

Bilgisayar Programlama - I

1-Algoritma Giriş

Tanım

- Algoritmalar, problemleri çözmek için adım adım prosedürlerdir.
- Algoritma, soruların herhangi verilen bir sınıfına cevaplar bulmakta kullanılabilen bir hesaplama prosedürü için etkili komutların kümesidir

Yapısı

- Atama adımları, (Bir değişkene bazı değerlerin atanması gibi)
- Aritmetik adımlar,
(Toplama,bölme,çıkarma,çarpma gibi)
- Mantıki adımlardır. (İki sayının karşılaştırılması gibi)

Algoritma Oluşturma

- Sonlu bir tamsayı dizisinin en büyük elemanını bulmak için bir örnek algoritmayı ele alalım.

Çözüm

- Maksimum değerini geçici olarak dizinin ilk elemanına eşitliyoruz.
- Dizinin bir sonraki elemanıyla geçici maksimum değerini karşılaştırıyoruz, eğer dizinin bir sonraki elemanı geçici maksimum değerinden büyükse, geçici maksimum değerine bu sayıyı atıyoruz.
- Eğer dizide başka sayılar varsa önceki adımları tekrarlıyoruz.
- Dizide karşılaştıracak tamsayı kalmadığında duruyoruz. Bu noktada geçici maksimum dizideki en büyük tamsayıdır.

Algoritmaların Özellikleri

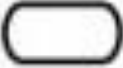
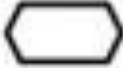





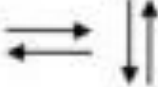





- **Giriş:** Bir algoritma açıkça belirtilen bir kümeden giriş değerlere sahiptir.
- **Çıkış:** Bir algoritmanın her bir giriş değerinin kümesinden, çıkış değerinin kümesi üretilir. Çıkış değerleri problemin sonucunu içerir.
- **Tanımlılık:** Algoritmanın adımları tam olarak tanımlanmalıdır.
- **Sonluluk:** Bir algoritma herhangi bir giriş kümesi için sonlu sayıdaki adımlardan sonra istenilen sonucu üretmelidir.
- **Etkinlik:** Algoritmanın her bir adımı tam olarak ve sınırlı bir zamanda gerçekleşebilmelidir.
- **Genellik:** Prosedür, sadece belli giriş değerleri için değil istenilen formdaki bütün problemler için uygulanabilir olmalıdır.

2-Sözde Kod ve Akış Şemaları

Pseudocode

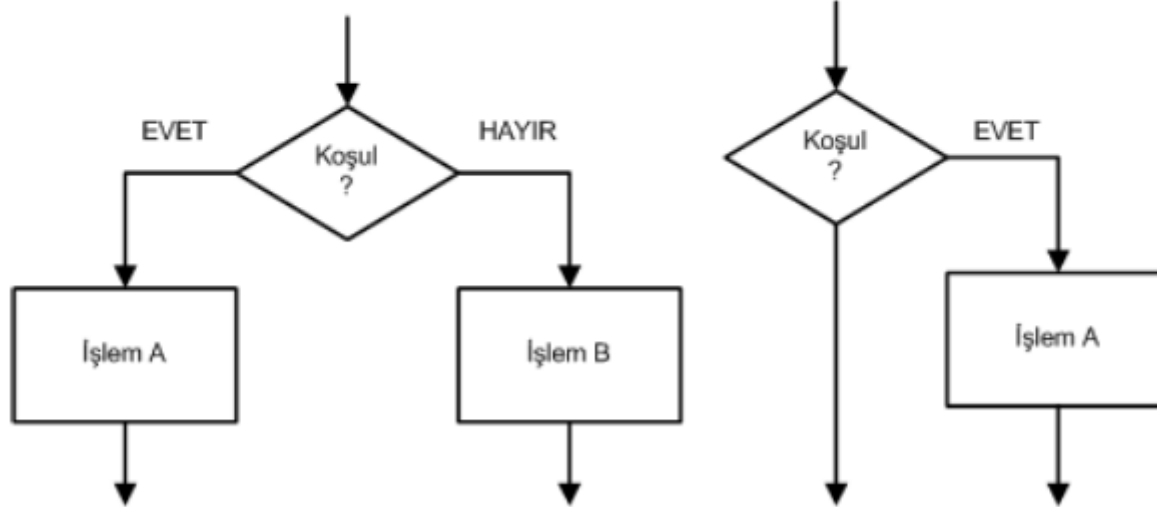
- Algoritmaları belirtirken özel bir bilgisayar dili yerine pseudocode kullanılır.
- Pseudocode bir algoritmanın İngilizce dil tanımını ve bu algoritmanın programlama diline dönüştürülmesi arasında bir ara basamak oluşturur.

