PROGRAMLAMA DİLİNDE ALGORİTMA

Algoritma Nedir?

Bir sorunu çözebilmek için gerekli olan sıralı mantıksal adımların tümüne denir. Doğal dille yazılabileceği için fazlaca formal değildir. Bir algoritma için aşağıdaki ifadelerin mutlaka doğrulanması gereklidir.

- Her adım son derece belirleyici olmalıdır. Hiç bir şey şansa bağlı değildir.
- Belirli bir sayıda adım sonunda algoritma sonlanmalıdır.
- Algoritmalar karşılaşılabilecek tüm ihtimalleri ele alabilecek kadar genel olmalıdır.

Akış Çizgesi Nedir?

Bir algoritmanın daha görsel gösterimidir. Çizgiler, Dörtgen, daire vb. geometrik şekillerle algoritmanın gösterilmesini sağlar. Doğal dille yazılmadığı için daha formal olduğu düşünülebilir.

Algoritmalar

I.S. 9.yy da Iranli Musaoglu Horzumlu Mehmet

(Alharezmi adini araplar takmıştır.) problemlerin çözümü için genel kurallar oluşturdu. Algoritma Alharezmi'nin Latince okunuşu.

I.S. 9.yy da Iranli Musaoglu Horzumlu Mehmet

(Alharezmi adini araplar takmistir) problemlerin çözümü için genel kurallar oluşturdu. Algoritma Alharezmi'nin Latince okunuşu.

Her algoritma aşağıdaki kriterleri sağlamalıdır.

1. Girdi: Sıfır veya daha fazla değer dışarıdan verilmeli.

2. Çıktı: En azından bir değer üretilmeli.

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

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

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

Not: Bir program için 4. özellik geçerli değil. işletim sistemleri gibi program sonsuza dek çalışırlar

Bir bilgisayar programı aslında sıra düzensel olarak tanımlanmış bir dizi komuttan başka bir şey değildir. Bu açıdan bizim yazmaya çalışacağımız programda bir dizi komut yani eylem topluluğudur. Her programda bu eylemler yazıldıkları sırada gerçekleştirilir veya çalıştırılırlar. Aslında bizim günlük hayattaki yaşantı tarzımız dahi düzenli olarak bir takım işlemlerin sıra ile yapılması şeklindedir. Yani bir iş yapabilmek için bir takım alt iş veya olayları peş peşe gerçekleştiririz.

Algoritmanın tanımını daha önce vermiştik burada bu tanımı tekrar etmek faydalı olacaktır. Bir sorunu çözebilmek için gerekli olan sıralı mantıksal adımların tümüne algoritma denir. Bir algoritmadan beklenen bir takım özellikler olduğunu da yine daha önceki tanımlar bölümünde bahsetmiştik. Biz şimdi mümkün olduğu kadar bu tanım ve özelliklerden yola çıkarak örneklerle bir kaç algoritma vermeye çalışacağız.

Öncelikle bir ev hanımının pasta yapmak istediğini varsayalım. Bu pastanın yapılabilmesi için gerekli bir takım işlemler ve alt adımlar bellidir. bir ev hanımıda sıra ile bu adımları uygulayarak bu pastayı yapar. Şöyle ki:

1. Pastanın yapımı için gerekli malzemeleri hazırla
2. Yağı bir kaba koy
3. Şekeri aynı kaba yağın üzerine koy
4. Yağ ve şekeri çırp
5. Karışımın üzerine yumurtayı kır
6. Tekrar çırp
7. Kıvama geldimi diye kontrol et
8. a. Kıvamlı ise 9. adıma devam et
   b. Değilse 6. adıma dön.
9. Karışıma un koy
10. Karışıma vanilya, kabartma tozu vb. koy
11. Karışımı Kıvama gelinceye kadar çırp
12. Pastayı Kek kalıbına koy
13. Yeteri kadar ısınan fırına pastayı koy
14. Pişimi diye kontrol et
15. a. Pişmiş ise 16. adıma devam et
    b. Değilse 14. adıma dön
16. Keki fırından çıkart
17. Fırını kapat
18. Kekin soğumasını bekle
19. Keki servis edebilirsin.

Bu algoritma günlük hayattan bir örnek. Gerçekte biz her işimizi algoritmik olarak yaparız ancak bunu farkına varmayız. Yukarıdaki algoritmayı inceleyecek olursak bir kekin yapılması için gerekli tüm adımlar sıra ile yer almış durumda. Gerçi algoritma anlatacağımız konuların daha iyi anlaşılabilmesi için biraz farklı ele alınmıştır ama gerçek bir Pasta yapım aşamasını içerir. Bu algoritma ve diğer tüm algoritmalar için bilmemiz gereken bazı konular bulunmaktadır:

- Her adım son derece belirleyici olmalıdır. Hiç bir şey şansa bağlı olmamalıdır.
- Belirli bir sayıda adım sonunda algoritma sonlanmalıdır.
- Algoritmalar karşılaşılabilecek tüm ihtimalleri ele alabilecek kadar genel olmalıdır.
- Algoritmada algoritmanın genel işleyişini etkileyebilecek hiç bir belirsizlik olmamalıdır.

