 ve N3>N2] demektir.*

5. MAX *değerini yaz.*
6. Son

Bilgisayarda İşlemler *(Devam)*
Karar (Karşılaştırma)İşlemleri *(Devam)*

Akış Diyagramı:



3. Mantıksal İşlemler

Mantıksal işlem	*Matematiksel sembol*	*Komut/Operatör*
VE	**.**	*AND*
VEYA	**+**	*OR*
DEĞİL	**‘**	*NOT*

NOT1:

- *“VE,VEYA,DEĞİL “ operatörleri hem matematiksel işlemlerde hem de karar ifadelerinde kullanılırlar.*

NOT2:

- *Bütün şartların sağlatılması isteniyorsa şartlar arasına **VE (AND)***
- *Herhangi birinin sağlatılması isteniyorsa şartlar arasına **VEYA (OR)***
- *Şartı sağlamayanlar isteniyorsa **DEĞİL (NOT)** mantıksal operatörü kullanılır.*

Bilgisayarda İşlemler *(Devam)*
Mantıksal İşlemler *(Devam)*

- **Örnek 8:** Bir sınıfta **Bilgisayar dersinden 65 in üzerinde** not alıp**, Türk Dili veya Yabancı Dil derslerinin herhangi birinden 65 in üzerinde** not alanların isimleri yazan bir Psedo Kod ‘un yazılması istensin.

Açıklama:

- *Burada **2 şart** vardır:*
- *Bilgisayar dersinden **65 in üzerinde not almış olmak temel şarttır**.*
- *Diğer iki dersin notlarının **herhangi birinin 65 in üzerinde** olması*

Bilgisayarda İşlemler *(Devam)*
Mantıksal İşlemler *(Devam)*

- Bu durumda **Psedo Kod** yazılırsa:

1. Başla
2. Bilg. Not, TDili Not , YDil Not değerlerini giriniz.
3. Eğer Bilg. Not>65 **VE (**TDili Not>65 **VEYA** YDil Not>65**)** ise
İsmi Yaz
4. Son

- ***Not :*** *Burada 3. satırdaki komut için* ***VE , VEYA operatörleri*** *kullanılmıştır.*

Bilgisayarda İşlemler *(Devam)*
Mantıksal İşlemler *(Devam)*

- **Örnek 9** : Bir işyerinde çalışan **işçiler arasından** yalnızca **yaşı 23 üzerinde olup Maaş olarak Asgari ücret alanların** isimleri yazılmak istenebilir.
- Bu durumda bir **Psedo Kod** yazarsak :

Bilgisayarda İşlemler *(Devam)*
Mantıksal İşlemler *(Devam)*

1. Başla

2. Yaş, Maaş, Asgari ücret ‘ i girin

3. Eğer Yaş>23 **VE** Maaş=Asgari ücret ise
İsmi Yaz

4. Dur

Not: İsmi Yaz *komutu* *1. ve 2. şartın her ikisi de sağlanıyorsa* *çalışır.*

ALGORİTMALARDA SIKÇA KULLANILAN BAZI TERİMLER

- **Tanımlayıcı**
- **Değişken**
- **Sabit**
- **Aktarma**
- **Sayaç**
- **Döngü**
- **Ardışık Toplama**
- **Ardışık Çarpma**
- **İçiçe Döngüler**

Algoritmalarda Sıkça Kullanılan Terimler *(Devam)*

- **Tanımlayıcı:**

- Programdaki ;
  - **Değişkenleri,**
  - **Sabitleri,**
  - **Kayıt alanlarını,**
  - **Özel bilgi tiplerini** vb. adlandırmak için kullanılan **kelimeler**dir.

- **Tanımlayıcılar**, yerini tuttukları ifadelere **çağrışım yapacak şekilde** seçilmelidir.
  (İsim ve Telefon için: ***adtelef*** *gibi)*

Algoritmalarda Sıkça Kullanılan Terimler *(Devam)*

- Tanımlayıcılarda ;
  - İngiliz alfabesindeki A-Z veya a-z arası 26 harf ile 0-9 arası rakamlar kullanılabilir.
  - Tanımlayıcı isimleri **harfle veya alt çizgiyle** başlayabilir.
  - Sembollerden sadece alt çizgi (_) kullanılabilir.
  - Tanımlayıcı ismi, **rakamla başlayamaz** veya **sadece rakamlardan** oluşamaz.