Akış Şemaları

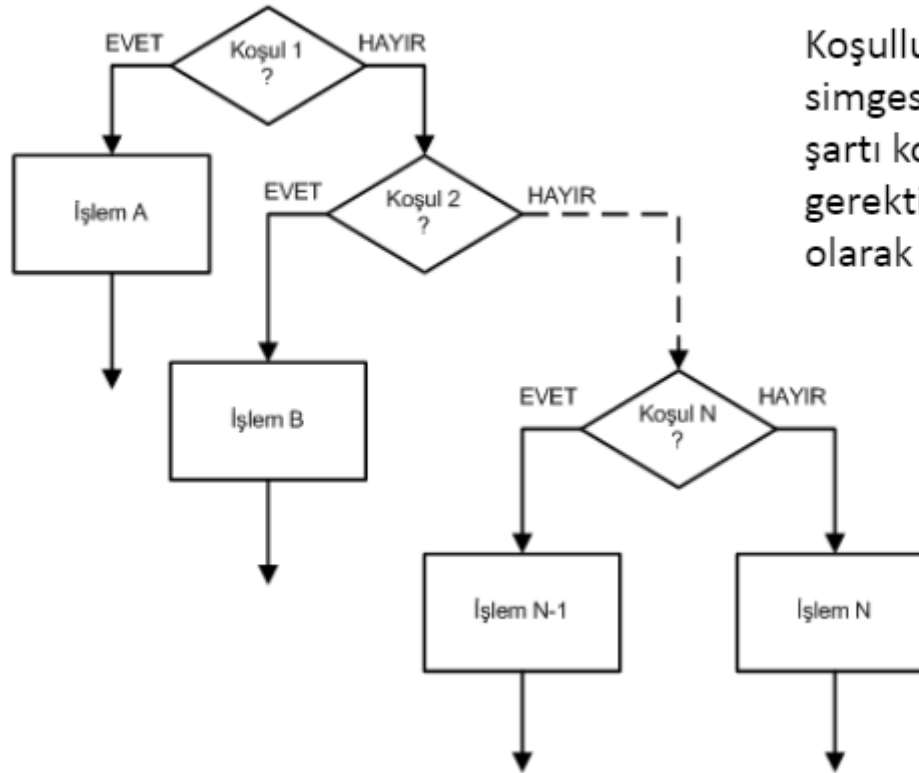
Simge	İşlev	Simge	İşlev
	Başla/Bitir		Döngü
	Genel Girdi/Çıktı		Manyetik Disk
	Genel İşlem		Yordam Çağırma
	Denetim (Karar)		Akış Yönü
	Yazıcı Çıktısı		Bağlaç
	Görüntü Çıktısı		Sayfa Bağlacı
	Ele ile Girdi (Klavye)		

Koşullu Dallanma Simgesi

Koşullu Dallanma simgesi baklava dilimi şeklindedir. Bir girişi ve iki çıkışı vardır. Çıkışlardan biri doğru ya da evet, diğeri yanlış ya da hayır çıkışıdır.



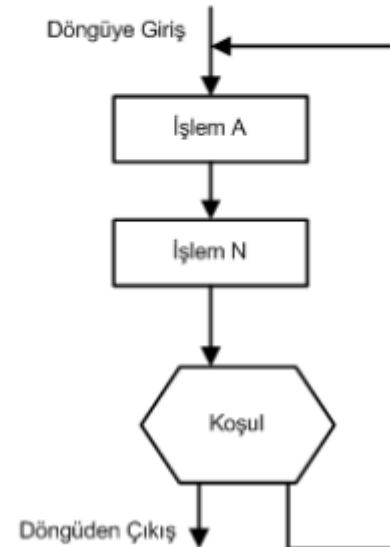
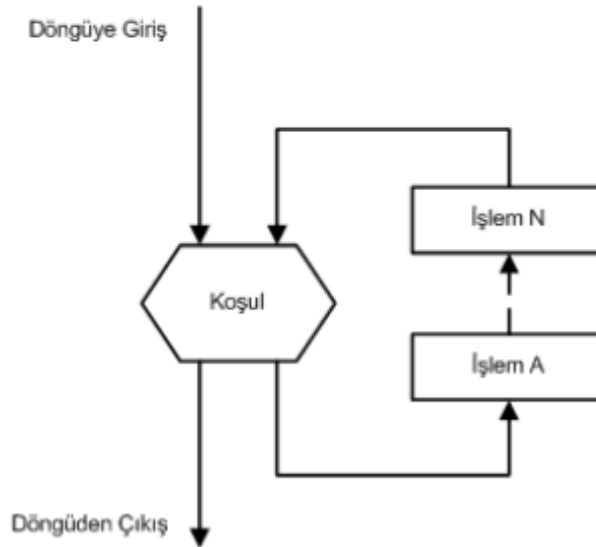
Koşullu Dallanma Simgesi



Koşullu dallanma simgesi, çok sayıda şartı kontrol etmek gerektiğinde ardışık olarak kullanılabilir.

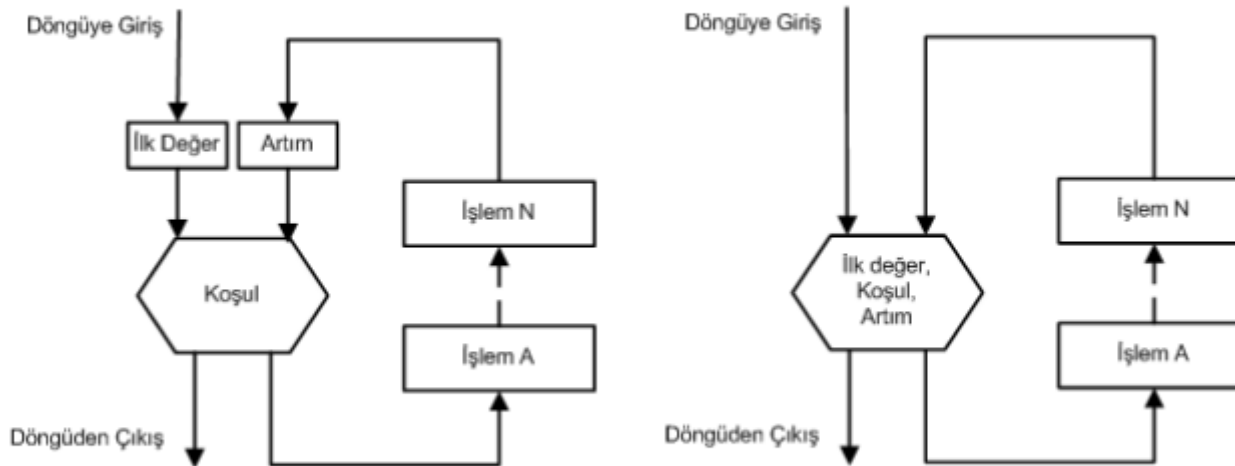
Döngü Simgesi

Döngü simgesi, basık altıgen şeklindedir. Göngüde belirtilen koşul doğru olduđu sürece döngü devam edecektir.



Döngü Simgesi

Tipik bir for döngüsü ise aşağıdaki gibi gösterilebilir.