(Bu örnekte öyle bir belirsizlik var. Bir fırının yeteri kadar ısına bilmesi hangi koşula bağlıdır, bu fırın ne zaman açılmış olmalıdır ve kaç dereceye ayarlanmış olmalıdır. gibi...) Algoritmada bazı adımlar yer değiştirebilir . Ancak bir çok adımın kesinlikle yer değiştiremeyeceğini bilmeliyiz. Yanlış sıradaki adımlar algoritmanın yanlış çalılaşmasına neden olacaktır. (9 ve 10. adım değiştirilebilir. 2-3. adımlar yer değiştirebilir.) Ancak 13-16. adımlar kesinlikle yer değiştiremezler.

Peki Bilgisayarda çözülecek bir sorunu nasıl algoritma ile ifade ederiz? Bunun için öncelikle bir sorun tanımlayalım. Başlangıç ta basit olması için şöyle bir problem üzerinde düşünelim: Bilgisayara verilecek iki sayıyı toplayıp sonucu ekrana yazacak bir program için algoritma geliştirmek isteyelim. Sorun son derece basit ancak sistem tasarımının net yapılabilmesi için sorun hakkında anlaşılamayan tüm belirsiz noktalar açıklığa kavuşturulmalıdır. Örneğin sayışlar bilgisayara nereden verilecek, Klavye, Dosya veya belki başka bir ortam. Bu ve buna benzer soru ve tereddütleriniz varsa sorun sahibine bunları sormalı ve sistem analizi yapmalısınız.

Sonra bulacağımız çözümü algoritma haline dönüştürebiliriz.

1. BAŞLA
2. A sayısını oku
3. B sayısını oku
4. TOPLAM=A + B işlemini yap
5. TOPLAM değerini ekrana yaz
6. SON

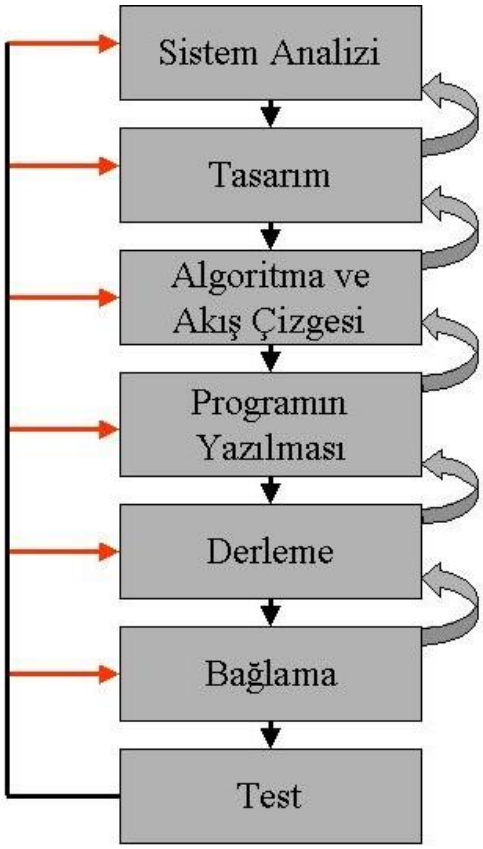
Biraz daha bir sorun şöyle olsun: Klavyeden girilecek iki sayıdan büyük olanından küçük olanını çıkarıp sonucu ekrana yazacak program için bir algoritma geliştiriniz.

1. BAŞLA
2. A sayısını oku
3. B sayısını oku
4. Eğer A büyüktür B SONUC=A-B
Değilse SONUC=B-A
5. SONUC değerini ekrana yaz
6. SON

Bu algoritmalar oldukça basit algoritmalar olup algoritma kavramının yerleşmesini sağlayan örneklerdir .Bütün bunların dışında algoritmaların yazılım geliştirme konusunda da önemli görevi vardır .Aşağıdaki sıralamada bu önem sanırım daha iyi ortaya çıkacaktır.

Yazılım Geliştirme

Yazılım Geliştirilirken Bir Programcı ve Yazılım Gurubunun takip edeceği adımlar şu şekildedir.



Bu çizgeden anlaşılacağı gibi adımlardan birinde bir sorunla karşılaşılırsa bu sorunu çözebilmek için bir önceki adıma geri dönmek gerekecektir. Bu geri dönüş bazen bir kaç adım olabilir.

Sistem Analizi : Sorunun çözülebilmesi için tamamen anlaşılmasını sağlayan çalışmalardır.

Tasarım : İsteklerle ilgili olarak belirlenen bir takım çözümlerin tanımlanmasıdır.

Programlama Stili :

Algoritma : Çözümün adımlarla ifade edilmesidir.

Akış Çizgesi : Algoritmanın şekillerle ifade edilmesidir.

Programlama Dili Seçimi : Çözümün netleşmesinden sonra yapılacak işlemleri kolay bir şekilde bilgisayar ortamına aktaracak dilin seçilmesidir. Önemli olan bu dilin özelliklerinin programcı tarafından iyi bilinmesidir.

Programın Yazılması : Seçilen Programlama dilinin kuralları kullanılarak program yazılmaya başlanır. bu amaçla çoğunlukla sade bir metin editörü kullanılır. Bazı durumlarda Syntax highlighting denilen bir özelliğe sahip olan daha akıllı editörler de kullanılabilir. Bazen de editör ile Programlama dilinin derleyicisinin, bağlayıcısının hatta hata ayıklayıcısının iç içe bulunduğu IDE (Integrated Development Environment) denilen türde derleyiciler kullanılır.