Algoritmalarda Sıkça Kullanılan Terimler *(Devam)*

- **Değişken:**

- Genelde **programın her çalıştırılmasında, farklı değerler** alırlar.
- **Değişken isimlendirilmeleri** de yukarıda sayılan **tanımlayıcı kurallarına uygun** şekilde yapılmalıdır.

- Meselâ ;
  - 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** şeklinde verilebilir.

Algoritmalarda Sıkça Kullanılan Terimler *(Devam)*

- **Sabit:**

- Programdaki **değeri değişmeyen** ifadelere "**sabit**" denir.
- Yukarda açıklanan "**İsimlendirme kuralları"na uygun** olarak oluşturulan **sabitlere**…
  - **Sayısal** veriler doğrudan… *(Örn:* **A=7***)*
  - **Alfasayısal veriler** ise **tek** yada **çift tırnak** içinde aktarılır. *(Örn:* ***A="puan25"*** *veya* ***A='puan25')***

Not: *Alfasayısal (alfanümerik), Verinin* ***hem harf hem de rakamlardan*** *oluştuğunu belirtmek için kullanılır.*

Algoritmalarda Sıkça Kullanılan Terimler *(Devam)*

- **<u>Örnek 10 :</u> Öğrenci adını içeren** bir mesajı **Psedo Kod** ile yazalım;

1. Başla
2. Bir isim giriniz (**A**)
3. "Derse Hoş geldin" mesajı (**B**) yaz
4. **B** ve **A** yı Yaz
5. Dur

Not: *Burada **(A) nın isim bilgisi taşıdığı yani** öğrenciye göre değişeceği için "**Değişken**" olduğuna…*
***(B) nin de değişmeyen** bir ifade yani "**Sabit**" olduğuna dikkat edelim.*

Algoritmalarda Sıkça Kullanılan Terimler *(Devam)*

Yukarıdaki algoritma (psedo kod), dışarıdan girilen bir **A değişkeni ile B sabitini birleştirip** aşağıdaki iki örnekteki gibi ekrana yazar.

A değişkeni	***B sabiti***	***Ekranda yazılan***
Onur	Derse Hoş geldin	**Derse Hoş geldin Onur**
Emre	Derse Hoş geldin	**Derse Hoş geldin Emre**

Algoritmalarda Sıkça Kullanılan Terimler *(Devam)*

- **Aktarma:**

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

değişken=ifade *(Örn:* **puan=a*65** *)*

- **=** **sembolü**, **aktarma operatörüdür** ve **sağdaki ifadenin/işlemin sonucunu soldaki değişkene** aktarır.

Algoritmalarda Sıkça Kullanılan Terimler *(Devam)*

- Değişken yazan soldaki kısım **herhangi bir değişken** ismidir.

  *(Yukarıdaki örnekte **puan** kelimesidir.)*

- İfade yazan sağdaki kısımda ise **matematiksel, mantıksal veya alfasayısal** ifade olabilir.

  (Yukarıdaki örnekte bir çarpma ifadesi)

- Bu durumda **değişkenin eğer varsa bir önceki değeri silinir**.

Algoritmalarda Sıkça Kullanılan Terimler *(Devam)*

- **Sayaç :**

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

  Örnek: **say=say+1** *(**say** adlı değişken için)*

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

  Not: *Bu ifadeden önce **say=7** ise sonra **say=8** olur.*

Algoritmalarda Sıkça Kullanılan Terimler *(Devam)*

- Sayacın **genel formülü;**

  Sayaç değişkeni = Sayaç değişkeni + adım

- Örnek: X=X+3

  *(**X** değişkenini **adım kadar** (3) **arttırır**)*

- Örnek: S=S-5

  *(**S** değişkenini **adım kadar** (5) **azaltır**)*

Algoritmalarda Sıkça Kullanılan Terimler *(Devam)*

- **Örnek 11:** Aşağıda **1 - 5 arası sayılar yazdırılmaktadır**.
  *(Sayac için **S** değişkeni kullanılmıştır)*