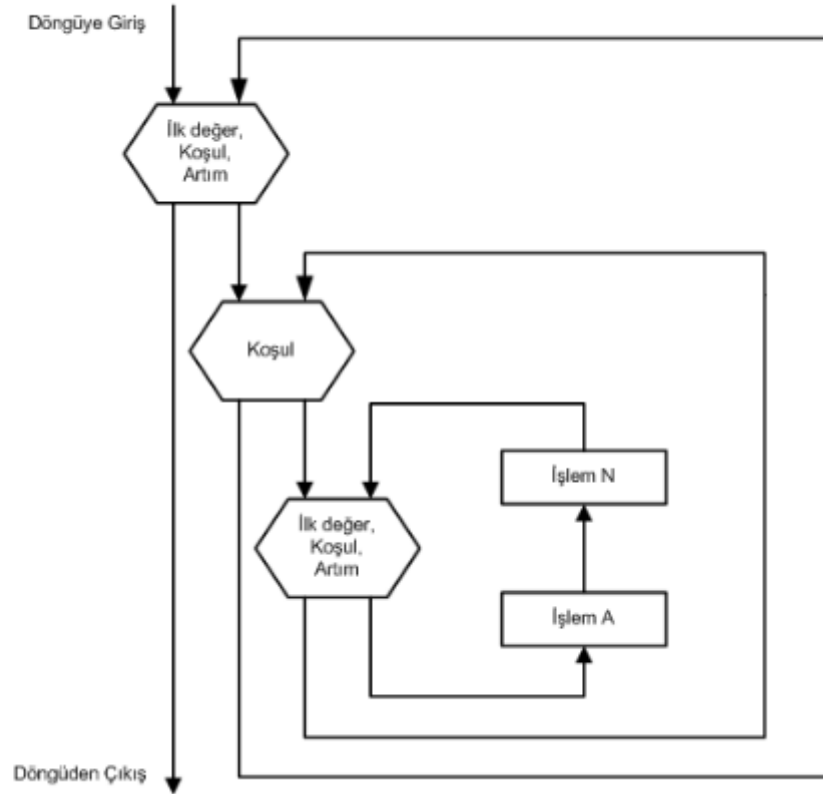


Ancak, ilk değer ve artım işlemlerini her defasında ayrıca göstermek yerine, bunları döngü simgesi içerisine yazmak daha pratik olacaktır.

İç İçe Döngüler

İç içe döngüler kurulabilir.

İç içe döngüler aynı tipte olabileceği gibi, farklı tipte döngülerden de oluşabilir.



Değişken

- Bir problemin çözümünde tanımlanan bir bilgi alanı, farklı adımlarda farklı değerler alabiliyorsa bu bilgi alanına “**değişken**” adı verilir.

- Bir değişken adı A ile Z arasındaki alfabetik harfler ile başlamalıdır. Değişken adı bir kelime ya da arada bir boşluk olmama koşuluyla bir cümle olabilir.

A, TOPLAM, SAYI, SONUC, TOPLAM, SAYI, SONUC, ADSOYAD gibi tanımlanabilir.

- Bir değişken adının ilk karakteri sayısal olamaz, yani 0 ile 9 arasında bir rakam ile başlayamaz. Ancak, ilk karakterden sonra istenilen bir sayı kullanılabilir.

A1, TOPLAM1, KIA37, B1589,..... şeklinde kullanılabilir.

- Değişken adı algoritmanın kodlanacağı programlama diline ait bir komut ya da deyim olamaz.

PRINT, END, NO, READ,..... şeklinde kullanılamaz.

- Algoritmada değişken adı verilirken Türkçe karakterler kullanılmamaya dikkat edilmelidir.

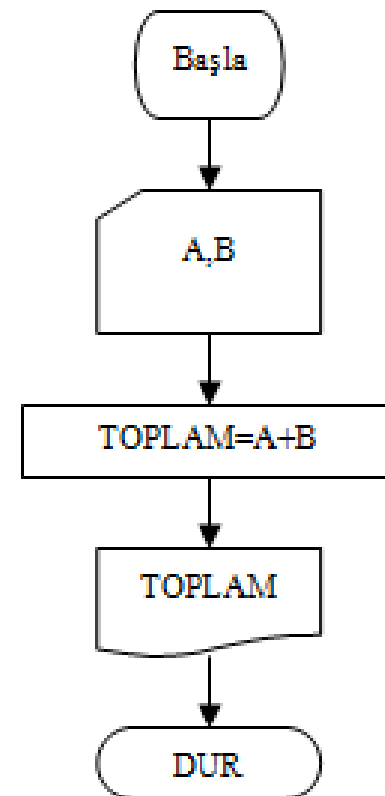
SONUÇ, DEĞER, KOŞUL,.... gibi.

3-Örnekler

Örnek 1

- Girilen iki sayının toplamını bulan algoritma ve akış şemasının oluşturulması.

A1. Başla,
A2. A ve B sayılarını gir/oku,
A3. $TOPLAM = A + B$ al,
A4. $TOPLAM$ 'ı yaz,
A5. Dur.

