

ALGORİTMA VE PROGRAMLAMAYA GİRİŞ

Yrd.Doç.Dr. Kadriye ERGÜN
kergun@balikesir.edu.tr

Ders Bilgileri

- BIL1202 Algoritma ve Programlamaya Giriş, (2+1 Kredi)
- Ders ile ilgili duyurular
 - <http://kergun.baun.edu.tr/>
- Kaynaklar
 - Algoritma ve Programlamaya Giriş, Ebubekir Yaşar
 - Algoritma ve Programlama Mantığı, Burak Tungut
 - Algoritma, Kadir Çamoğlu
 - Visual Basic 6.0 Pro, İhsan Karagülle, Zeydin Pala
- Başarı Notu
 - Vize (%40)
 - Final (%60)
 - Kısa Sınavlar (Vize veya Final notuna %10 oranında eklenir)

Haftalar

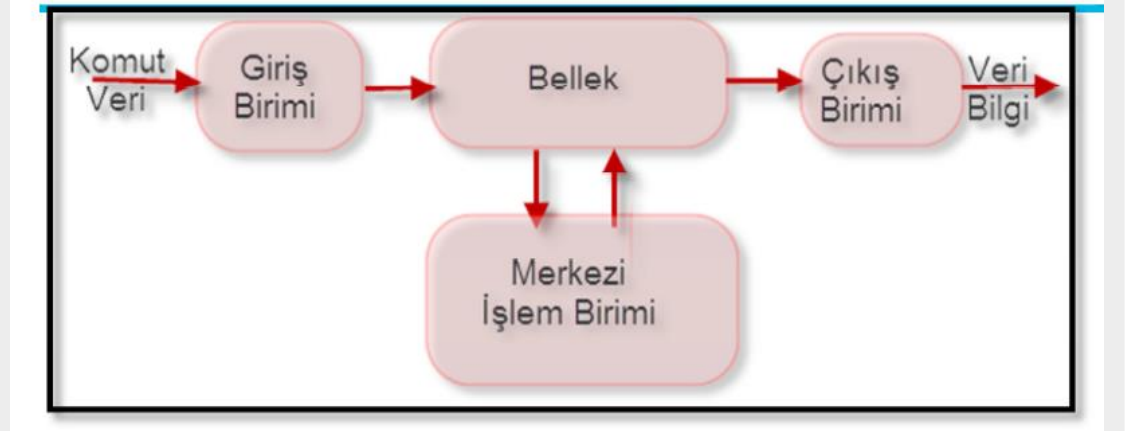
- Temel Kavramlar
- Program Geliştirme Süreci
- Algoritmalar
- Akış Diyagramları
- Karar Yapıları
- Döngüler
- Diziler
- Sıralama Algoritmaları(Bubble Sort)
- Visual Basic 6.0 Uygulaması
- Visual Basic 6.0 Uygulaması
- Visual Basic 6.0 Uygulaması
- Visual Basic 6.0 Uygulaması

İçerik

- Temel Kavramlar
- Program Geliştirme Süreci
- Algoritma Nedir?
 - Basit Algoritmalar
 - Mantıksal Algoritmalar
 - Döngüsel Algoritmalar
- Akış şeması
 - Basit Akış Diyagramları
 - Mantıksal Akış Diyagramları
 - Döngüsel Akış Diyagramları

Temel Kavramlar

- Bilgisayar, aritmetiksel ve mantıksal işlemlerden oluşan bir işi, önceden verilmiş programa göre yapıp sonuçlandıran elektronik bir araçtır.
- Bir bilgisayarın çalışabilmesi için üç temel birime ihtiyaç vardır.
 - Merkezi İşlem Birimi
 - Bellek Birimi
 - Giriş-Çıkış Birimi(I/O)



Temel Kavramlar

- **Merkezi İşlem Birimi:** Bilgisayardaki tüm karar verme ve kontrol işlemlerini gerçekleştirir. Matematiksel işlemleri gerçekleştirdiği gibi bilgisayarda hangi birimlerden giriş yapılacak hangi sırada çıkış yapılacak öncelikler nasıl olacak vb. işlemleri de gerçekleştirir.
- **Bellek Birimi:** Bilgisayarlar çalıştıkları süre boyunca giriş biriminden aldığı veya hesaplama sonucu elde ettiği verileri bellek üzerinde saklayarak işlemleri gerçekleştirirler.
- **Giriş Çıkış Birimleri:** Kullanıcıdan veya diğer aygıtlardan (fare, klavye, mikrofon, kamera, tarayıcı vb.) bilgisayara veri aktarmak için kullanılan birimlere Giriş Birimleri, bilgisayarda bulunan verileri kullanıcıları bilgilendirmek amacıyla veya diğer aygıtlara (ekran, yazıcı, tarayıcı, hoparlör, kulaklık vb.) göndermek amacıyla kullanılan birimlere de Çıkış Birimleri denir.

Bellek Ölçü Birimleri

- 1 Byte = 8 Bit
- 1 Kilobyte (KB) = 10^3 = 1.024 byte.
- 1 Megabyte (MB) = 10^6 = 1.048.576 byte.
- 1 Gigabyte (GB) = 10^9 = 1.073.741.824 byte.
- 1 Terabyte (TB) = 10^{12} = 1.099.511.627.776 byte.
- 1 Petabyte (PB) = 10^{15} byte.
- 1 Exabyte (EB) = 10^{18} byte.
- 1 Zettabyte (ZB) = 10^{21} byte.
- 1 Yottabyte (YB) = 10^{24} byte.

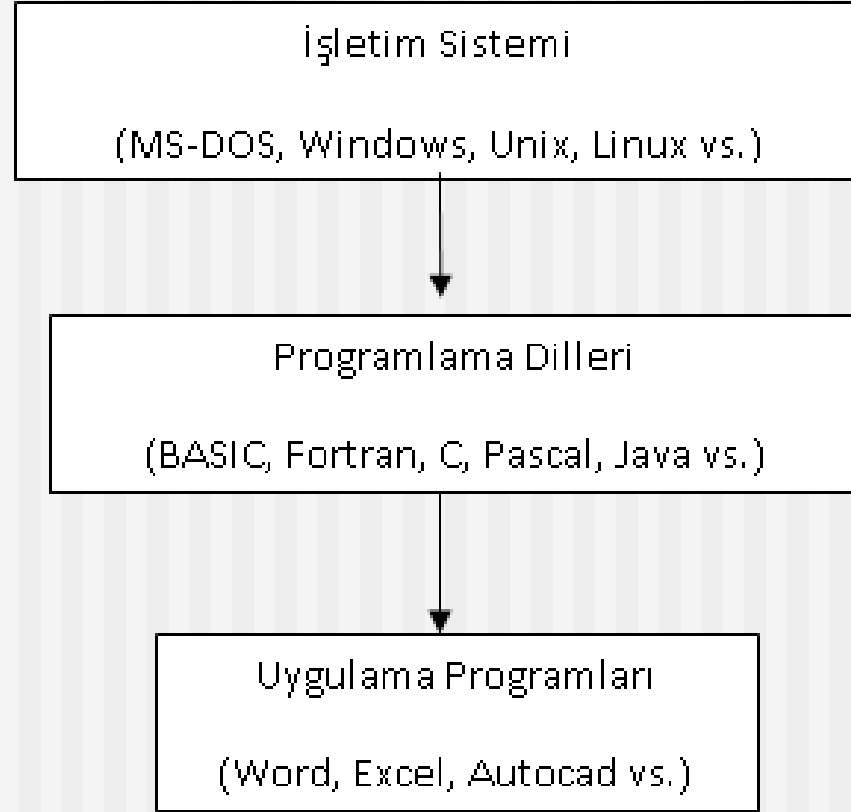
Yazılım-Donanım

- Bilgisayar sistemleri yazılım ve donanım olmak üzere iki kısımdan oluşmaktadır.
- **Donanım:** Bilgisayarda gözle görebildiğimiz fiziksel parçalar donanım olarak isimlendirilir. Donanımlar kullanım amaçlarına göre 4 kısımda incelenirler.
 - Merkezi İşlem Birimi
 - Bellek Birimi
 - Depolama Birimleri
 - Çevre Birimleri