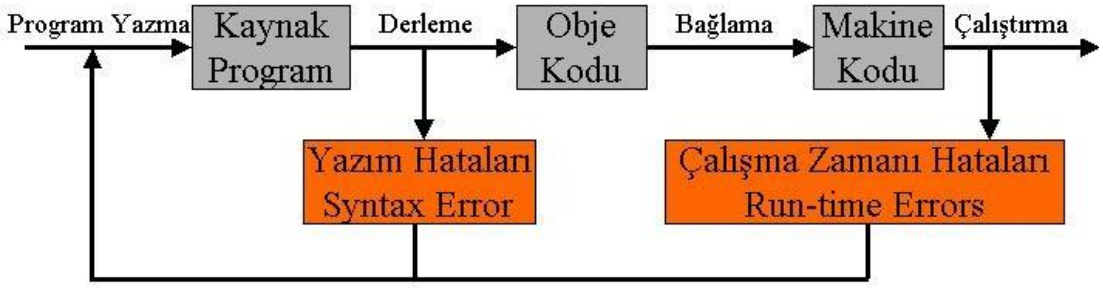
Derleme : Programlama Dili ile yazılmış programın yazım hatalarının olup olmadığının kontrol edilmesini ve ara kod olarak Obje kodun üretilmesini sağlama adımıdır.

Bağlama : Derlenmiş ara kod diğer kütüphane ve parça programlarla birleştirilerek Makine dilinde programın oluşturulması adımıdır. Ancak bazı IDE ortamlarda ve derleyicilerde Derleme ve Bağlama bir bütündür ve beraberce halledilirler. Programcının ayrıca bir bağlama işlemi yapması gerekmez.

Çalıştırma : Oluşturulan Makine dili Programının çalıştırılması adımıdır. Yukarıdaki adımların hepsi yolunda gittiyse program sorunsuz olarak çalışabilmelidir.

Test : Programın Mantıksal olarak test edilmesini sağlar ve içerik olarak her ihtimal için doğru sonuçlar üretip üretmediğini kontrol etmenizi sağlar.

Yaşam Döngüsünün Sağlanması : Yukarıdaki Akış Çizgesi dikkat edilirse aslında bir döngüdür. Hatta test aşamasında sorun çıkmazsa bile Sorunun tanımında yani ihtiyaçlarda bazı değişiklikler olursa adımlar baştan aşağı tekrar incelenmek zorunda kalınır. Bu çizgiye bir Yazılımın Yaşam Döngüsü de denilebilir. Bu çizimde Yazılımın Bakım süreci göz önüne alınmamıştır.



Örnek 1. 1'den 100'e kadar olan sayıların toplamını veren algoritma.

1. Toplam T, sayılar da i diye çağırılsın.
2. Başlangıçta T'nin değeri 0 ve i'nin değeri 1 olsun.
3. i'nin değerini T'ye ekle.
4. i'nin değerini 1 arttır.
5. Eğer i'nin değeri 100'den büyük değil ise 3. adıma git.
6. T'nin değerini yaz.

Algoritmaların yazım dili değişik olabilir. Günlük konuşma diline yakın bir dil olabileceği gibi simgelere dayalı da olabilir. Akis seması eskiden beri kullanıla gelen bir yapıdır. Algoritmayı yazarken farklı anlamlar taşıyan değişik şekildeki kutulardan yararlanılır. Yine ayni amaç için kullanılan programlama diline yakın bir (sözde kod = pseudo code) dil , bu kendimize özgü de olabilir, kullanılabilir.

Ayni Algoritmayı aşağıdaki gibi yazabiliriz.

1. Başla
2. sayac=0
3. sayac=sayac+1
4. toplam=toplam+sayac
5. eğer sayac<100 ise 3. Adıma git
6. toplamı yaz
7. bitir

Örnek 2. ax2+bx+c=0 tipi bir denklemin köklerini veren algoritma.

Girdi : a, b ve c katsayıları Çıktı : denklemin kökleri

1. başla
2. a, b ve c katsayılarını al.
3. d=b^2-4*a*c
4. eğer d<0 ise
    4.1. "kök yok yaz"
    4.2. bitir
5. eğer d=0 ise
    5.1. "tek kök" yaz
    5.2. x=-b/2*a
    5.3. x'i yaz
    5.4. bitir
6. eğer d>0 ise
    6.1. "çift kök" yaz
    6.2. x1=-b+math.sqrt(d)/2*a
    6.3. x1=-b-math.sqrt(d)/2*a
    6.4 . x1 ve x2' yi yaz
    6.5. bitir

Örnek 3. İki tamsayının çarpma işlemini sadece toplama işlemi kullanarak gerçekleyin.

Girdi : iki tamsayı

Çıktı : sayıların çarpımı

1. Başla
2. Sayac=0
3. A,B' yi gir
4. Sayac=sayac+1
5. Sonuc=sonuc+B
6. Eğer sayac<a ise 4. Adıma git
7. Sonucu yaz
8. Bitir

Örnek 4. Bir tamsayının faktoriyelini hesaplayınız.

Girdi : Bir tamsayı

Çıktı : sayının faktoriyel

İlgili formül: Faktoriyel(n)=1*2*...*n

1. Başla
2. Sonuc=1
3. Sayıyı gir (x)
4. Sonuc=sonuc*x
5. X=x-1
6. Eğer x>1 ise 4. Adıma git
7. Sonucu yaz
8. Bitir

Örnek 5. Dışarıdan girilen 50 sayı içerisindeki en büyük ve en küçük sayıyı bulan programın algoritması.