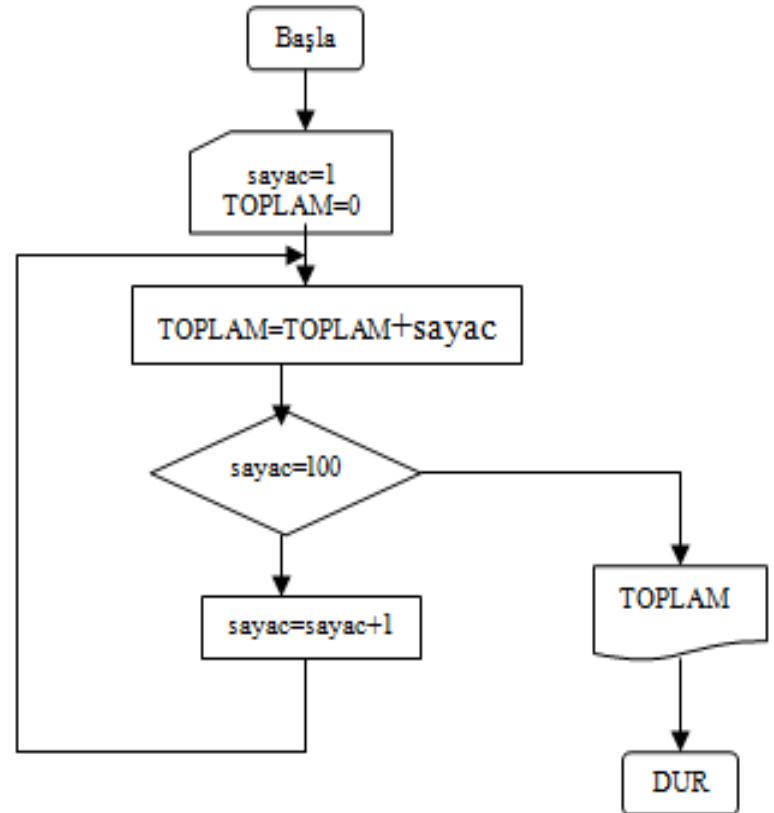


Örnek 2

- 1'den 100'e kadar olan tamsayıların toplamını bulan algoritma ve akış şemasının oluşturulması.

Çözüm 2

- A1. Başla,
- A2. $\text{sayac}=1$, $\text{TOPLAM}=0$ al,
- A3. $\text{TOPLAM}=\text{TOPLAM}+\text{sayac}$ al
- A4. Eğer $\text{sayac}=100$ ise A6. adıma git,
- A5. $\text{sayac}=\text{sayac}+1$ al ve A3. adıma geri dön,
- A6. TOPLAM değerini yaz,
- A7. Dur.

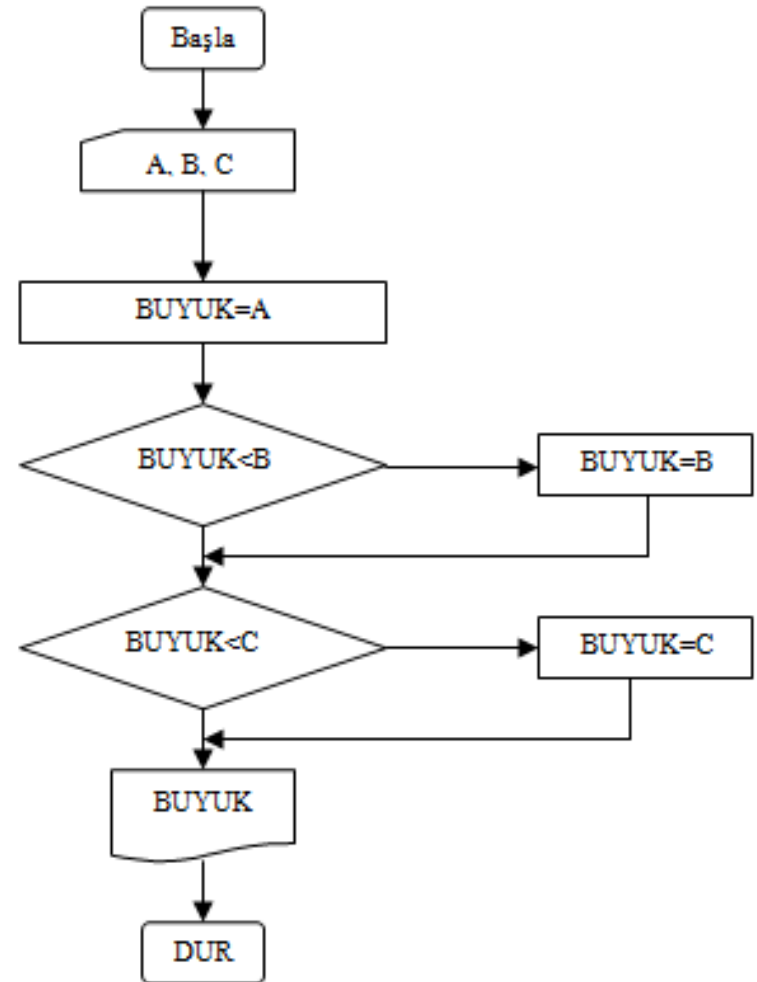


Örnek 3

- Girilen üç tamsayıdan en büyüğünü bulan algoritma ve akış şemasının oluşturulması.

Çözüm 3

- A1. Başla,
- A2. A, B ve C sayılarını gir,
- A3. $BUYUK=A$ al,
- A4. Eğer $BUYUK < B$ ise $BUYUK=B$ al,
- A5. Eğer $BUYUK < C$ ise $BUYUK=C$ al,
- A6. $BUYUK$ değerini yaz,
- A7. Dur.