Yazılım-Donanım

- **Yazılım:** Bilgisayarın çalışması için donanım dışında kalan kısma yazılım denilir. Genel olarak üç kısımda incelenebilir.
 - Sistem Yazılımları
 - Program Geliştirme Yazılımları
 - Uygulama Yazılımları
- Yazılım geliştirme sonucu ortaya çıkan ürüne program denir. Bir problemin bilgisayar tarafından çözülebilmesi için öncelikle algoritmasının oluşturulması gerekmektedir.

Yazılım Çeşitleri



Derleyici-Yorumlayıcı

- Yüksek seviyeli programlama dilinin anladığı komutları bilgisayarın anladığı komutlara çeviren yazılımlara ihtiyaç vardır. Bu yazılımlar derleyici ve yorumlayıcı olarak iki sınıfa ayrılır.
- **Derleyici(Compiler):** Yüksek seviyeli programlama dilinde yazılmış programın bütününün incelenerek eğer hata yoksa makine diline çevrilerek amaç program dosyasının üretildiği ve program sonuçlarının yine yüksek seviyeli program çıktıları olarak üretme işlemini gerçekleştiren birime Derleyici denir. Bu çevirme işlemine de Derleme denir. Eğer program kodları bir hata ile karşılaşırorsa amaç program dosyası oluşmaz. Amaç program dosyası bir kez oluşturulmuşsa yani makine diline çevrilmiş ise tekrar tekrar kullanılabilir.
- **Yorumlayıcı (Interpreter):** Yüksek seviyeli programlama dilinde yazılmış programı satır satır inceleyerek eğer hata varsa hata mesajı vererek işlemi durduran eğer hata yoksa satırdaki komutları makine diline çevirerek çalıştıran ve bir sonraki satıra geçen ve bu işlemi tüm program satırları için gerçekleştiren birime Yorumlayıcı denir. Bu işleme de Yorumlama denir.
- Bazı diller derlenirken (Turbo Pascal, C, vb.) bazı diller de yorumlanır(Qbasic, vb.). Bunun yanısıra Java gibi diller de derleyici yorumlayıcı kombinasyonunu kullanmaktadır.

Tanımlar

- **Algoritma** bir problemin mantıksal çözümünün adım adım nasıl gerçekleştirileceğinin sözlü ifadesidir.
- Algoritma ile oluşturulan çözümler sözel olarak ifade edildiğinden daha standart herkesin gördüğünde ortak olarak aynı sonucu çıkarabileceği hale getirmek için akış diyagramları kullanılır. **Akış diyagramları** sembollerden oluşmaktadır. Her sembolün belli bir işlevi vardır.
- Algoritması oluşturulmuş bir problemin bilgisayar ortamına aktarılmış haline **program** denir.
- Program, problemin çözümünde yapılması gereken işlemler bütünüdür **kod** karşılığıdır.
- Algoritmaların program haline getirilmesi için **programlama dilleri** kullanılır.
- Programlama dilleri kullanılarak **yazılımlar** geliştirilir.

Program Geliştirme Süreci

- Analiz
- Tasarım
- Gerçekleme
- Çalıştırma
- Hata Ayıklama

Algoritmalar

- Algoritma bir problemin mantıksal çözümünün adım adım nasıl gerçekleştirildiğinin sözle ifadesidir. Algoritma ilk defa Harezmi tarafından bir cebir kitabında kullanılmıştır.
- Algoritma, matematikte ve bilgisayar biliminde bir işi yapmak için tanımlanan bir başlangıç durumundan başlandığında, açıkça belirlenmiş bir son durumda sonlanan sonlu işlemler (adımlar) kümesidir.

Örnek

- Algoritma ile oluşturulacak çözümler sözel olarak ifade edilir. Örneğin sabah kalktığımızda kahvaltı yapılacağı zaman kahvaltı hazırlama algoritması oluşturulursa:
 - Yataktan kalk
 - Mutfağa git
 - Ekmek al
 - Çayı hazırla
 - Dolaptan kahvaltılıkları çıkar
 - Bardağın bitince çayını doldur
 - Karnın doyunca sofradan kalk
 - Kahvaltılıkları dolaba koy
 - Sofrayı temizle

Örnek

- Adım 1:Yataktan kalk
- Adım 2:Mutfağa git
- Adım 3:Ekmeğ al
- Adım 4:Çayı hazırla
- Adım 5:Dolaptan kahvaltılıkaları çıkar
- Adım 6:Bardağın bitince çayını doldur
- Adım 7:Karnın doyunca sofradan kalk
- Adım 8:Kahvaltılıkaları dolaba koy
- Adım 9:Sofrayı temizle

Algoritmaların Sınıflandırılması

- Algoritmalar yapılarına ve biçimlerine göre 3 grupta incelenirler.
 - Basit (Lineer) Algoritmalar
 - Mantıksal Algoritmalar
 - Döngüsel Algoritmalar

Basit (Linear) Algoritmalar

- İçerisinde mantıksal ifadelerin yer almadığı, program akış dallanmalarının olmadığı algoritmalarlardır. Bu algoritmalarda akış düz bir halde baştan sona doğru olacaktır. Çoğunlukla küçük hesaplamaları gerçekleştirmek için kullanılırlar. Bir önceki algoritmaya bakılırsa bir karar yapısının olmadığı görülmektedir.
- Örnek:
 - Adım 1: Hesaplanacak kilometre uzunluğunuz giriniz; km
 - Adım 2: Girilen değeri 1000 ile çarpınız; $m = km * 1000$
 - Adım 3: Hesaplanan değeri ekrana yazdırınız; m

Örnek-Basit Algoritma

- Örnek: Dışarıdan girilen üç adet sayısının toplamını, çarpımını ve ortalamasını hesaplayan algoritma;
 - Adım 1: Üç adet sayı giriniz; a, b, c
 - Adım 2: Sayıların toplamını hesaplayınız; $\text{toplam} = a + b + c$
 - Adım 3: Sayıların çarpımlarını hesaplayınız; $\text{çarpım} = a * b * c$
 - Adım 4: Sayıların ortalamasını hesaplayınız; $\text{ort} = \text{toplam} / 3$
 - Adım 5: Sayıların toplamını, çarpımını ve ortalamasını ekrana yazdırınız.

Mantıksal Algoritmalar

- Algoritma içerisinde mantıksal karşılaştırmaların bulunduğu yapılardır. Mantıksal karşılaştırmalara göre algoritmanın akışı farklı adımlara geçecektir. Bu şekilde oluşturulan algoritmalara Mantıksal Algoritmalar denir.
- İlk oluşturulan algoritma biraz daha ayrıntılanırsa karar yapılarının ortaya çıktığı görülür.