1. Başla
2. S=0 *(**S** değişkenine **0** ilk değerini ver.)*
3. Eğer S>4 ise
git 7. *(**S, 4** den büyükse **7.** satıra git)*
4. Değilse
S=S+1 *(**S, 4** den büyük değilse **S** yi 1 arttır)*
5. S 'yi Yaz *(**S** değerini yaz**)***
6. Git 3. *(**3.** satıra git)*
7. Dur

Algoritmalarda Sıkça Kullanılan Terimler *(Devam)*

Önceki S Değeri	Yeni S Değeri	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 Sıkça Kullanılan Terimler *(Devam)*

- **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.

Algoritmalarda Sıkça Kullanılan Terimler *(Devam)*

- **<u>Örnek 12:</u> 1 ile 10 arası tek sayıların toplamını** hesaplayan ve yazan bir **Psedo Kod** yazınız.

Not: ***T****, toplamı;*

J*, tek sayıları (1,3,5,7,9) temsil eden değişkenlerdir.*

Algoritmalarda Sıkça Kullanılan Terimler *(Devam)*

1. Başla

2. T=0 *(Önce **T** toplam değişkenini sıfırla)*

3. J=1 *(İlk tek sayıyı **J** değişkenine aktar)*

4. Eğer J>10 ise

git 8. *(**J>10** olduysa 8. adıma git)*

5. Değilse

T=T+J *(Aksi halde **T** değişkenini **J** kadar arttır)*

6. J=J+2 *(**J** yi adım yani 2 kadar arttır)*

7. Git 4. *(4. adıma git)*

8. T yi yaz *(Toplamı yani **T** yi yazdır**)***

9. Dur

Not: 7. ile 4. satırlar arasında Döngü vardır.

Algoritmalarda Sıkça Kullanılan Terimler *(Devam)*

Önceki J Değeri	Önceki T Değeri	Yeni T Değeri	Yeni J Değeri
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 Sıkça Kullanılan Terimler *(Devam)*

- **Ardışık Toplama:**

- Programlarda, **aynı değerin üzerine yeni değerler eklemek** için kullanılır.

- **Örnek11:** Klavyeden girilen **5 sayının ortalamasını** bulan programın algoritmasını **Psedo Kod ile** yazalım.

Not: T: *(Ardışık)* ***Toplamı****;*

S: Girilen sayının sırasını *göstersin.*

Algoritmalarda Sıkça Kullanılan Terimler *(Devam)*

1. Başla

2. T=0 *(Toplamı yani **T** yi sıfırla)*

3. S=0 *(Girilen sayının sırasını sayan **S** yi sıfırla)*

4. Eğer S>4 ise

git 8 *(**S>4** yani **S=5** olduysa 8. adıma git)*

5. Değilse S=S+1 *(Aksi halde **S** yi 1 arttır)*

Sayıyı (A) girin *(Klavyeden **A** sayısının girilmesini iste)*

6. T=T+A *(Girilen sayıyı **T** ile topla)*

7. Git 4. *(4. adıma git)*

8. Ortalama=T/5 *(Ortalama'yı hesapla)*

9. Ortalama'yı yaz *(Ortalama'yı yaz)*

10. Dur

Algoritmalarda Sıkça Kullanılan Terimler *(Devam)*

- **Ardışık Çarpma :**

- Ardışık çarpma işleminde; aynı değer (çarpım), yeni değerlerle çarpılarak eskisinin üzerine aktarılmaktadır.

- **Örnek 14:** Klavyeden girilen **N sayısının faktöriyelini hesaplayan** programın algoritmasını **Psedo Kod** ile yazınız.