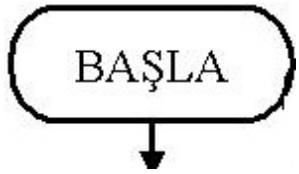
1. Başla
2. Sayac=0
3. Sayac=sayac+1
4. Sayı gir (x)
5. Eğer x=1 ise
5.1. EBS=x
5.2. EKS=x
6. Eğer x>EBS ise EBS=x
7. Eğer x<EKS ise EKS=x
8. Eğer sayac<50 ise 3. Adıma git
9. EBS ve EKS' yi yaz
10. Bitir

Örnek 6. Dışarıdan girilen 10 sayı içerisinden negatif, pozitif ve sıfırların sayısını bulan programın algoritması

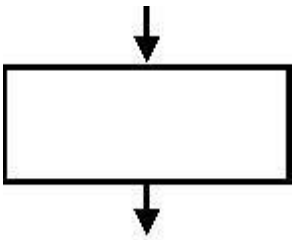
1. Başla
2. Sayac=0, p=0, n=0, s=0
3. Sayac=sayac+1
4. Sayı gir (x)
5. Eğer x<0 ise n=n+1
6. Eğer x=0 ise s=s+1
7. Eğer s>0 ise p=p+1
8. Eğer sayac<10 ise 3. Adıma git
9. N,s,p' yi yaz
10. Bitir

Akış Çizgeleri

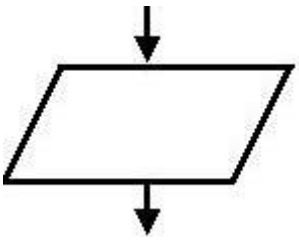
Bir algoritmanın şekillerle görsel gösterimidir. Dikkat edildiyse algoritma doğal dille yazıldığı için herkes tarafından anlaşılamayabilir ya da istenmesele başka anlamlar çıkarılabilir. Ancak akış çizgelerinde her bir şekil standart bir anlam taşıdığı için farklı yorumlanıp anlaşılamaması mümkün değildir. Bir algoritmanın ifade edilebilmesi için kullanılan şekiller ve anlamları şunlardır:



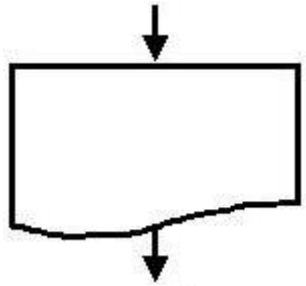
Bir algoritmanın başladığı konumu göstermektedir. Tek çıkışlı bir şekildir.



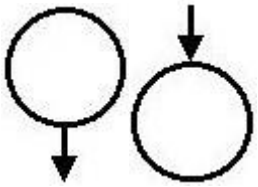
Bir algoritmada aritmetik işlem yapılmasını sağlayan şekildir. Bu dörtgen kutu içerisine yapılmak istenen işlem yazılır. Tek girişli ve tek çıkışlı bir şekildir.



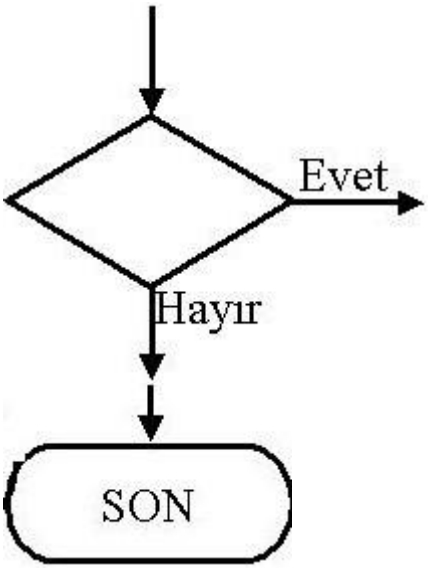
Giriş Çıkış komutunun kullanılacağı yeri belirler ve kutu içerisine hangi değişkeni ve OKU mamı YAZ mı yapılacağını belirtmeniz gerekir



Sonucun yazdırılması



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

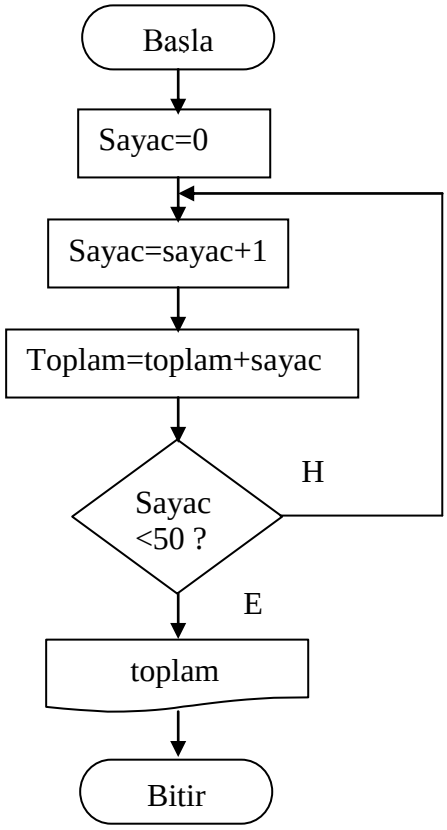


Bir algoritmada bir kararın verilmesini ve bu karara göre iki seçenekten birinin uygulanmasını sağlayan şekildir. burada eşkenar dörtgen içerisine kontrol edilecek mantıksal koşul yazılır. Program akışı sırasında koşulun doğru olması durumunda "Evet" yazılan kısma Yanlış olması durumunda "Hayır" yazılan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

