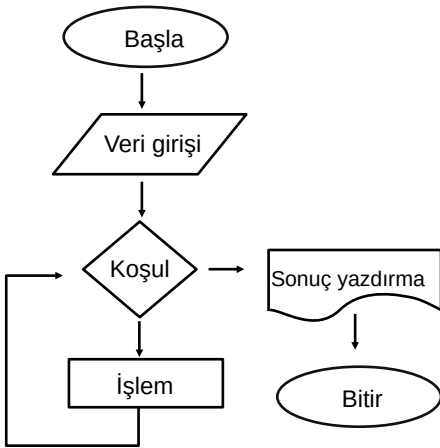


## ALGORİTMİK YAPILAR

Karmaşık problem durumlarında çözüm adımlarının belirli bir sıraya göre düzenlenmesini algoritmik yapılar ile sağlarız. Çözüm basamaklarında her adım bir sonraki adıma geçmeden önce doğru bir şekilde tamamlanır. Algoritmik düşünce ile genel çözüm yolları oluşturulduktan sonra benzer türde problemlere uygulanabilir ve bu sayede çözüm süreci hızlandırılarak problem çözüme daha verimli hâle gelmiş olur.

## Genel Algoritma Yapısı



### MANTIK BAĞLAÇLARININ ALGORİTMADAKİ ANLAMLARI VE İŞLEVLERİ

### ALGORİTMİK YAPILAR İÇERİSİNDEKİ MANTIK BAĞLAÇLARI VE NİCELEYİCİLER

Algoritmik yapılar içinde kullanılan mantık bağlaçlarının işlevleri (mantıksal operatörler) aşağıda verilmiştir.

**Ve bağlacı:** İki veya daha fazla önermenin hepsinin doğru olup olmadığını kontrol eder.

**Veya bağlacı:** İki veya daha fazla önermeden en az birinin doğru olup olmadığını kontrol eder.

**Ya da bağlacı:** İki önermeden yalnızca birinin doğru olup olmadığını kontrol eder.

**İse bağlacı:** Bir koşulun sağlanması durumunda belirli bir komutun yerine getirilmesini sağlar.

**Her Niceleyicisi** Bir önermenin herhangi bir öge için doğru olduğunu ifade eder.

**Bazı Niceleyicisi** Bir önermenin en az bir öge için doğru olduğunu ifade eder.

Operatörler
Ve $\wedge$
Veya $\vee$
İse $\Rightarrow$
Ya da $\nabla$
Her $\forall$
Bazı $\exists$

## Örnek...1 :

Tablo1: 6 öğrenciye ait burs komitesi verileri

İsim	A	B	C	D	E	F
Not Ortalaması	92	85	80	76	94	98
Gönüllü etkinlik sayısı	2	1	0	2	4	1
Bilimsel Proje Sayısı	1	1	0	0	3	2
Özürsüz devamsızlık	0	0	1	3	1	2
Davranış Notu	90	85	90	95	85	100

a) Öğrencilerin bilgilerine göre, algoritmik yapıyı ortaya koymak için koşul içeren sözel ifadelere karşılık gelen mantık bağlaçlarını ve niceleyicileri kullanarak tabloyu doldurunuz.

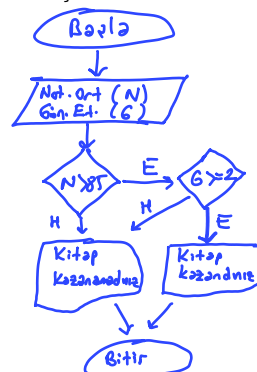
Koşulun sözel ifadesi	Koşulun mantık bağlacıyla ifadesi
Not ortalaması 90 dan büyük olanlar ve özürsüz devamsızlık gün sayısı 3 den küçük olanlar	$\text{Not ortalaması} > 90 \wedge \text{özürsüz devamsızlık gün sayısı} < 3$
Not ortalaması 85 den küçük veya gönüllü etkinlik sayısı 5'ten büyük olanlar	$\text{Not ortalaması} < 85 \vee \text{gönüllü etkinlik sayısı} > 5$
Davranış Notu 90 dan büyük olanlar ya da bilimsel proje sayısı 1 den büyük olanlar	$\text{Davranış Notu} > 90 \vee \text{bilimsel proje sayısı} > 1$
Sınıftaki bazı öğrencilerin gönüllü etkinlik sayısı 2'den fazladır	$\exists x \in \text{öğrenci}, x \text{ gönüllü etkinlik sayısı} > 2$

b) Tablo 1 deki bilgilere göre, öğrencilerden not ortalaması 90 üzerinde olup davranış notu da 85 üzerinde olan öğrencilerin belge ile ödüllendirilmesine karar verilmiştir. Buna göre, belge ile ödüllendirilecek öğrencileri veren koşulu mantık bağlaçları ve niceleyicilerle ifade ediniz.