Not: *5! (5 faktöriyel); 1.2.3.4.5 sayılarının çarpımıdır.*
Faktöriyel değeri **Faktor** *adlı değişkende (**ardışık**) **çarpma** ile hesaplanır.*

Algoritmalarda Sıkça Kullanılan Terimler *(Devam)*

1. Başla
2. N sayısını girin *(Hangi sayının faktöriyelini hesaplasın)*
3. Faktor=1 *(İstenen Faktöriyel değeri 1 den başlasın)*
4. S=0
5. Eğer S>N-1 ise
git 9. *(**S, N-1 den büyükse** işlem bitecek)*
6. S=S+1 *(Çarpılacak sayıları elde etmek için arttır)*
7. Faktor=Faktor*S *(Önceki **Faktor** değerini yeni **S** ile çarp)*
8. Git 5. *(5. adıma git)*
9. Faktor'u yaz *(**Faktor** değerini yaz)*
10. Dur

Algoritmalarda Sıkça Kullanılan Terimler *(Devam)*

- **Döngü Yapısı :**

- Bu yapı kullanılırken;
  - **Döngü sayacı…**
  - **Şart bilgisi** ve …
  - **Sayacın artım bilgisi** gibi değerler verilmelidir.

***Not: Döngü sayacı kullanılmıyorsa** sadece döngüye devam edebilmek için gerekli olan **şart bilgisi** verilmelidir.*

Algoritmalarda Sıkça Kullanılan Terimler *(Devam)*

- Genel olarak pek çok **programlama dilinin döngü deyimleri :**

1) **While**
2) **Do-while**
3) **For** gibi yapılar üzerine kurulmuştur.

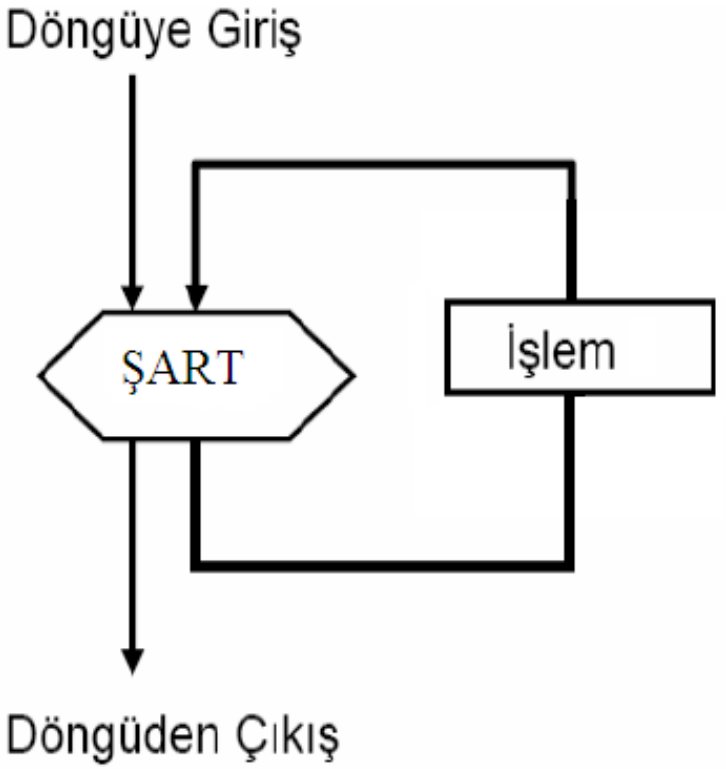
Not: *Farklı dillerde bu yapılara farklı alternatifler olsa da* ***döngülerin çalışma mantığı*** *genel olarak* ***benzerdir****.*

Algoritmalarda Sıkça Kullanılan Terimler *(Devam)*
Döngü Yapısı *(Devam)*

- **1. Durum (While):**
  - Şart daha **çevrim içerisine girmeden** sınanır.
  - Şart **sağlanmıyorsa İşlem hiç yapılmaz…**
  - Ve **döngü içerisindeki İşlem'de yapılması gerekenler atlanır.**

Algoritmalarda Sıkça Kullanılan Terimler *(Devam)*
Döngü Yapısı (Devam)