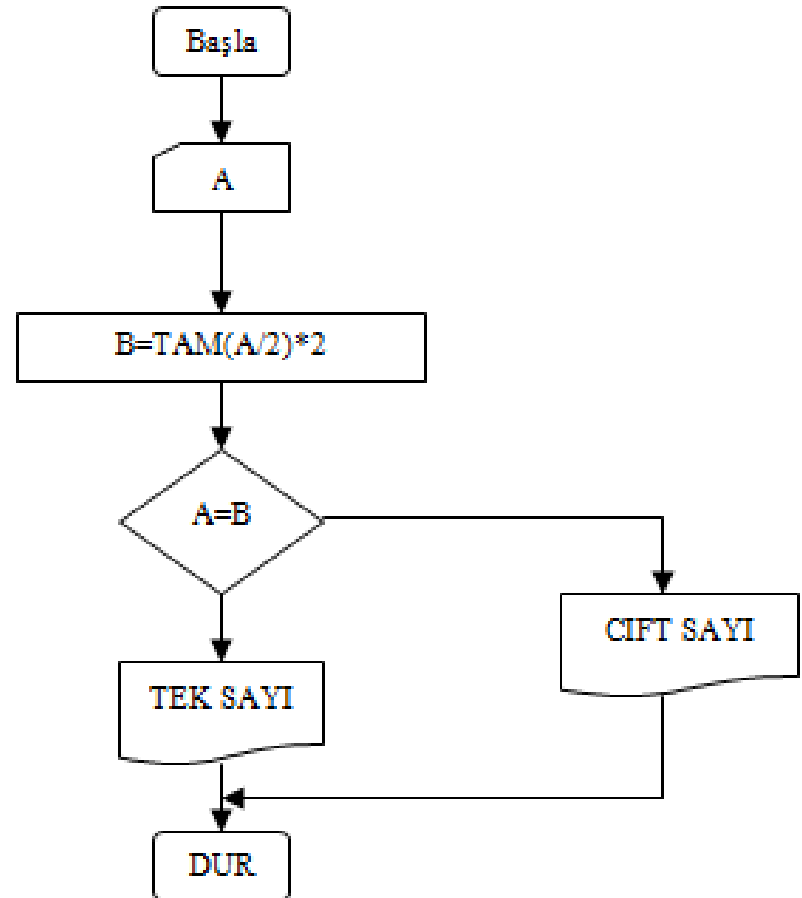


Örnek 4

- Girilen bir tamsayının tek ya da çift olduğunu tespit eden algoritma ve akış şemasının oluşturulması.

Çözüm 4

- A1. Başla,
- A2. A sayısını gir,
- A3. $B = \text{TAM}(A/2) * 2$ al,
- A4. Eğer $A = B$ ise A6. adıma git,
- A5. “Girilen sayı tek sayıdır” yaz ve A7. adıma git,
- A6. “Girilen sayı çift sayıdır” yaz,
- A7. Dur.

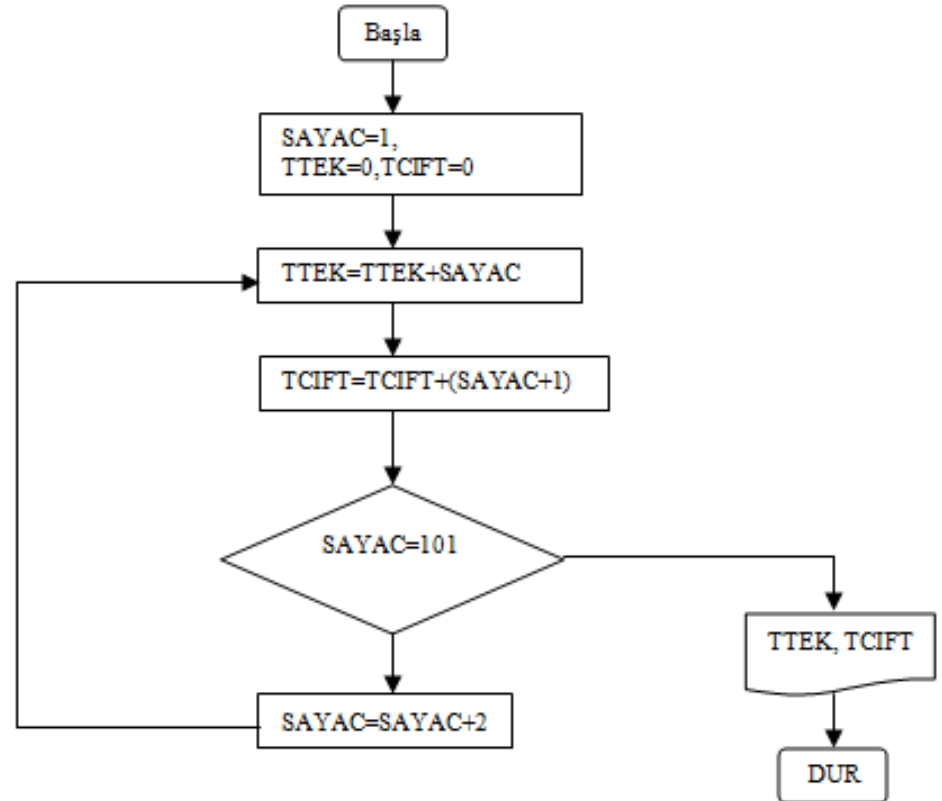


Örnek 5

- 1 ile 100 arasındaki tam sayılardan tek ve çift olanların ayrı ayrı toplamını bulan algoritma ve akış şemasının oluşturulması.

Çözüm 5

- A1. Başla
- A2. $SAYAC=1$, $TTEK=0$, $TCIFT=0$ al
- A3. $TTEK=TTEK+SAYAC$ al
- A4. $TCIFT=TCIFT+(SAYAC+1)$ al
- A5. Eğer $SAYAC=101$ ise A8. adıma git
- A6. $SAYAC=SAYAC+2$ al
- A7. A3 adımı geri dön
- A8. TTEK ve TCIFT değerini yaz
- A9. Bitir.

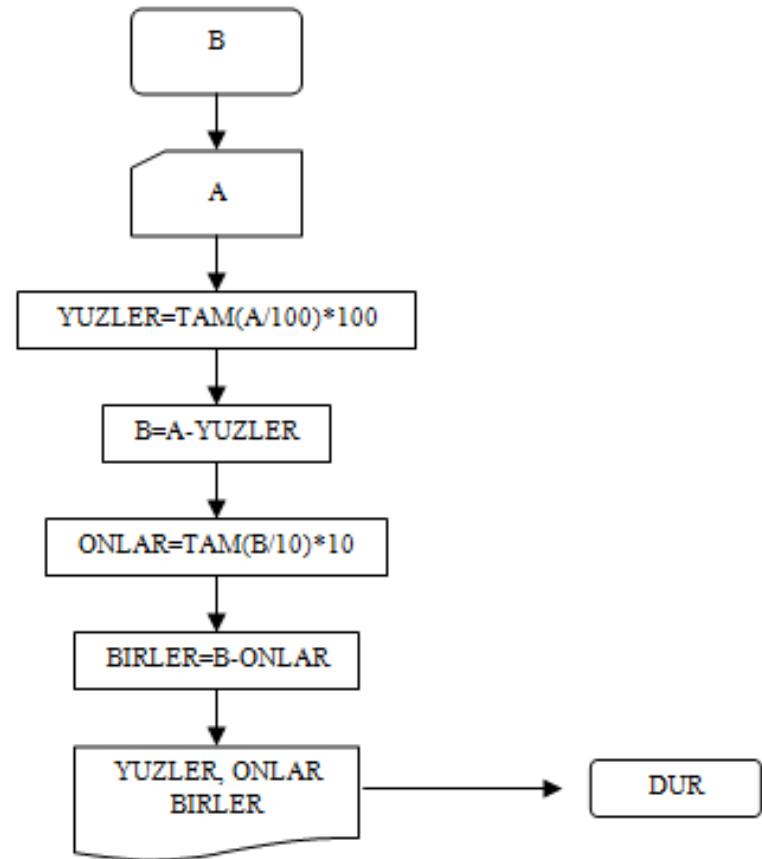


Örnek 6

- Üç haneli bir tamsayının birler, onlar ve yüzler hanesini bulan algoritma ve akış şemasının oluşturulması.