$\forall x \in \text{öğrenci}$  belge ile ödüllendirilir  $\Leftrightarrow N.O > 90 \wedge D.N > 85$

(Tabloya göre A ve F belge alır)

c) Aşağıda belirtilen koşulları sağlayan öğrencilere kitap hediye edilecektir.  
 † Not ortalaması 85'den büyük olan.  
 ‡ En az 2 gönüllü etkinliğe katılımı olan.  
 Buna göre kitap hediye edilecek öğrencileri bulan algoritmanın işleyişini algoritmik doğal dil ile ifade edip, akış diyagramını oluşturunuz.



1.adım: Başla

2.adım: girdilerin alınması  
Not Ortalaması (N), Gönüllü Etkinlik sayısı (G)

3.adım: kitap kazanma durumunun bulunması  
Eğer not ortalaması 85 den büyükse ve gönüllü etkinlik sayısı 2 den büyük ya da eşit ise 'Kitap Kazandınız' yazdır  
Değilse 'Kitap Kazanamadınız' yazdır

4.adım: Bitir

Sembol	Anlamı
	Birleştirici veya bağlantı noktalarını temsil eder. Farklı işlem akışlarını birleştirmek veya akış sayfaları arasında bağlantı kurmak amacıyla kullanılan yapıdır.
	Döngü, istenen koşullarda tekrar eden (yinelenen) işlemler için kullanılan yapıdır.
$i=a,N,b$ ya da $i,a,N,b$	$i$ : döngü değişkeni $a$ : başlangıç değeri $N$ : bitiş değeri $b$ : döngü artış-azalış değerleri

Akış şemasında döngü oluşturulurken

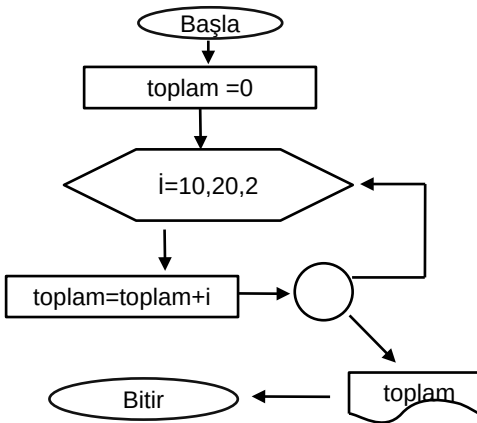
- 1) döngü değişkeni
- 2) başlangıç değeri
- 3) bitiş değeri

4) artış-azalış değerleri verilir.

Başlangıç değeri, bitiş değerinden küçük ise artan döngü; başlangıç değeri bitiş değerinden büyükse azalan döngü oluşur.

### Örnek...2 :

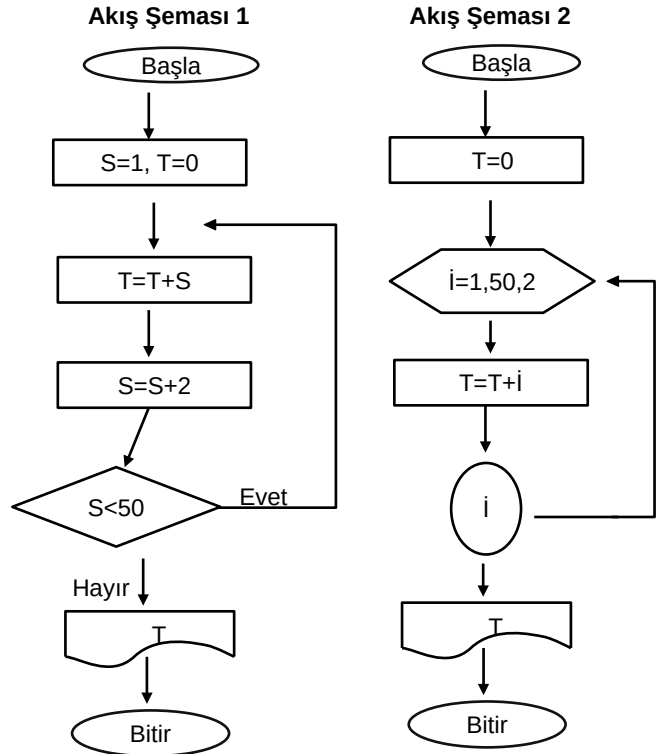
10 ile 20 arasındaki çift sayıların toplamını (10 ve 20 dahil) algoritmik akış şemasıyla gösterilmiştir, inceleyiniz.