- **Döngüye Girildikten sonra Şart'ın sağlanıp sağlanmadığına bakılır.**
- **Sağlanmıyorsa Döngüden çıkılır.**
- **Şart sağlandığı sürece (defa) İşlem yapıldıktan sonra döngüden çıkılır.**



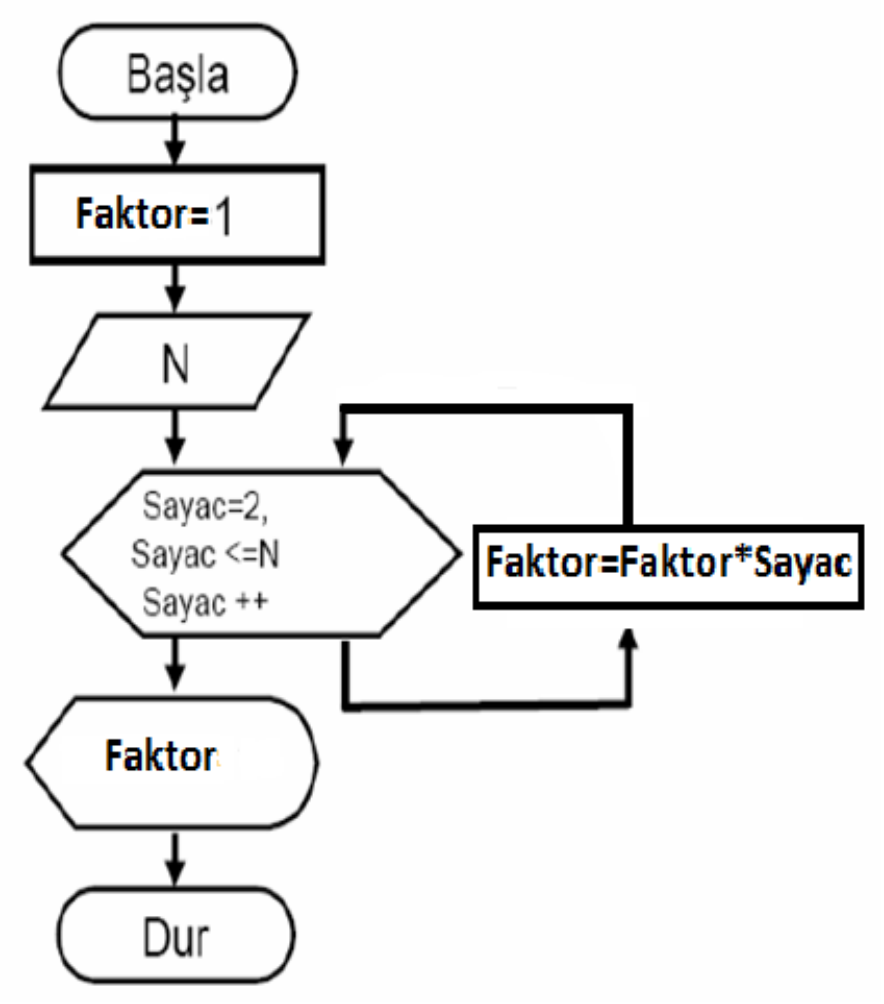
Algoritmalarda Sıkça Kullanılan Terimler *(Devam)*

- **<u>Örnek 15:</u> Bu tür bir döngü için daha önce algoritması yazılan Örnek 12 'e ait bir akış diyagramı çizelim.**

- Önce Örnek 12' i hatırlayalım:

  - Klavyeden girilen **N sayısının faktöriyelini hesaplayan** programın algoritmasını **Psedo Kod** ile yazınız.

Algoritmalarda Sıkça Kullanılan Terimler *(Devam)*



Not:

• ***Faktor=1*** *ile Faktöriyele ilk değeri veriliyor.*
•Böylece 1 sayısının Faktöriyeli baştan yazılmış oluyor.