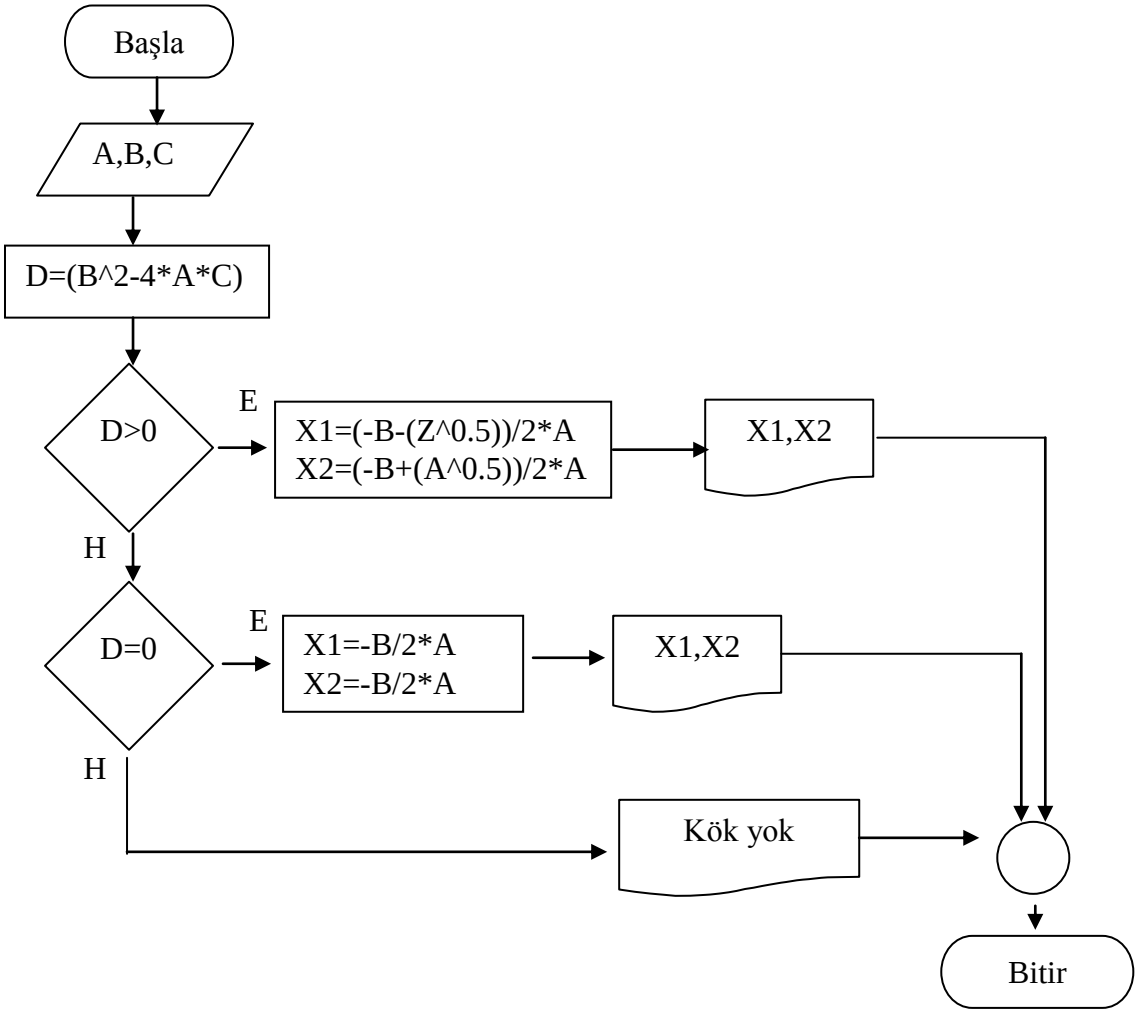
Programın bittiği yer ya da yerleri gösteren bir şekildir.

Bu şekiller kullanılarak algoritma ile oluşturulan çözümler akış çizgelerine çevrilir. Bu şekiller herkes tarafından anlaşılabilir ve doğru olarak yorumlanabilir bir özellik arz ederler.

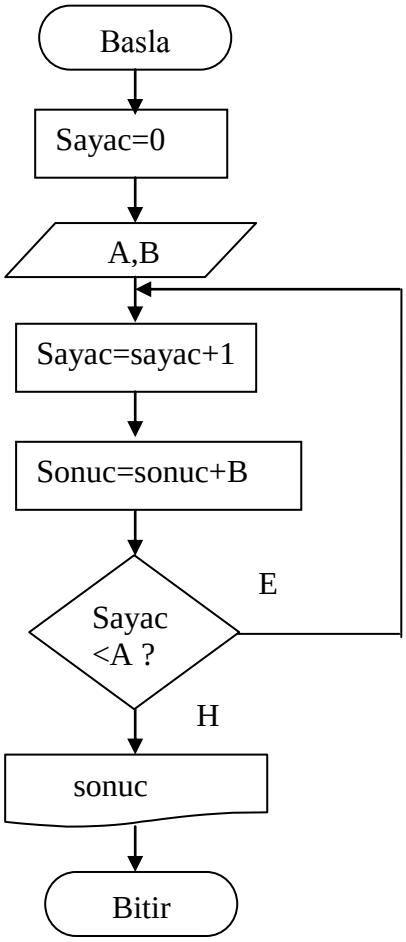
Örnek 1.