Mantıksal Algoritmalar

- Adım 1: Yataktan kalk
- Adım 2: Mutfağa git
- Adım 3: Eğer ekmek yoksa ekmek al
- Adım 4: Çayı hazırla
- Adım 5: Dolaptan kahvaltılıkları çıkar
- Adım 6: Bardağın bitince çayını doldur
- Adım 7: Karnın doyunca sofradan kalk
- Adım 8: Eğer kahvaltılıklar bitmişse bulaşık makinesine koy
- Adım 9: Eğer kahvaltılıklar bitmemişse kahvaltılıkları dolaba koy
- Adım 10: Sofrayı temizle

Örnek-Mantıksal Algoritma

- Girilen üç adet sayı içinde en büyük sayıyı bulan algoritma
 - Adım 1: Üç adet sayı giriniz; a, b, c
 - Adım 2: En büyük sayı a olsun; $eb = a$
 - Adım 3: Eğer b en büyükten büyük ise en büyük b ($eb = b$) olsun
 - Adım 4: Eğer c en büyükten büyük ise en büyük c ($eb = c$) olsun
 - Adım 5: En büyük sayıyı ekrana yazdır; eb

Örnek-Mantıksal Algoritma

- Sayının pozitif, negatif veya sıfır olduğunu bulan algoritma
 - Adım 1: Sayıyı giriniz; a
 - Adım 2: Eğer a sayısı sıfırdan büyük ise ekrana 'pozitif' yaz ve adım 5'e git
 - Adım 3: Eğer a sayısı sıfırdan küçük ise ekrana 'negatif' yaz ve adım 5'e git
 - Adım 4: Eğer a sayısı sıfıra eşit ise ekrana 'sıfır' yaz ve adım 5'e git
 - Adım 5: Program bitti.

Döngüsel Algoritmalar

- Program için geliştirilen algoritmada bir işlem birden fazla tekrar ediyorsa döngülü algoritma yapısı kullanılır. Döngüsel algoritmalarda mantıksal karşılaştırma yapısı özel olarak kullanılır. Eğer algoritma içerisinde kullanılan mantıksal karşılaştırma işlemi sonucunda programın akışı karşılaştırma yapılan yerden daha ileriki bir adıma değil de daha önceki adıma gidiyorsa bu şekilde oluşturulmuş algoritmalara döngüsel algoritma denir. Yani döngüsel algoritmalarda mantıksal karşılaştırma sonucunda program daha önceki adımlara gider.

Örnek

Adım 1:Yataktan kalk

Adım 2:Mutfağa git

Adım 3: Eğer Ekmek yoksa Ekmek al

Adım 4: Çay suyunu ocağa koy

Adım 5: Suyun kaynamasını bekle

Adım 6: Eğer su kaynamadıysa Adım 5'e git

Adım 7:Eğer Su kaynadıysa Çayı demle

Adım 8:Bardağını doldur

Adım 9: Kahvaltını et

Adım 10: Eğer Karnın doyduysa sofradan kalk Adım 12'ye git

Adım 11: Eğer Bardağın bitti ise Adım 8'e git

Adım 12: Eğer Kahvaltılıklar bitmişse Kahvaltılıkları Bulaşık Makinesine koy

Adım 13:Eğer Kahvaltılıklar bitmemiş ise Kahvaltılıkları dolaba koy

Adım 14:Sofrayı temizle

Örnek-Döngüsel Algoritma

- Girilen iki sayının en büyük ortak bölenini (EBOB) bulan algoritma

Adım 1: sayıları giriniz; a, b

Adım 2: Eğer a b'den büyükse b'yi a'dan çıkar ve tekrar a'ya yaz ($a=a-b$) ve Adım 2'e git

Adım 3: Eğer b a'dan büyükse a'yı b'den çıkar ve tekrar b'ye yaz ($b=b-a$) ve Adım 3'e git

Adım 4: Eğer a b'ye eşitse Adım 5'e git

Adım 5: en büyük ortak bölen a değerini ekrana yaz

Örnek-Döngüsel Algoritma

■ Bir sayının faktöriyeli

Adım 1: faktöriyeli hesaplanacak sayıyı giriniz; n

Adım 2: faktöriyel değerini 1 yap; $f=1$

Adım 3: indeks değerini 1 yap; $i=1$

Adım 4: faktöriyel değeri ile indeksi çarp ve hesaplanan değeri faktöriyele yaz; $f=f \times i$





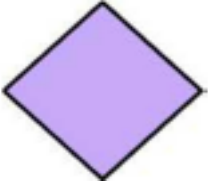


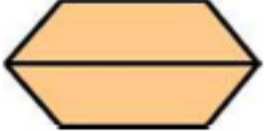
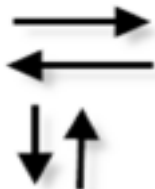
Adım 5: indeks değerini bir artır; $i=i+1$

Adım 6: eğer indeks değeri hesaplanacak sayıdan küçük veya eşit ise Adım 4' git

Adım 7: faktöriyel değerini ekrana yaz; f

AKIŞ DİYAGRAMLARI(FLOWCHARTS)

- Algoritma ile oluşturulan çözümler sözel olarak ifade edildiğinden daha standart olması herkes tarafından ortak olarak aynı anlamın çıkarılabilmesi için akış diyagramları kullanılır. Akış diyagramları sembollerden oluşmaktadır. Her sembolün belli bir işlevi vardır. Akış diyagramlarında birden fazla sembol kullanılabilir.

	Programın Başladığını ve bittiğini gösteren sembol
	Giriş Bilgilerinin yer aldığı sembol
	Çıkış Bilgilerinin yer aldığı sembol
	İşlemlerin yer aldığı işlem sembolü
	Karar ve Kontrol İşlemlerinin sembolü
	Bağlantı noktası sembolü
	Döngü Sembolü
	Altprogram Sembolü
	Akış yönü sembolleri

Akış Diyagramı Sembolleri

Örnek

Başla

Sembollü kullanılır.

Adım 1: üç adet sayı giriniz; $a \times b \times c$

Giriş Bilgilerinin yer aldığı sembolün içerisine girilmesi gereken değerler yazılır.

a,b,c

Adım 2: sayıların toplamını hesaplayınız; $\text{toplam} = a + b + c$

Bu adımda işlem gerçekleştirileceği işlemlerin yer aldığı sembol kullanılmaktadır.

Gerçekleştirilecek işlem bu sembol içerisine yazılır.

Toplam= $a+b+c$

Adım 3: sayıların çarpımını hesaplayınız; $\text{çarpım} = a \times b \times c$

Örnek

Çarpım=axbxc

Adım 4: sayıların çarpımını hesaplayınız; $ort = toplam / 3$

Yine bu adımda işlem gerçekleştirileceği için işlemlerin yer aldığı sembol kullanılmaktadır.

Ort=Toplam/3

Adım 5: sayıların toplamını, çarpımını ve ortalamasını ekrana yazdırınız; toplam, çarpım, ort

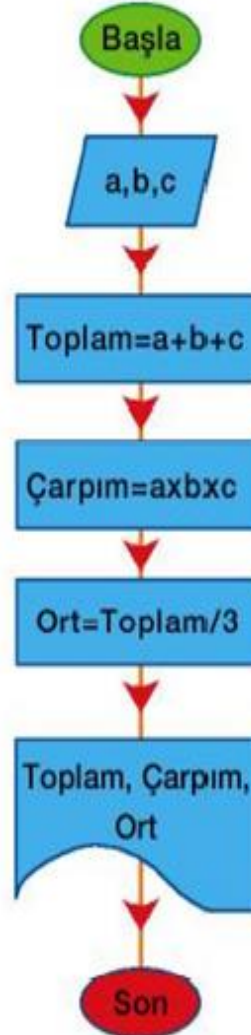
Bu adımda kullanıcılara hesaplanan bilgilerin gösterilmesi gerçekleşeceği için çıkış bilgilerinin yer aldığı sembol kullanılacaktır. Çıkışı yapılacak değerler bu sembolün içerisine yazılır.