• ***Sayac=2*** *ile Sayac'a* ***başlangıç olarak 2 değeri*** *veriliyor.*

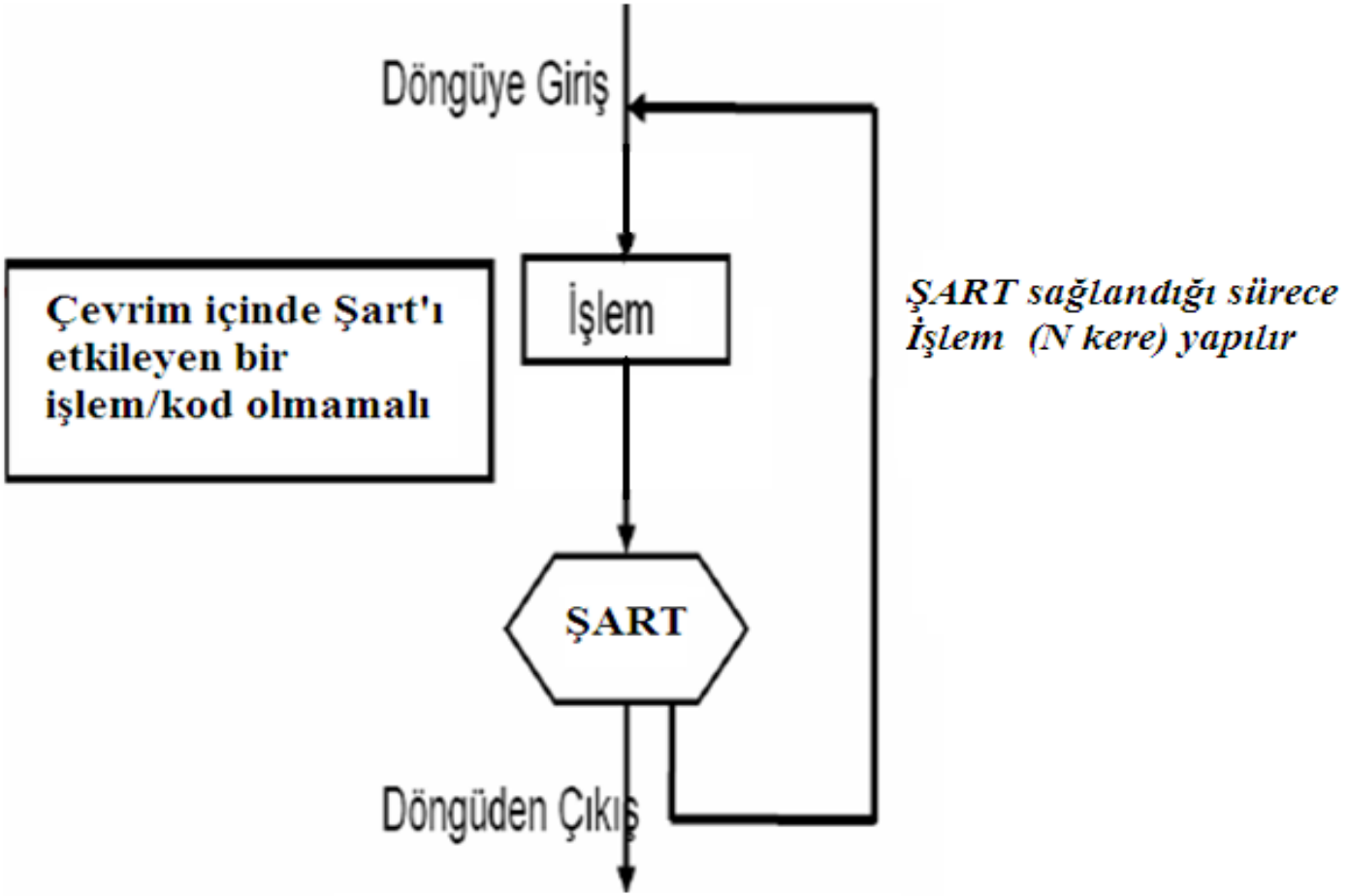
*•**Sayac değeri 2 den başlayarak** girilen*
Örn: *N=10 değerine eşit olana kadar (2,3,4….9,10)* ***Faktor=Faktor*Sayac*** *işlemi tekrar yapılıyor.*