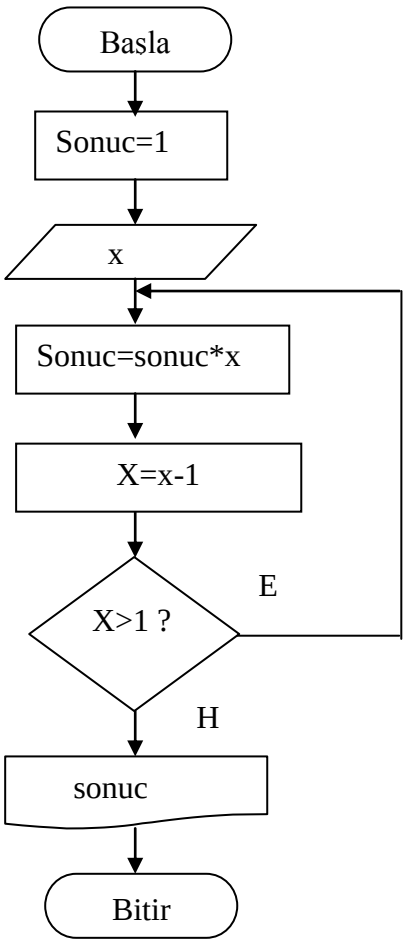
Örnek 2. Katsayıları dışarıdan girilen ikinci dereceden denklemin köklerini bulan program.



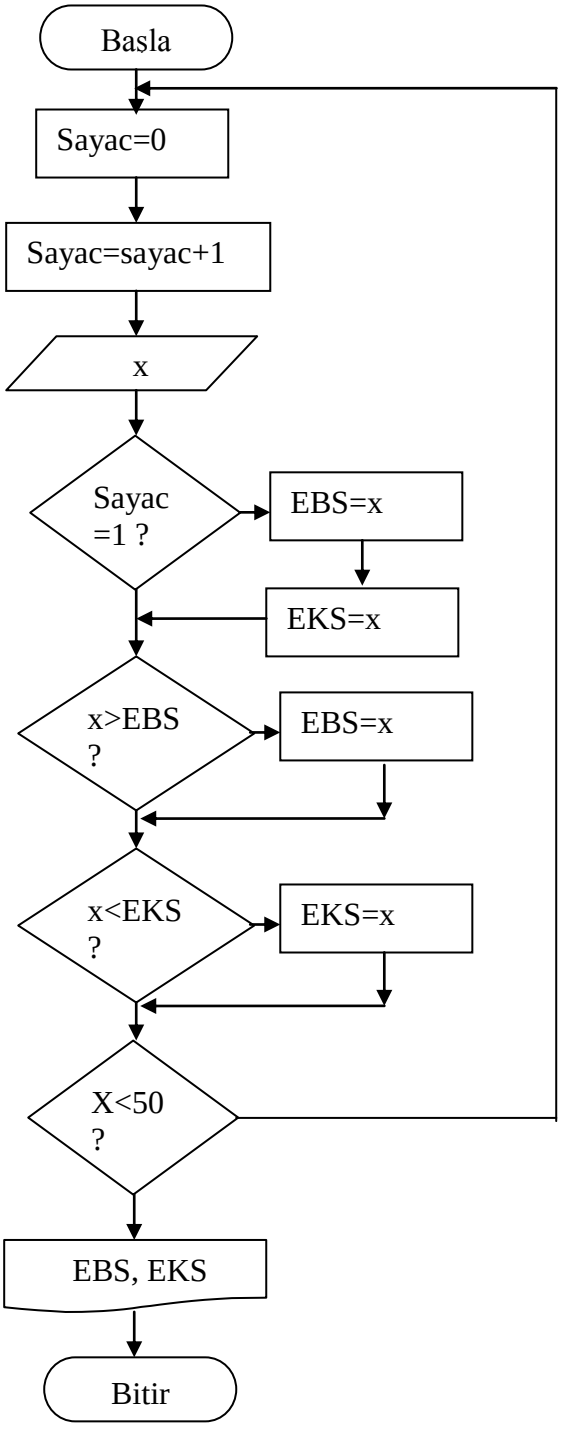
Örnek 3.