**Toplam, Çarpım,
Ort**

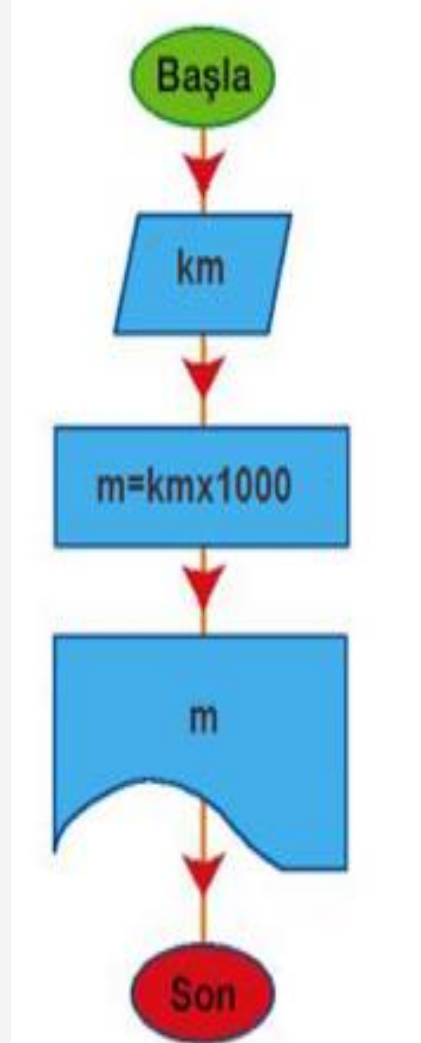
Son işlem ise başlangıçta olduğu gibi program bitirilirken problemin çözümü için yapılması gereken ön işlemler (açılan parantezlerin kapatılması gibi) için

Son

Örnek-Basit Akış Diyagramı



Örnek-Basit Akış Diyagramı



Örnek-Basit Algoritma

Adım 1: üçgenin üç kenarı girilir; a, b, c

Adım 2: yüksekliği hesaplamada kullanılacak U değeri hesaplanır;

$$U=(a+b+c)/2$$

Adım 3: yüksekliği hesaplamada kullanılacak K değeri hesaplanır;

$$K=\sqrt{U(U-a)(U-b)(U-c)}$$

Adım 4: a kenarına ait yükseklik hesap edilir; $h1=2K/a$

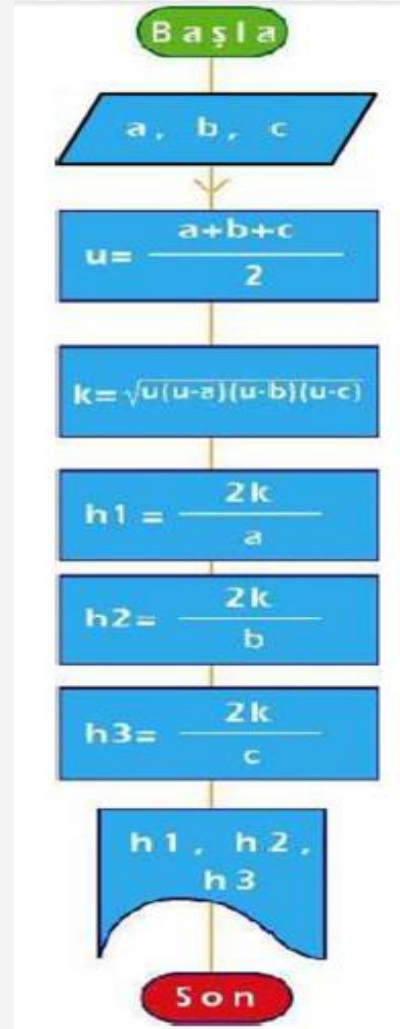
Adım 5: b kenarına ait yükseklik hesap edilir; $h2=2K/b$

Adım 6: c kenarına ait yükseklik hesap edilir; $h3=2K/c$

Adım 7: hesaplanan yükseklikler ekrana yazılır; $h1,h2,h3$

Kenarları verilmiş üçgenin kenarlarına ait yüksekliklerin bulunması

Örnek-Basit Akış Diyagramı



Örnek-Mantıksal Algoritma

Adım 1: üç adet sayı giriniz; a, b, c

Adım 2: en büyük sayı a olsun; $eb=a$

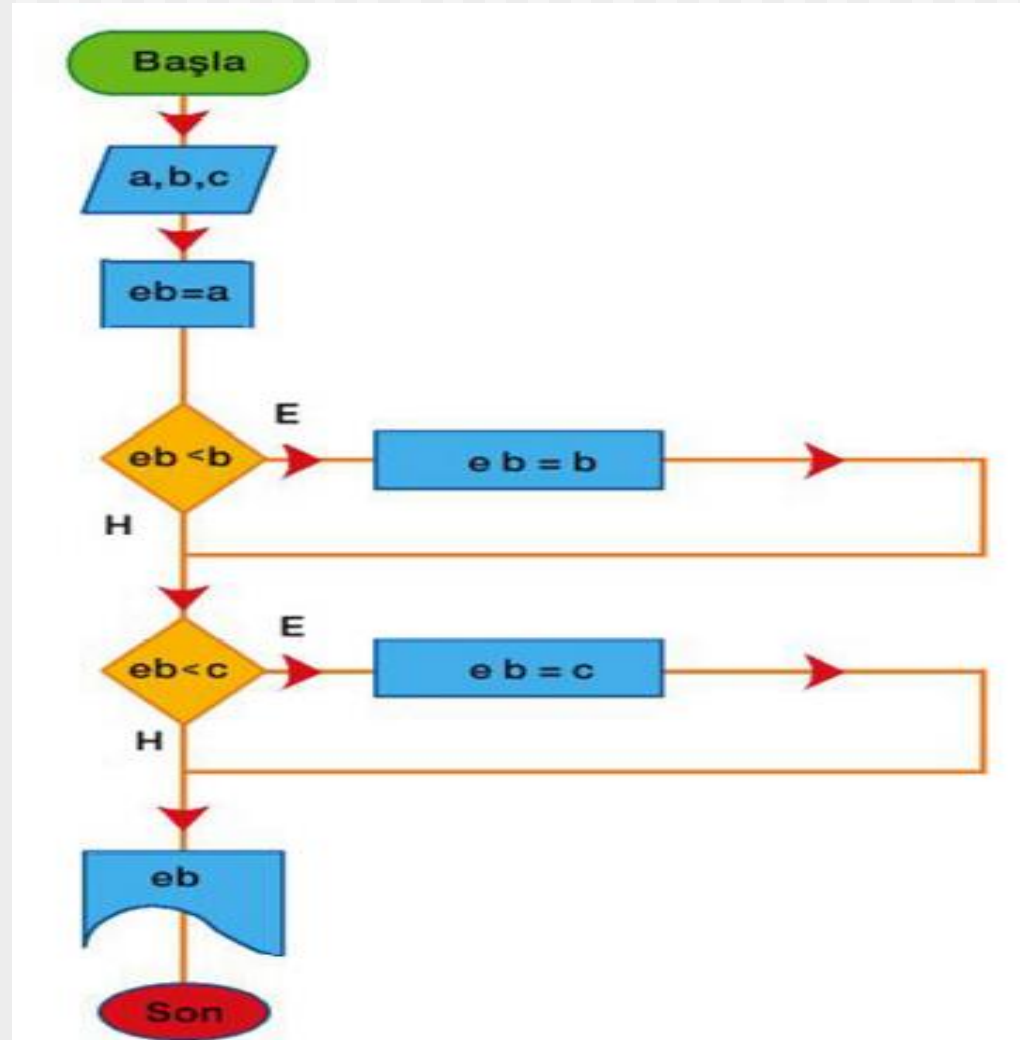
Adım 3: Eğer b en büyükten büyük ise en büyük b ($eb=sayi2$) olsun