Çözüm 6

- A1. Başla,
- A2. A sayısını gir { 3 haneli bir sayı },
- A3. $YUZLER = TAM(A/100) * 100$ al
- A4. $B = A - YUZLER$ al,
- A5. $ONLAR = TAM(B/10) * 10$ al,
- A6. $BIRLER = B - ONLAR$ al,
- A7. YUZLER, ONLAR ve BIRLER değerlerini yaz
- A8. Dur.

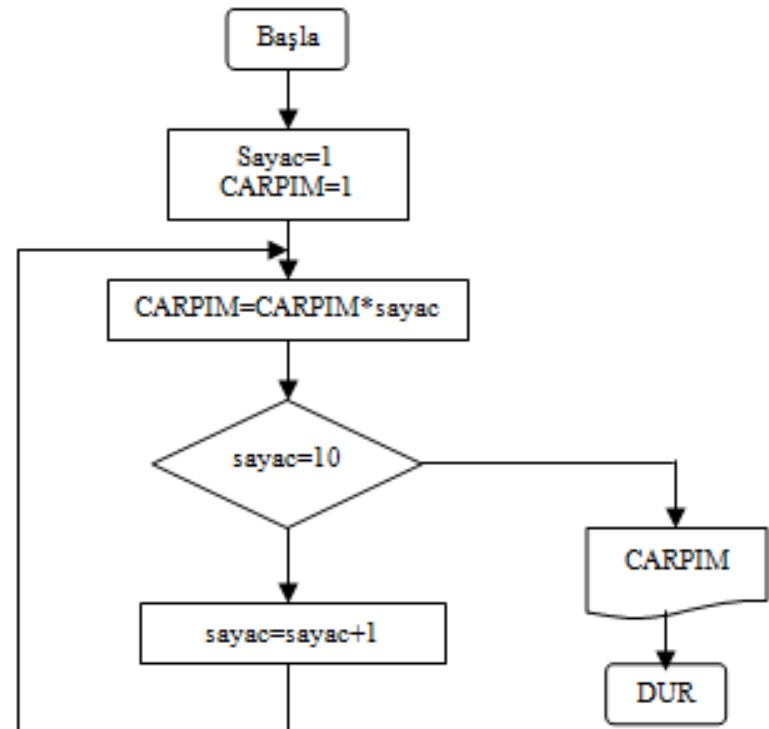


Örnek 7

- 1 ile 10 arasındaki tamsayıların çarpımı bulan algoritma ve akış şemasının oluşturulması.

Çözüm 7

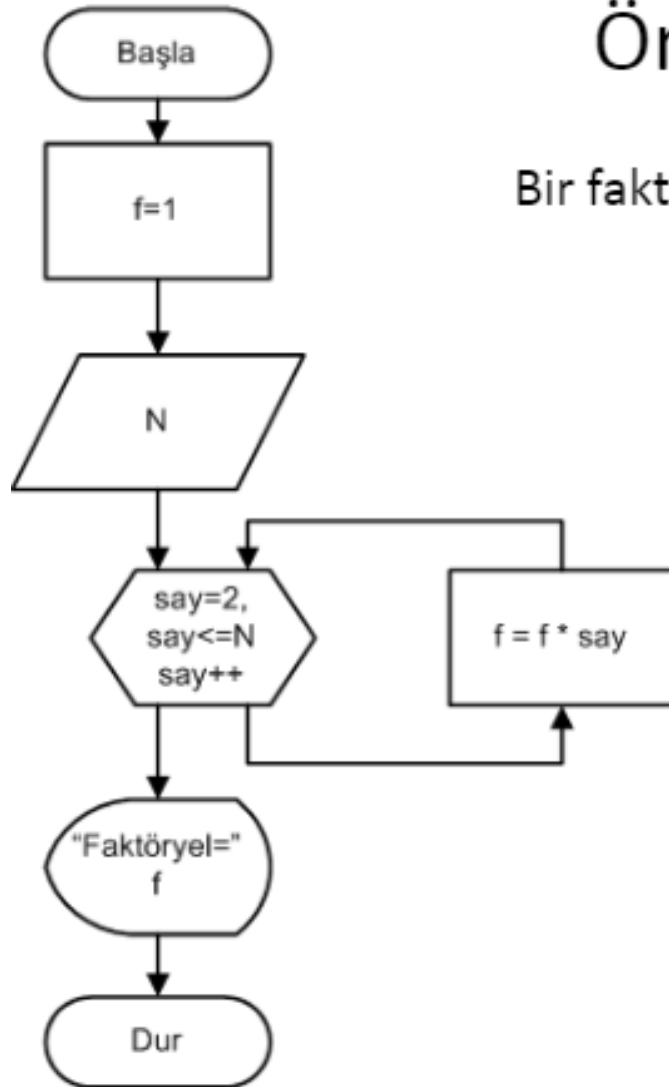
- A1. Başla,
- A2. Sayac=1, CARPIM=1 al,
- A3. $CARPIM = CARPIM * sayac$ al,
- A4. Eğer sayac=10 ise A7. adıma git,
- A5. $sayac = sayac + 1$ al,
- A6. A3. adıma geri dön,
- A7. CARPIM' ı yaz,
- A8. Dur.