• ***Sayac++*** *ile her seferinde* ***Sayac 1*** *arttırılıyor.*

Algoritmalarda Sıkça Kullanılan Terimler *(Devam)*
Döngü Yapısı (Devam)

- **2. Durum (Do-While):**
  - Bu durumda **çevrim en az bir defa** gerçekleşir. Çünkü **şart sınaması döngü sonunda** yapılmaktadır.
  - Eğer **şart sağlanmıyorsa** bir sonraki çevrime **geçilmeden döngüden çıkılır**.
  - Çevrimin **Devam edebilmesi için** her döngü sonunda yapılan **şart testinin** olumlu sonuçlanması gerekir.

Algoritmalarda Sıkça Kullanılan Terimler *(Devam)*
Döngü Yapısı (Devam)



Algoritmalarda Sıkça Kullanılan Terimler *(Devam)*
Döngü Yapısı (Devam)

- **3. Durum (For):**

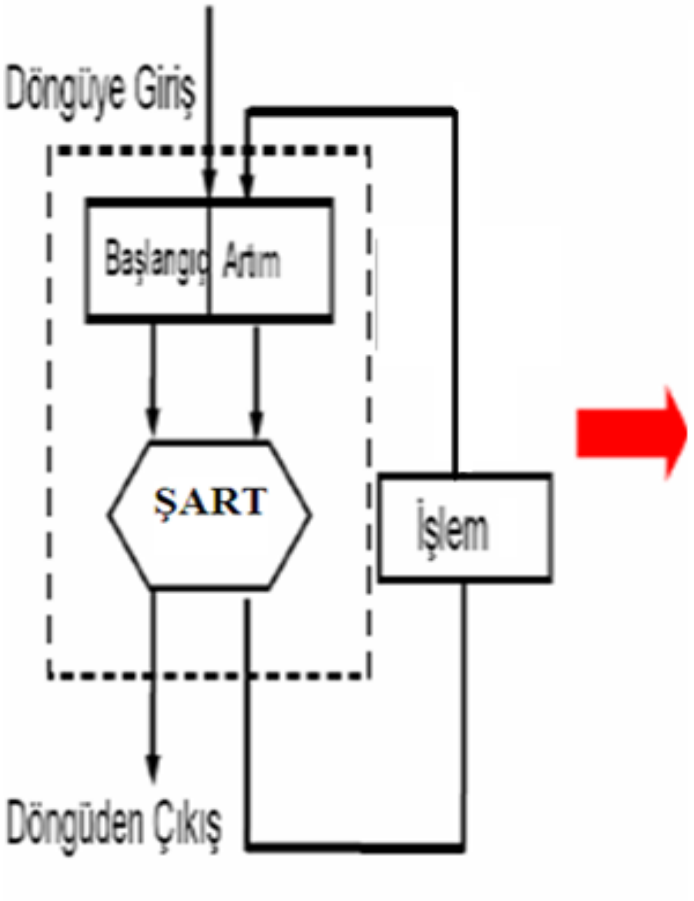
- Diğer deyimlerden farklı olarak, **döngü sayacı doğrudan şart parametreleri olarak başta verilir**.

- Döngüye girmeden önce **sayaç değişkenine başlangıç ve artım değeri atanmakta** ve daha **sonra şarta** bakılmaktadır.

- Şart sağlandığı sürece belirtilen **«İşlem» yapıldıktan sonra** her seferinde **sayaç değişkeni arttırılmaktadır**.