Basla
Sayac=0
A,B
Sayac=sayac+1
Sonuc=sonuc+B
Sayac
<A ?
E
H
sonuc
Bitir

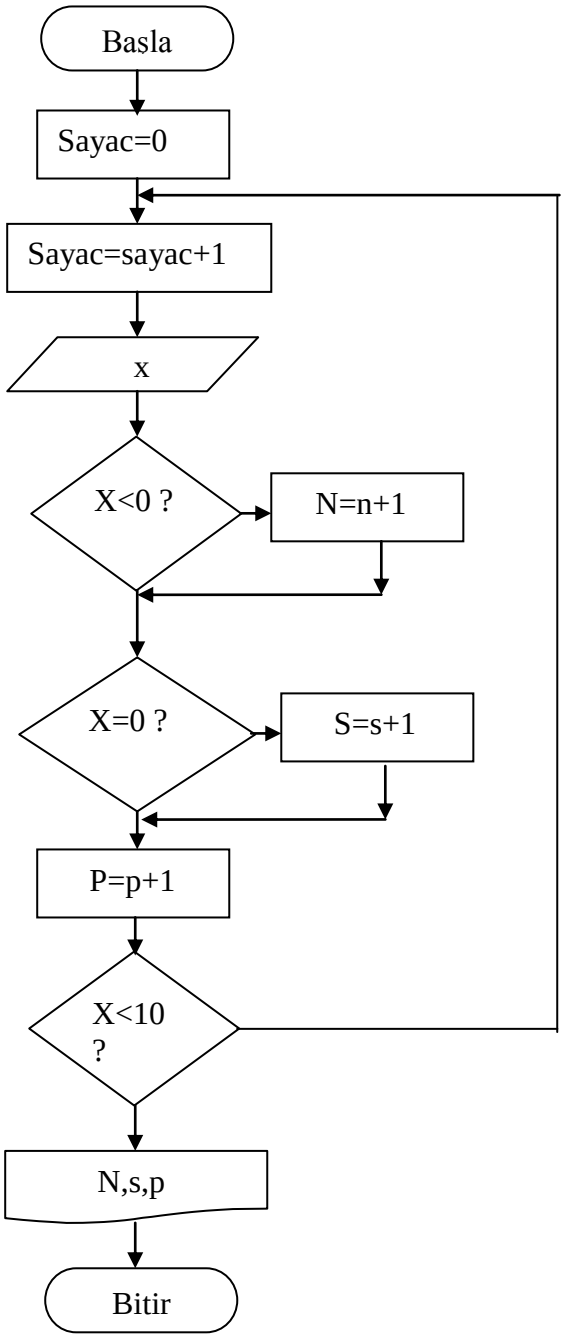
Örnek 4.



Örnek 5.


Basla
Sayac=0
Sayac=sayac+1
x
Sayac =1 ?
EBS=x
EKS=x
x>EBS ?
EBS=x
x<EKS ?
EKS=x
X<50 ?
EBS, EKS
Bitir

Örnek 6.



Örnek 1.

```
Sub Main()
        Dim sayac, toplam As Integer
a:      sayac = sayac + 1
        toplam = toplam + sayac
        If sayac < 50 Then
            GoTo a
        Else
            Console.Write("Toplam=" & toplam)
        End If
        Console.ReadLine()
 End Sub
```

Örnek 2.

```
Sub Main()
        Dim a, b, c, d As Integer
        Dim x, x1, x2 As Double
        Console.Write("a katsayısını gir")
        a = Console.ReadLine
        Console.Write("b katsayısını gir")
        b = Console.ReadLine
        Console.Write("c katsayısını gir")
        c = Console.ReadLine
        d = b ^ 2 - 4 * a * c
        If d < 0 Then
            Console.WriteLine("kök yok")
        ElseIf d = 0 Then
            Console.WriteLine("tek kök")
            x = -b / 2 * a
            Console.WriteLine("x=" & x)
        Else
            Console.WriteLine("çift kök")
            x1 = -b + Math.Sqrt(d) / 2 * a
            x2 = -b - Math.Sqrt(d) / 2 * a
            Console.WriteLine("x1=" & x1)
            Console.WriteLine("x2=" & x2)
        End If
        Console.ReadLine()
    End Sub
```

Örnek 3.

```
Sub Main()
        Dim sayac, sonuc, a, b As Integer
        Console.Write("a sayısını gir")
        a = Console.ReadLine
        Console.Write("b sayısını gir")
        b = Console.ReadLine
al:     sayac = sayac + 1
        sonuc = sonuc + b
        If sayac < a Then
            GoTo al
        Else
            Console.Write("sonuc=" & sonuc)
        End If
        Console.ReadLine()
    End Sub
```

Örnek 4.

```
Sub Main()
        Dim sonuc As Integer = 1
        Dim x As Integer

        Console.Write("bir sayı gir")
        x = Console.ReadLine
al:     sonuc = sonuc * x
        x = x - 1

        If x > 1 Then
            GoTo al
        Else
            Console.Write("sonuc=" & sonuc)
        End If
        Console.ReadLine()
    End Sub
```

Örnek 5.

```
Sub Main()
        Dim x, ebs, eks, sayac As Integer
al:     sayac = sayac + 1
        Console.Write("sayı gir=")
        x = Console.ReadLine
        If sayac = 1 Then
            ebs = x
            eks = x
        End If
        If x > ebs Then
            ebs = x
        End If
        If x < eks Then
            eks = x
        End If
        If x < 50 Then
            GoTo al
        Else
            Console.Write("ebs=" & ebs)
            Console.Write("eks=" & eks)
        End If
        Console.ReadLine()
    End Sub
```

Örnek 6.

```
Sub Main()
        Dim x, n, s, p, sayac As Integer
al:     sayac = sayac + 1
        Console.Write("sayı gir=")
        x = Console.ReadLine
        If x < 0 Then
            n = n + 1
        ElseIf x = 0 Then
            s = s + 1
        Else
            p = p + 1
        End If
        If x < 10 Then
            GoTo al
        Else
            Console.Write("n=" & n)
            Console.Write("s=" & s)
            Console.Write("p=" & p)
        End If
        Console.ReadLine()
    End Sub
```

LABARATUAR UYGULAMALARI

TEK VE ÇİFT SAYILARI BULMA

```
Sub Main()
        Dim sayi As Byte
        Console.Write("Sayı gir=")
        sayi = Console.ReadLine()

        If (sayi Mod 2 = 0) Then
            Console.WriteLine("çift")
        Else
            Console.WriteLine("tek")
        End If
        Console.ReadKey()
    End Sub
```