Özetlersek: toplam 0 ile başladık, oku takip ederek döngü sembolüne geldik, döngü başlangıç değeri 10, dolayısıyla ilk olarak  $i=10$  aldık, oku takip ettik, toplam olan 0 ile  $i$  değeri 10'u topladık toplam 10 oldu. Döngünün birleştirici noktasına gittik ve bizi  $i$ 'nin sıradaki değeri olan 12 değerine yönlendirdi, şu halde toplam 10 ve  $i=12$ , işlemi tekrar yaptık ve toplam değeri  $10+12$  oldu, sırada  $i=14$  değeri kullanılacaktır. Bu şekilde devam ederek son  $i$  değeri olan 20 toplam ile toplandıktan sonra birleştirici noktası bizi toplamı yazdırmaya yönlendirir.

### Örnek...3 :

1 ile 50 arasındaki tek sayıların toplamını bulup yazdıran algoritmanın işleyişi iki farklı akış şeması ile verilmiştir. Şemaları inceleyip aralarındaki farkı açıklayınız. (S=sayaç, T=toplam dır)



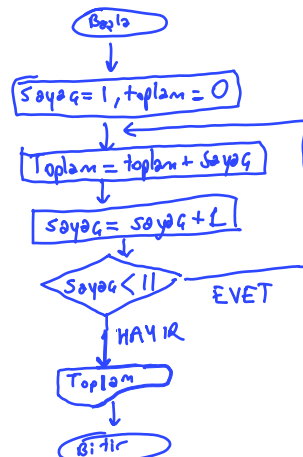
Akış Şeması 1'de döngü, koşul sembolü ile Akış Şeması 2'de ise döngü, döngü sembolü ile sağlanmıştır.

b) Akış Şeması 1'de başlangıçta toplam 0, sayaç 1 kabul edilmiş; sayaç 2 artım ile artırılmış koşulda sayacın 50'den küçük olması istenmiş ve her artımda toplam sayaç kadar artırılmıştır.

Akış Şeması 2'de başlangıçta toplam 0 kabul edilmiş, döngü değişkeni olarak  $i$  seçilip, döngünün 1'den başlayarak 50'ye kadar 2 artım değeri ile  $i$ 'nin artırılması sağlanmış ve her artımda toplama  $i$  eklenmiştir.

### Örnek...4 :

1'den 10'a kadar olan tam sayıların toplamını bulan algoritmayı akış şeması, algoritmik doğal dil ile ifade ediniz.



1. adım: Başla
2. adım: Girdilerin alınması  
Toplam = 0, sayaç = 1 olarak tanımla'
3. adım: Toplamın hesaplanması  
Toplam = Toplam + sayaç  
sayaç = sayaç + 1
4. adım: Kontrolün yapılması  
Eğer sayaç < 11 ise 3. adıma git  
değilse döngüyü sonlandır
5. adım: Sonucun yazdırılması  
Yazdır "Toplam"
6. adım: Bitiriniz.

Döngü koşul sembolü ile yazılmıştır, sizde döngü sembolüyle farklı bir akış şeması oluşturabilirsiniz

### BİR DEN ÇOK FARKLI DURUM VEYA DÖNGÜ İÇEREN ALGORİTMİK YAPILAR

"Ancak ve ancak" bağlacı, iki koşulun aynı anda doğru ya da aynı anda yanlış olduğu durumda doğru sonucu verir.

#### Örnek...5 :

Bir okul öğrencilerinden aşağıdaki koşulları sağlayanlara üstün başarılı öğrenci belgesi vermeyi kararlaştırmıştır.

Koşullar:

‡ Not ortalaması 85 veya üzeri olmalı (N)

‡ Özürsüz devamsızlığı 3 günden az olmalıdır (Ö)

‡ Türkçeyi düzgün kullanmalı (T)

‡ Davranış notu 95 veya üzeri olmalı (A)

‡ Sosyal etkinliklere katılmış olmalı (S)

Yukarıdaki şartların tümünü sağlayan öğrenci üstün başarılı öğrenci belgesi almaya hak kazanmıştır.

Belirtilen koşullar ve belge alma durumunu belirleyen

a) ancak ve ancak bağlacı kullanan

b) niceleyiciler kullanan

önergeler yazınız.

c) belge verilecek öğrencileri belirleyen algoritmayı doğal dil ile yazınız.

a) "Üstün başarılı öğrenci belgesi ver"  $\iff N \geq 85 \wedge \bar{O} < 3 \wedge T = 1 \wedge A \geq 95 \wedge S > 0$

b) Tüm koşulları sağlayan bir öğrenci üstün başarılı belgesi alır.  
Bazı koşulları sağlamayan bir öğrenci üstün başarılı belgesi alamaz.

c)

1.adım: Başla

2.adım: girdilerin alınması

Not Ortalaması (N), Özürsüz devamsızlık sayısı (Ö), Türkçeyi düzgün kullanma (T), Davranış Notu (D), Sosyal etkinlik katılım durumu (S) verilerini al.