Adım 4: Eğer c en büyükten büyük ise en büyük c ($eb=c$) olsun

Adım 5: En büyük sayıyı ekrana yazdır; eb

3 adet sayıdan en büyük olanını bulma

Örnek-Mantıksal Akış Diyagramı



Örnek-Mantıksal Algoritma

Başka bir örnek ile girilen bir sayının pozitif, negatif veya sıfır olduğunu bulan algoritmanın akış diyagramını gerçeklersek;

Adım 1: sayıyı giriniz; a

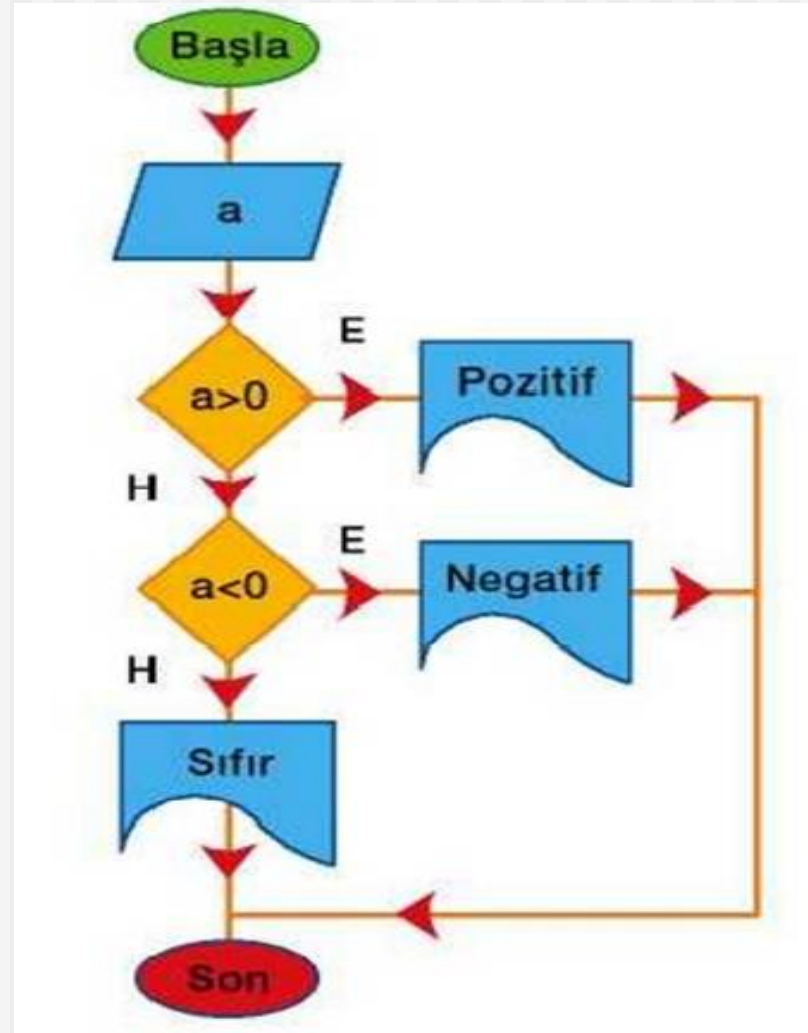
Adım 2: Eğer a sayısı sıfırdan büyük ise ekrana “pozitif” yaz ve adım 5’ya git

Adım 3: Eğer a sayısı sıfırdan küçük ise ekrana “negatif” yaz ve adım 5’ya git

Adım 4: Eğer a sayısı sıfırdan küçük ise ekrana “sıfır” yaz

Adım 5: program bitti

Örnek-Mantıksal Akış Diyagramı



Örnek-Mantıksal Algoritma

□ İkinci Dereceden Bir Denklemin Köklerinin Bulunması

Adım 1: hesaplanacak denklemin katsayılarını giriniz; a, b, c

Adım 2: kökleri bulmak için önce Δ değeri hesaplanır; $\Delta=b^2-4ac$

Adım 3: Eğer Δ 'nın değeri 0' dan küçükse "Ekranı kök yoktur" yaz ve adım 8'e git

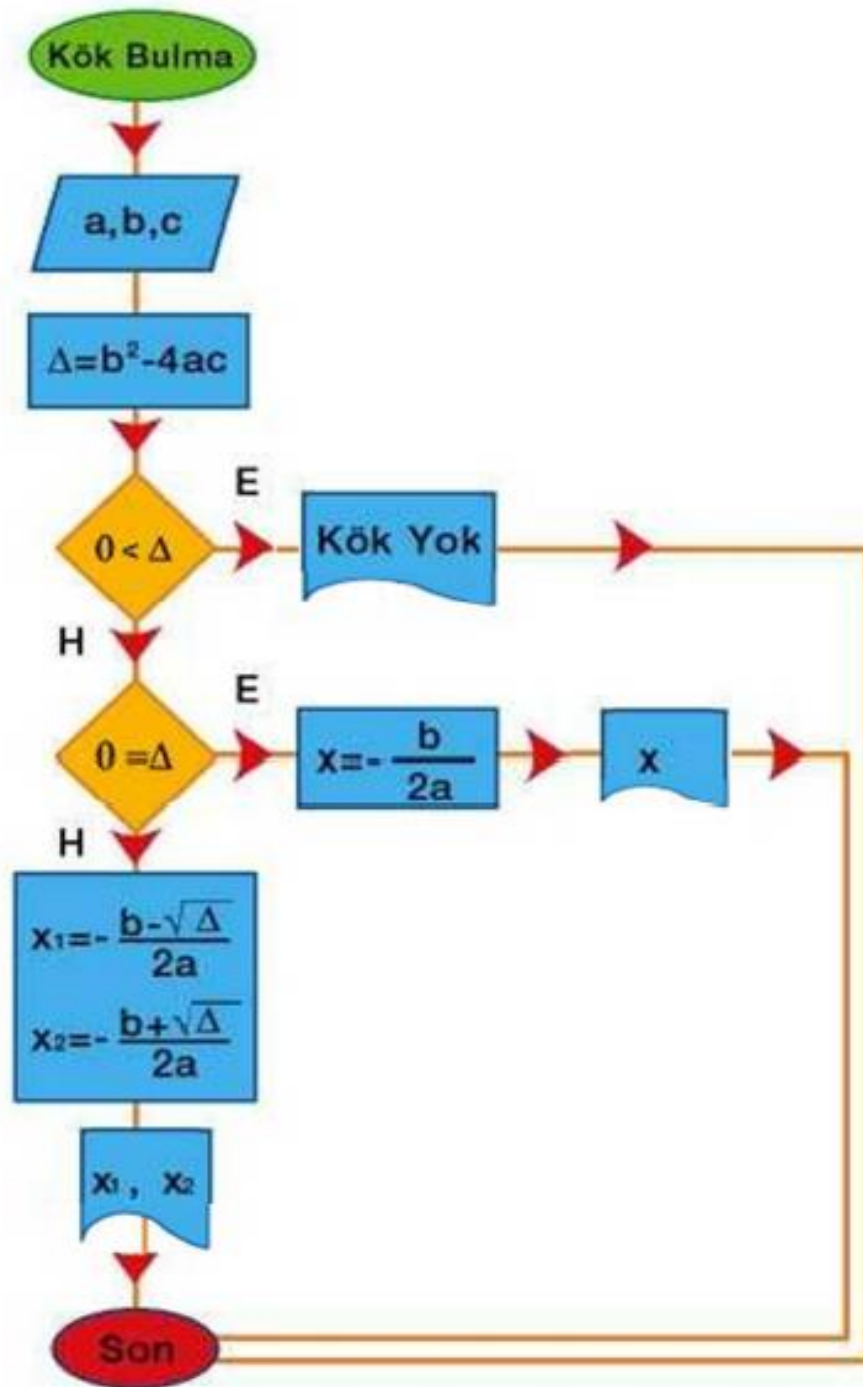
Adım 4: Eğer Δ 'nın değeri 0 eşitse kökü hesapla; $x=b/2a$; ekrana yaz ve adım 8'e git