4-İleri Örnekler

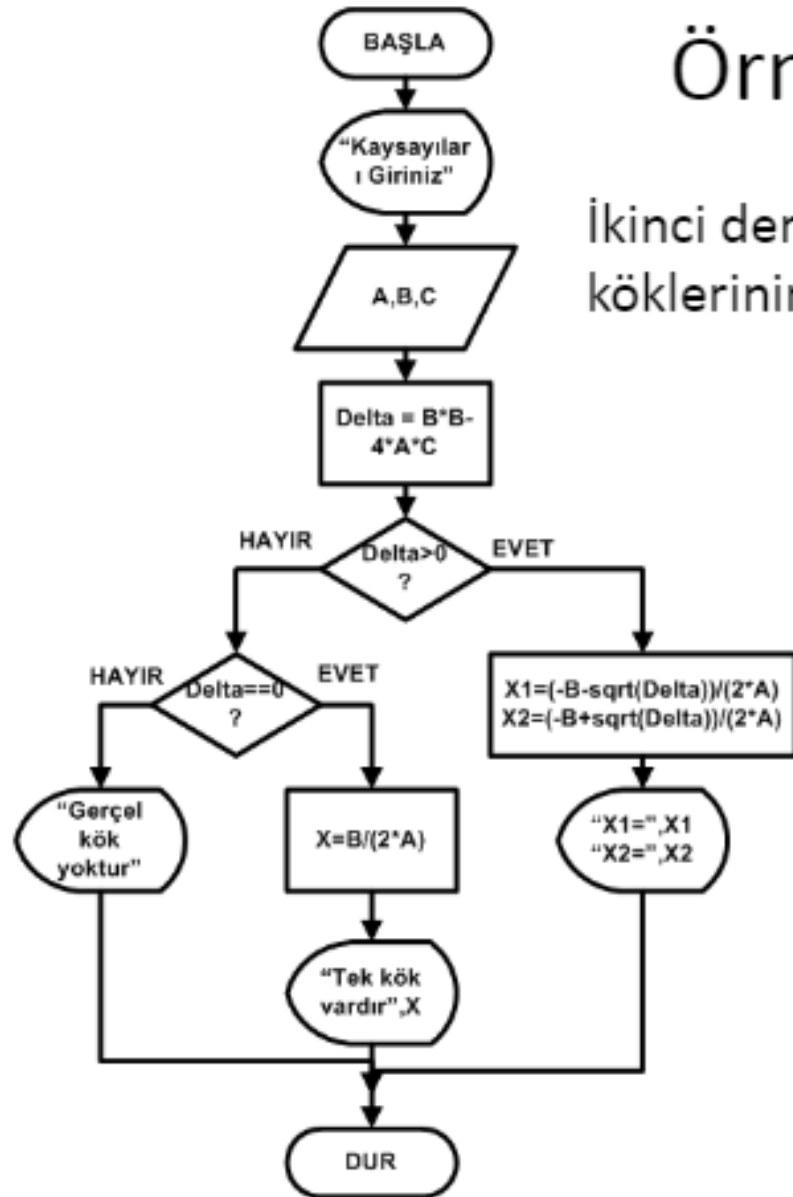
Örnekler

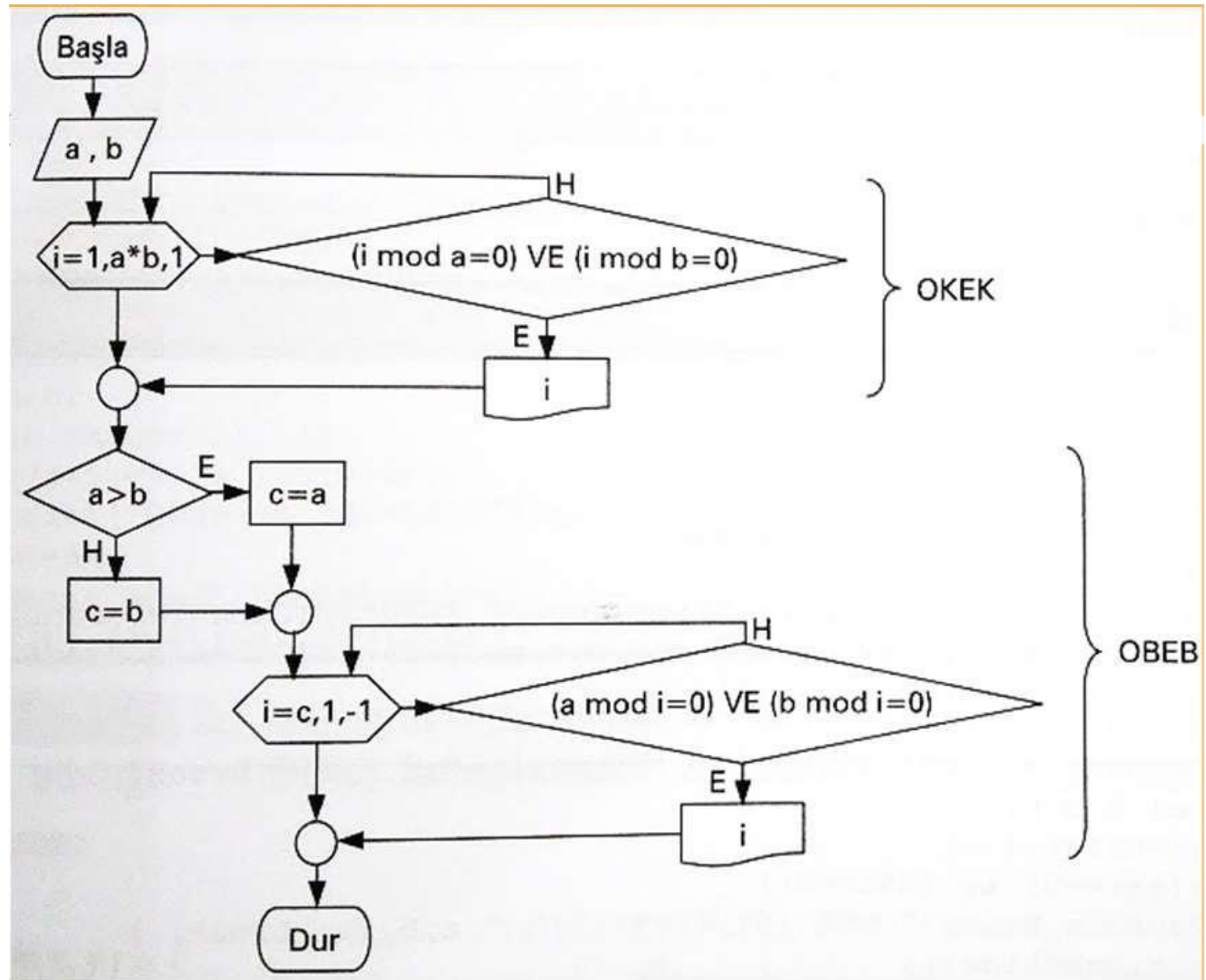
Bir faktöryel hesabının akış şemasını çiziniz.