- **Döngü sayısına ulaşınca (şart sağlanamayınca) döngü sona erer.**

Algoritmalarda Sıkça Kullanılan Terimler *(Devam)*



Notlar:

**** Önce Başlangıç değeri verilir...***

**** ŞART içinde son değere ulaşılıp ulaşılmadığı kontrol edilir...***

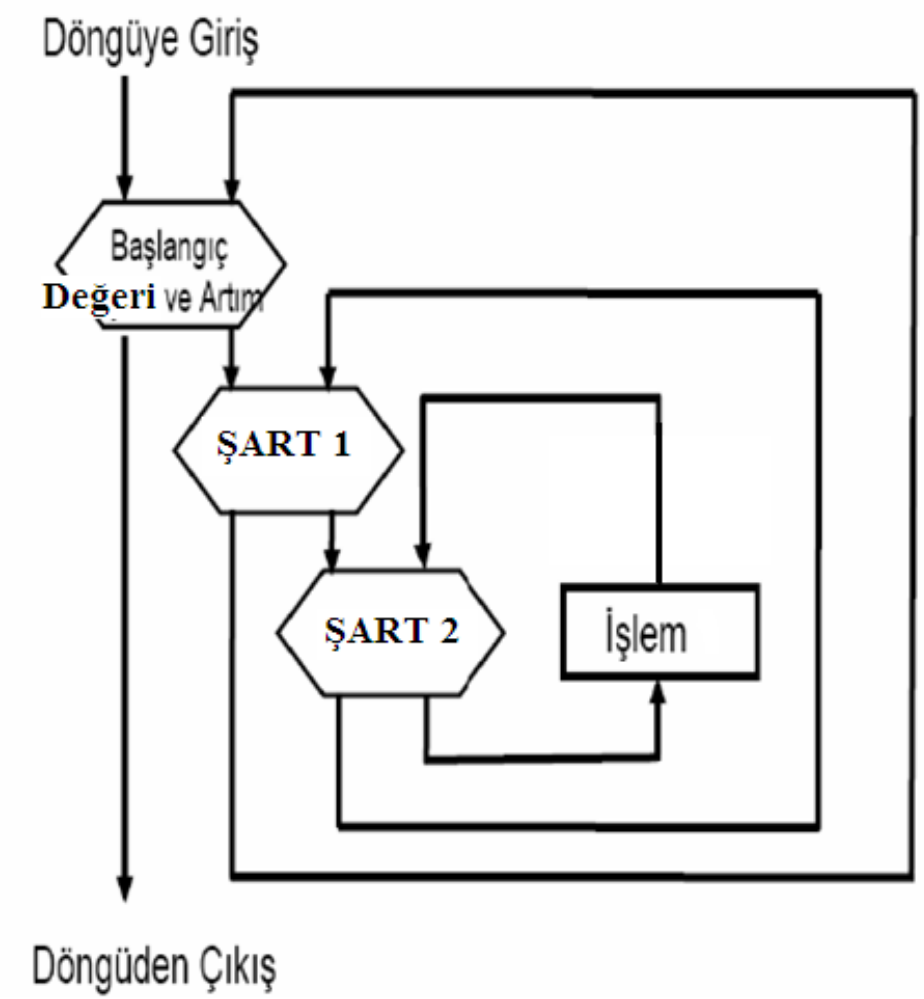
**** Her seferinde Artım sayısı kadar kadar Başlangıç değeri arttırılır...***

**** Döngü sayısı kadar İşlem tekrar yapılmış olur.***

Algoritmalarda Sıkça Kullanılan Terimler *(Devam)*
İçiçe Döngüler

- **İçiçe Döngüler:**
- Programlamada yaygın kullanılan bir döngüdür.
- 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 kere tekrarlanır**.
- İç ve dış döngülerin çarpımı kadar “**İşlem**” yapılmış olur.

Algoritmalarda Sıkça Kullanılan Terimler *(Devam)*



Notlar:
**** Başlangıç Değerinden sonra ŞART 1 kontrol edilir...***
****Sağlanıyorsa ŞART 2 kontrol edilir...***
**** ŞART 2 sağlandığı sürece İşlem yapılır...***
**** ŞART 2 sağlanmadığında tekrar ŞART 1 kontrol edilir...***
**** ŞART 1 sağlandığında tekrar ŞART 2 kontrol edilerek aynı şekilde İşlem yapılır...***
**** Örn: ŞART 1 5 defa ; ŞART 2 10 defa yapılacak şekilde belirlenmişse İşlem 5*10=50 kere yapılmış olur.***