ÇARPMA İŞARETİ KULLANMADAN ÇARPMA

```
Sub Main()
        Dim sayac As Byte = 0
        Dim sayi1 As Byte = 0
        Dim sayi2 As Byte = 0
        Dim sonuc As Byte = 0
        Console.Write("1. sayıyı giriniz=")
        sayi1 = Console.ReadLine()
        Console.Write("2. sayıyı giriniz=")
        sayi2 = Console.ReadLine()
```

```
        sayac = 0
basa:   sayac = sayac + 1
        sonuc = sonuc + sayi1
        If sayac < sayi2 Then GoTo basa
        Console.SetCursorPosition(50, 12)
        Console.WriteLine(sayi1.ToString() + " x " +
sayi2.ToString() + " = " + sonuc.ToString())
        Console.ReadKey()
    End Sub
```

İKİNCİ DERECEDEN DENKLEM ÇÖZÜMÜ

```
Sub Main()
        Dim a, b, c, d As Integer
        Dim x, x1, x2 As Double
        Console.Write("A katsayısını gir=")
        a = Console.ReadLine
        Console.Write("B katsayısını gir=")
        b = Console.ReadLine
        Console.Write("C katsayısını gir=")
        c = Console.ReadLine
        d = b ^ 2 - 4 * a * c
        If d < 0 Then
            Console.WriteLine("Kök Yoktur")
        End If
        If d = 0 Then
            x = -b / 2 * a
            Console.WriteLine("X=" + x.ToString)
        End If
        If d > 0 Then
            x1 = -b + Math.Sqrt(d) / 2 * a
            x2 = -b - Math.Sqrt(d) / 2 * a
            Console.WriteLine("X1=" + x1.ToString)
            Console.WriteLine("X2=" + x2.ToString)
Console.ReadKey()
        End If
```

TEK SAYI VE ÇİFT SAYI BULMA

```
Sub Main()
        Dim sayi, sayac, tek, cift As Integer
        sayac = 0
ilk:    sayac = sayac + 1
        Console.Write("Sayı Gir=")
        sayi = Console.ReadLine
        If sayi Mod 2 = 0 Then
            tek = tek + 1
        Else
```

```
            cift = cift + 1
        End If
        If sayac < 10 Then GoTo ilk
        Console.WriteLine("Tek sayıların sayısı=" + tek.ToString)
        Console.WriteLine("Çift sayıların sayısı=" + cift.ToString)
        Console.ReadKey()
    End Sub
```

FAKTÖRİYEL HESAPLAMA

```
Sub Main()
        Dim sayi, sayac, fak As Integer
        sayac = 0
        fak = 1
        Console.Write("Sayı gir=")
        sayi = Console.ReadLine
ilk:    sayac = sayac + 1
        fak = fak * sayac
        If sayac < sayi Then GoTo ilk
        Console.WriteLine("Faktöryel=" + fak.ToString)
        Console.ReadKey()
    End Sub
```

SAYILARIN TOPLAMINI BULMA

```
Sub Main()
        Dim sayi, sayac, toplam As Integer
        sayac = 0
        toplam = 0
ilk:    sayac = sayac + 1
        Console.Write("Sayı gir=")
        sayi = Console.ReadLine
        toplam = toplam + sayi
        If sayac < 10 Then GoTo ilk
        Console.WriteLine("Toplam=" + toplam.ToString)
        Console.ReadKey()
    End Sub
```

EN BÜYÜK VE EN KÜÇÜK SAYIYI BULMA

```
Sub Main()
        Dim sayi, sayac, ebs, eks As Integer
        sayac = 0
ilk:    sayac = sayac + 1
        Console.Write("Sayı gir=")
        sayi = Console.ReadLine
```

```
        If sayac = 1 Then
            ebs = sayi
            eks = sayi
        End If
        If sayi < eks Then eks = sayi
        If sayi > ebs Then ebs = sayi
        If sayac < 10 Then GoTo ilk
        Console.WriteLine("En küçük sayı=" + eks.ToString)
        Console.WriteLine("En büyük sayı=" + ebs.ToString)
        Console.ReadKey()
    End Sub
```

ASAL SAYI BULMA

```
Sub Main()
        Dim sayi, sayac, kalan As Integer
        sayac = 0
        Console.Write("Sayı gir=")
        sayi = Console.ReadLine
ilk:    sayac = sayac + 1
        If sayi Mod sayac = 0 Then kalan = kalan + 1
        If sayac < sayi Then GoTo ilk
        If kalan = 2 Then
            Console.WriteLine("sayı asladır")
        Else
            Console.WriteLine("sayı asal değildir")
        End If
        Console.ReadKey()
```

MÜKEMMEL SAYI BULMA

```
Sub Main()
        Dim sayi, sayac, toplam As Integer
        sayac = 0
        Console.Write("Sayı gir=")
        sayi = Console.ReadLine
ilk:    sayac = sayac + 1
        If sayi Mod sayac = 0 Then toplam = toplam + sayac
        If sayac < sayi - 1 Then GoTo ilk
        If sayi = toplam Then
            Console.WriteLine("sayı mükemmeldir")
        Else
            Console.WriteLine("sayı mükemmel değildir")
        End If
        Console.ReadKey()
    End Sub
```