Örnekler

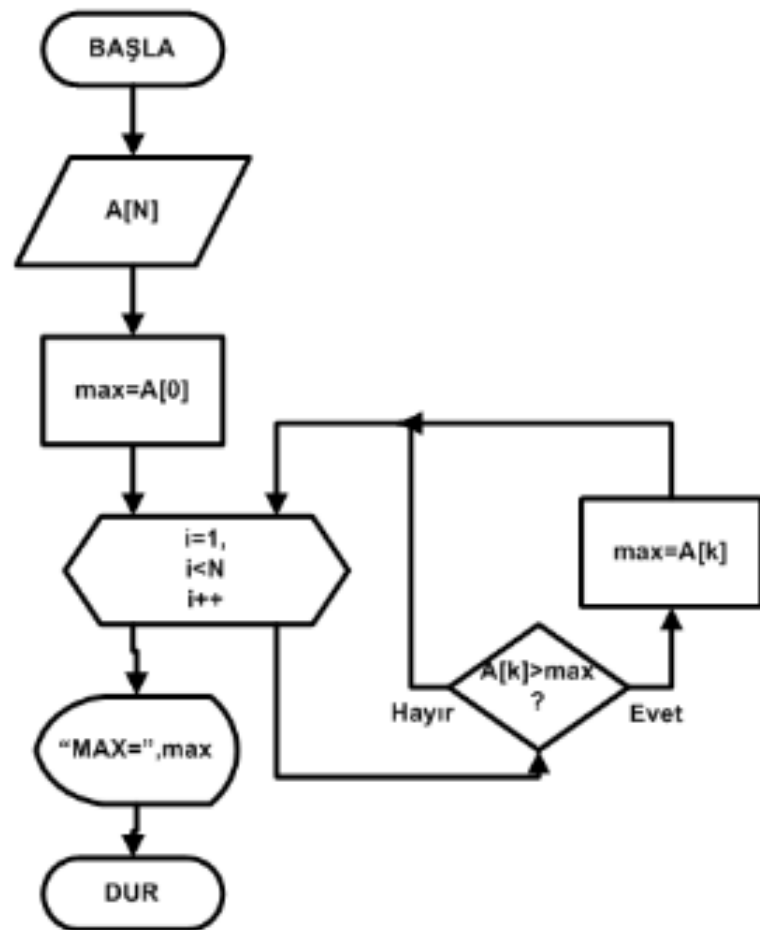
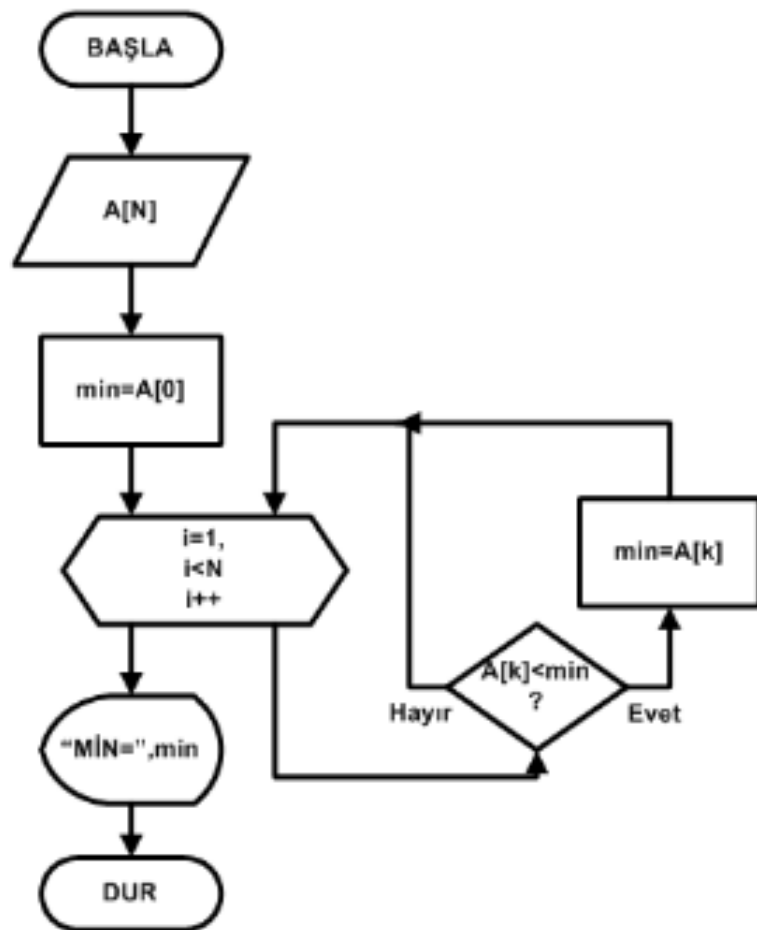
İkinci dereceden iki bilinmeyenli denklemin köklerinin hesaplanması





Örnekler

En küçük ve en büyük dizi elemanının bulunması



Örnekler

Matris toplamı