Adım 5: Eğer Δ 'nın değeri 0' dan büyükse farklı iki kök vardır

Adım 6: Kökleri hesapla; $x_1=(-b-\sqrt{\Delta})/2a$ diğer kök $x_2=(-b+\sqrt{\Delta})/2a$

Adım 7: Hesaplanan kökleri ekrana yaz

Adım 8: programı bitti



Örnek-Mantıksal Akış Diyagramı

Örnek-Döngüsel Algoritma

Adım 1: sayıları giriniz; a, b

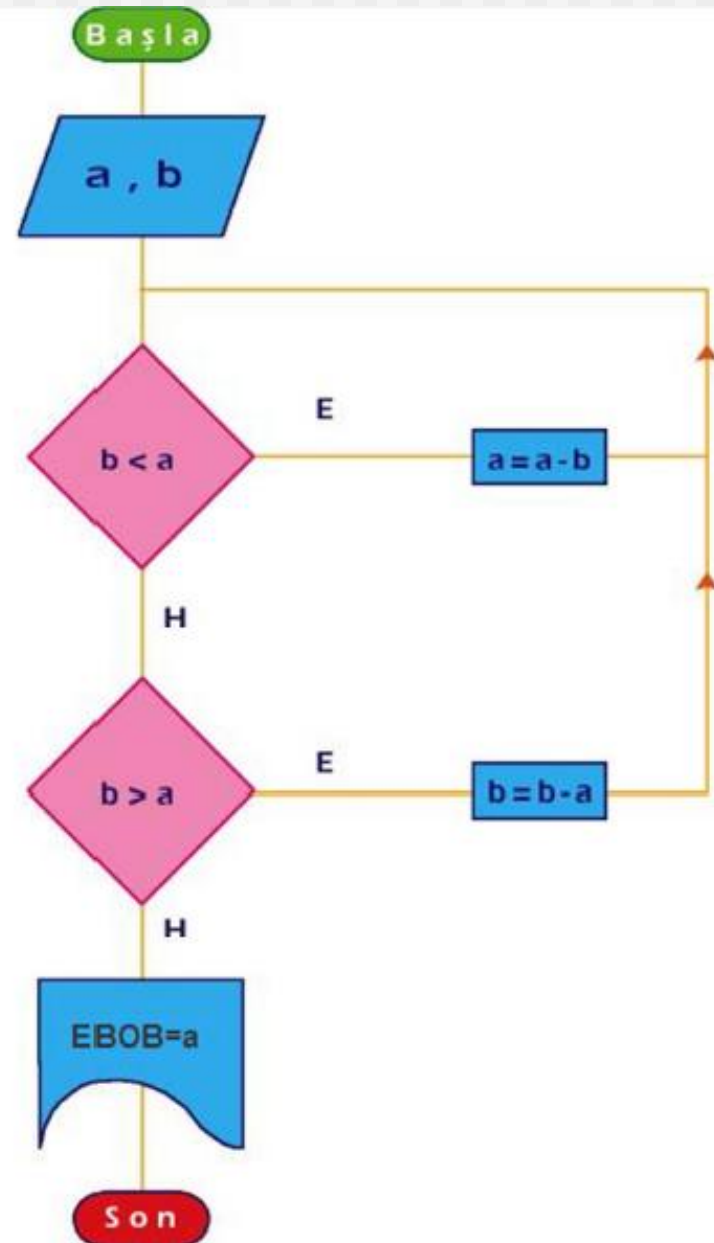
Adım 2: Eğer a b'den büyükse b'yi a'dan çıkar ve tekrar a'ya yaz ($a=a-b$) ve Adım 2'e git

Adım 3: Eğer b a'dan büyükse a'yı b'den çıkar ve tekrar b'ye yaz ($b=b-a$) ve Adım 3'e git

Adım 4: Eğer a b'ye eşitse Adım 4'ya git

Adım 5: en büyük ortak bölen a değerini ekrana yaz

**İki sayının en büyük ortak
bölenini bulma**



Örnek-Döngüsel Algoritma

Adım 1: faktöriyel hesaplanacak sayıyı giriniz; n

Adım 2: faktöriyel değerini 1 yap; $F=1$

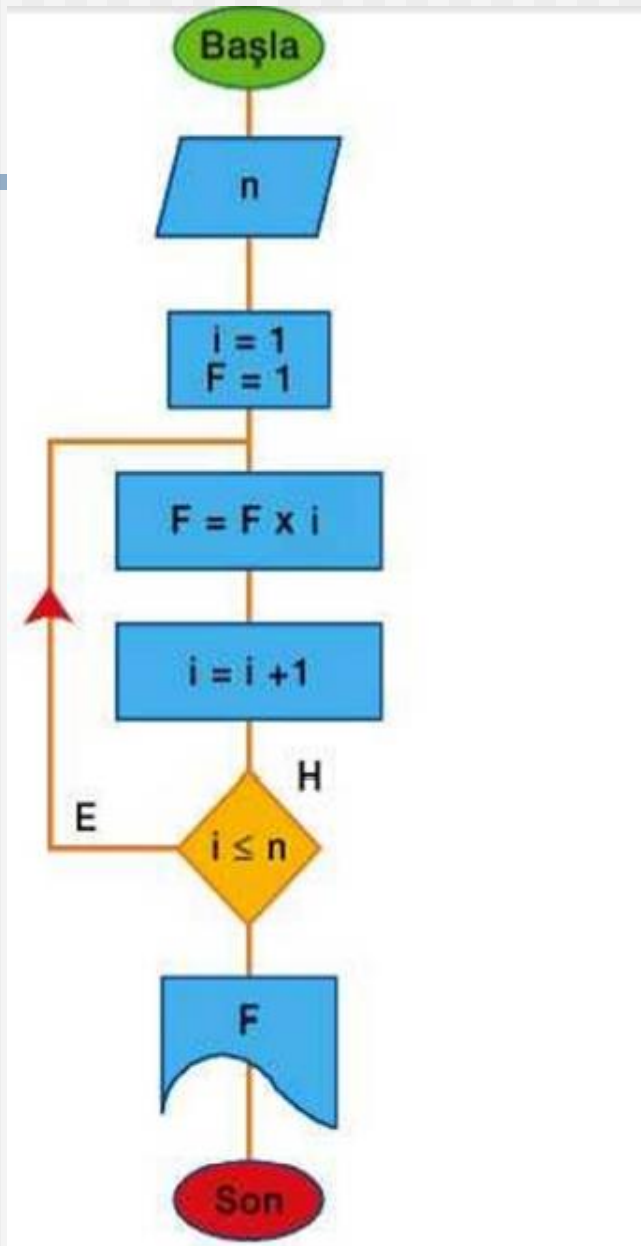
Adım 3: indeks değerini 1 yap; $i=1$

Adım 4: faktöriyel değeri ile indeksi çarp ve hesaplanan değeri faktöriyele yaz; $F=F \times i$

Adım 5: indeks değerini bir artır; $i=i+1$

Adım 6: eğer indeks değeri hesaplanacak sayıdan küçük veya eşit ise Adım 4' git

Adım 7: faktöriyel değerini ekrana yaz; F



Örnek-1'den n'e kadar olan sayıların toplamı

Adım 1: toplamı hesaplanacak sayıyı giriniz; n

Adım 2: Toplama işlemi sırasında döngüde kullanılacak

Döngü sayacına 1 ve toplama işleminde

kullanılacak değere 0 atanır; Sayı=1, Toplam=0

Adım 3: toplama değeri ile sayaç toplanır ve toplama değerine atanır;

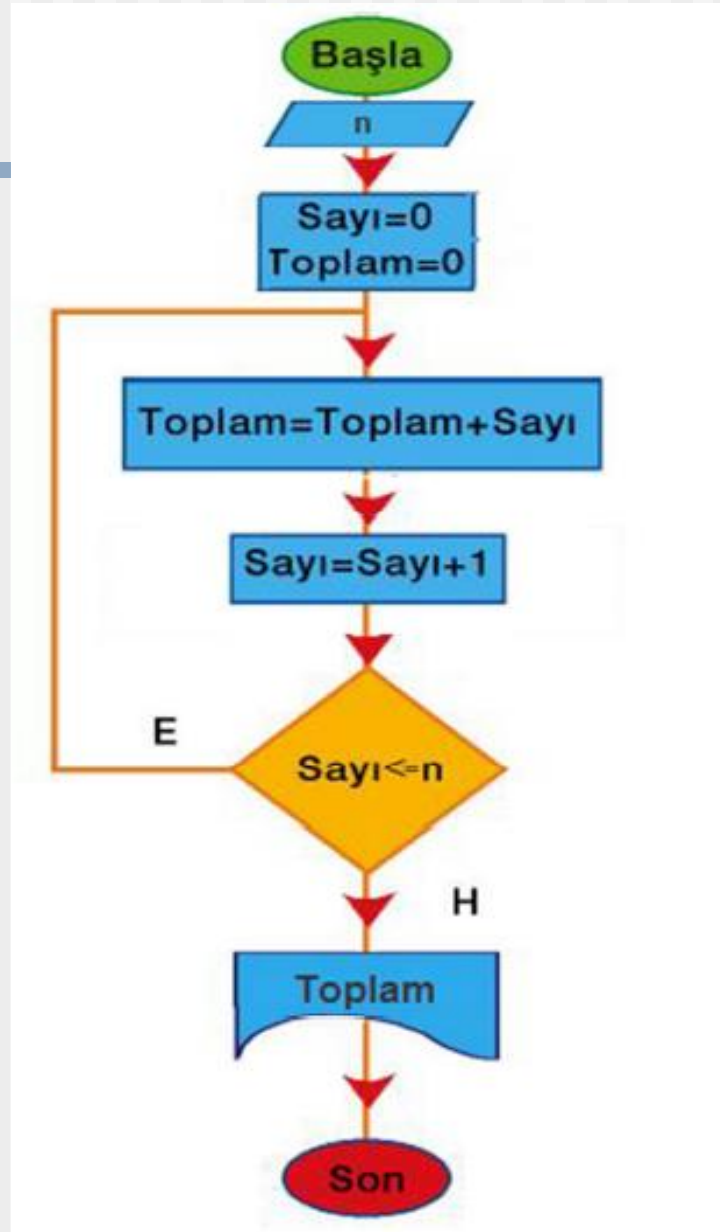
$\text{Toplam} = \text{Toplam} + \text{Sayı}$

Adım 4: döngü işlemi her adımda bir artırılır; Sayı=Sayı+1

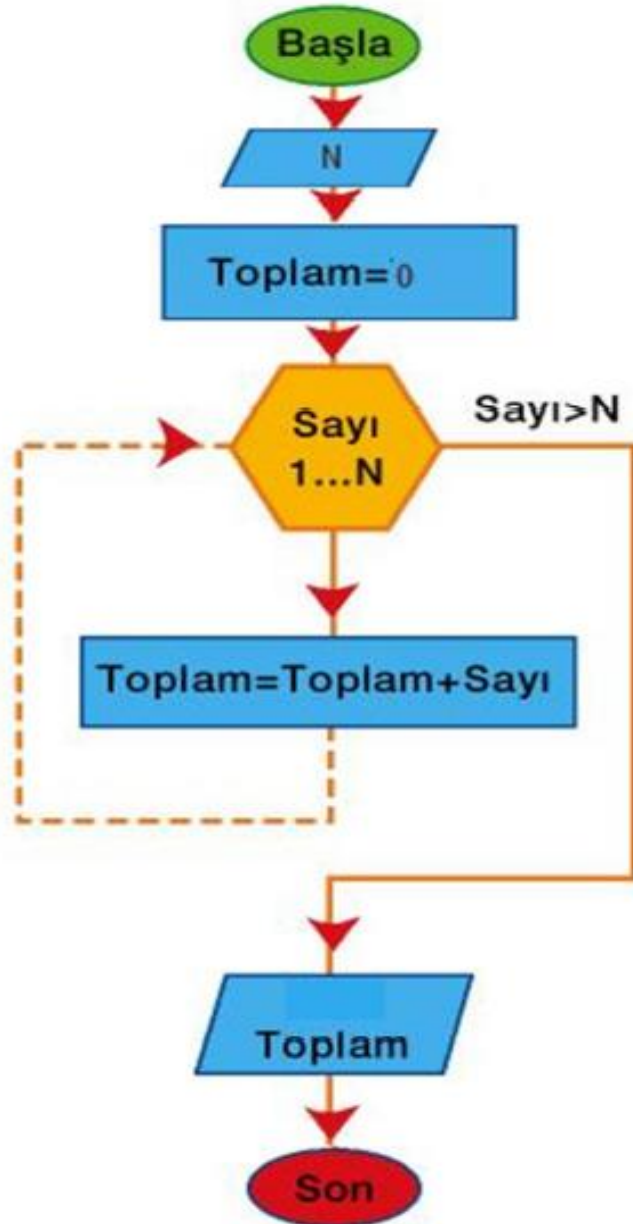
Adım 5: Eğer döngü sayacı toplamı hesaplanacak sayıdan küçük ve eşit ise

adım 3'e git

Adım 6: hesaplanan toplam değeri ekrana yaz; Toplam



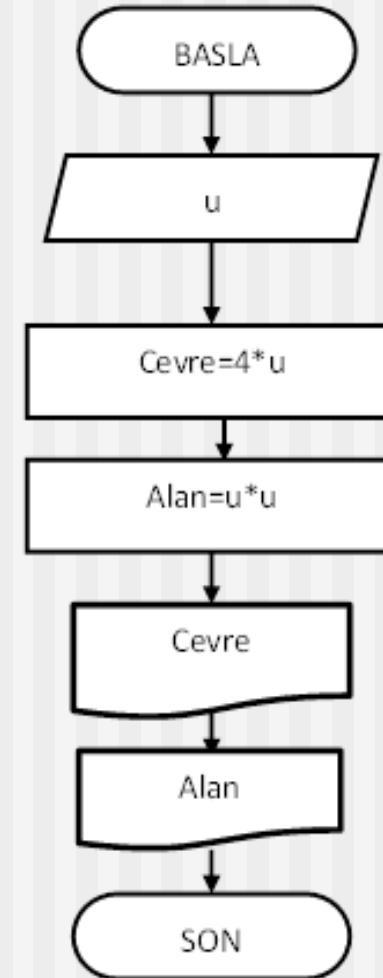
Sayaç ile çözüm



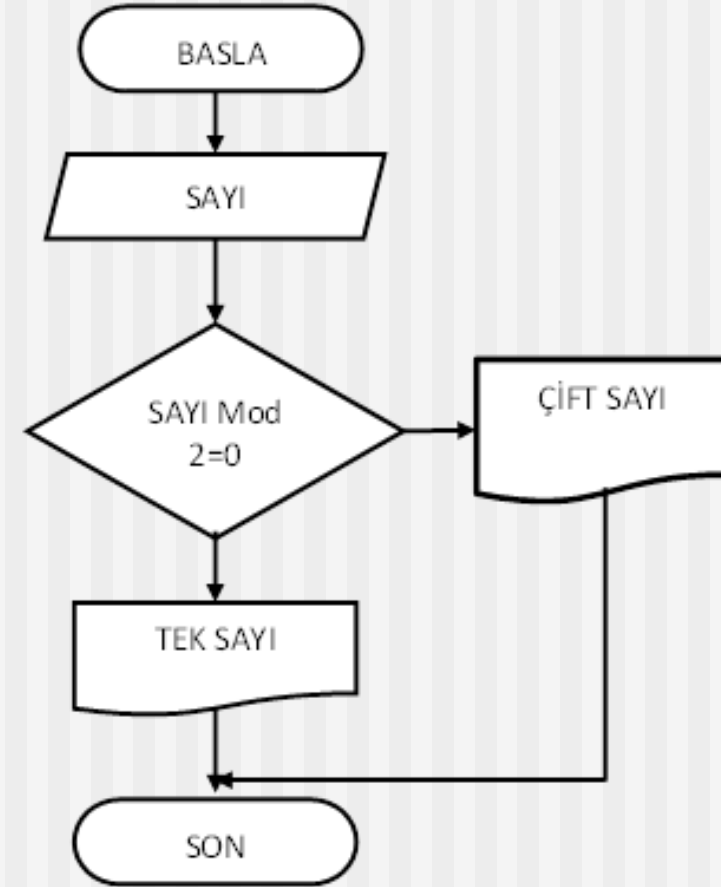
Döngü ile çözüm

Örnekler

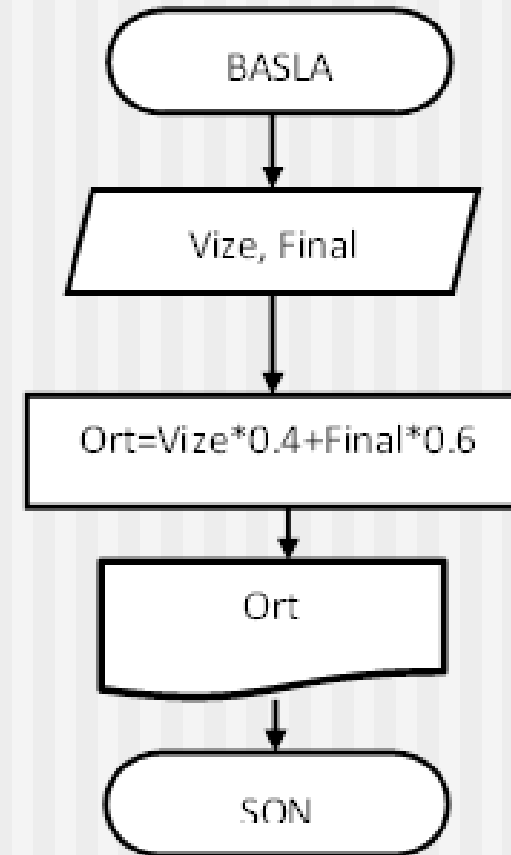
1-Kenarı verilen karenin alanı ve çevresini bulan programın akış şemasını çiziniz.



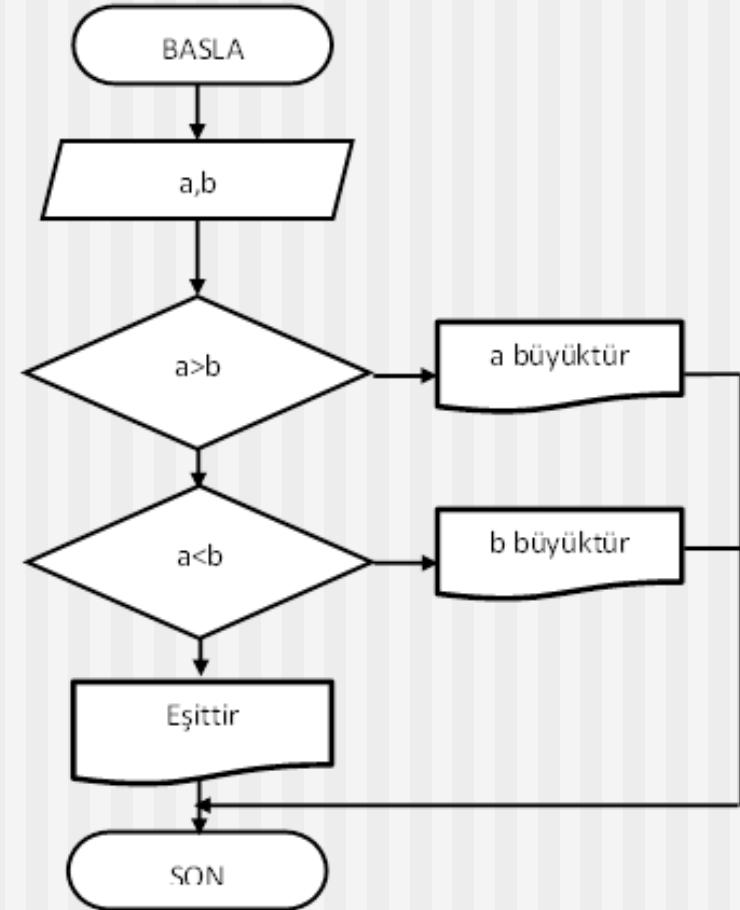
2-Girilen bir sayının tek mi çift mi olduğunu bulan programın akış şemasını çiziniz.



3- Not ortalamasını veren programın akış şemasını çiziniz.



4- Girilen iki sayının büyük olanını ekrana yazdıran programın akış şemasını çiziniz.



5-1 ile 100 arasındaki tek sayıların toplamını bulan programın akış şemasını çiziniz.