3.adım: Eğer not ortalaması 85 den büyük ya da eşitse 4. adıma değilse 9.adıma git

4.adım: Eğer özürsüz devamsızlığı 3 den küçükse 5. adıma değilse 9.adıma git

5.adım: Eğer Türkçeyi düzgün kullanıyorsa 6. adıma değilse 9.adıma git

6.adım: Eğer davranış notu 95 ve üzeriyse 7. adıma değilse 9.adıma git

7.adım: Eğer sosyal etkinliklere katılmışsa 8. adıma değilse 9.adıma git

8.adım: Ekranaya ' Öğrenci üstün başarılı belgesi kazanmıştır' yaz ve 10.adıma git

9.adım: Ekranaya ' Öğrenci üstün başarılı belgesi kazanmamıştır' yaz ve 10. adıma git

10.adım: Bitir

#### Örnek...6 :

Bilgi: Bir fonksiyonun sıfırının iki değer arasında kaldığı bilindiğinde bu iki değer arasında alınmış fonksiyonun sıfırına adım adım yaklaşılarak fonksiyonun sıfırının bulunması yöntemine "ortalama alarak yineleme yöntemi" adı verilir.

Bu yöntemde bir fonksiyonun sıfırının bulunması amacıyla fonksiyonun sıfırının arasında olduğu iki tahmini sayı alınır ve bu sayıların ortalaması ile yeni bir tahmin oluşturulur. Bu yeni tahmin oluşturma işlemi fonksiyonun sıfırının tahmin değeri sıfır olana kadar devam ettirilir.

Örneğin  $f(x)=3x-4$  fonksiyonunu düşünelim. Bu fonksiyonun sıfırını  $3x-4=0$  denklemini sağlayan köktür.

$f(1)=-1<0$  ve  $f(2)=2>0$  olduğundan (ve fonksiyon el kaldırmadan çizilebildiğinden yani sürekli olduğundan) mutlaka  $(1,2)$  aralığında bu fonksiyonun kökü vardır deriz.

**Kritik olan görüntülerin zıt işaretli olmasıdır.** Eğer görüntü sıfır ise kök bulunmuştur.

Elde edilen aralığın orta noktası olan  $\frac{1+2}{2}=\frac{3}{2}$

ile devam edelim  $f\left(\frac{3}{2}\right)=\frac{9}{2}-4=\frac{1}{2}>0$  olduğundan

kökün  $\left(1,\frac{3}{2}\right)$  aralığında olduğunu söyler ve bu şekilde devam ederiz.

Gerçek sayılarda tanımlı  $f(x) = ax + b$  doğrusal fonksiyonunun sıfırını ortalama alarak yineleme yöntemi ile bulunması işlemini algoritmik doğal dil ile ifade ediniz.

1. adım: Başla

2. adım: a, b katsayılarını ve başlangıç için iki değer  $(f(x_1) \cdot f(x_2) < 0$  olacak şekilde)  $(x_1, x_2)$  gir

3. adım:  $f(x) = a \cdot x + b$  fonksiyonunu tanımla

4. adım: Orta noktayı hesapla  
 $x_0 = (x_1 + x_2) / 2$

5. adım:  $f(x_0)$  değerini hesapla  $f(x_0)=0$  ise 8.adıma değilse 6.adıma git

6. adım:

Eğer  $f(x_1) \cdot f(x_0) < 0$  ise  
 $x_2 = x_0$  yap

Değilse

$x_1 = x_0$  yap

7. adım:

Eğer  $f(x_0)$  yaklaşık olarak sıfıra (burada diyelim ki  $|f(x_0)| < 0,000001$  yazabiliriz, bu sayı değerinin 0 yaklaştığını test için keyfi olarak belirlendi, daha küçük sayılar hata oranını daha da azaltacaktır.) 8.adıma değilse 4. adıma git

8. adım:

Kökü  $x_0$  olarak yazdır

9. adım: Bitir

**Örnek...7 :**

Bilgi: sayısal algoritmalar kullanılarak kareköklü ifadelerin yaklaşık değerlerini hesaplama işlemi, ilk kez Babiller tarafından bulunmuştur  
Babil yaklaşık karekök hesaplama metodu:

- Gerçek kök değerine mümkün olduğunca yakın başlangıç tahmini yapılır.  
 $a_n$  tahmini değer  
 $a_0 > 0$  başlangıç değeri  
 $x$  karekökü alınacak sayı olmak üzere

$$a_{n+1} = \frac{1}{2} \cdot \left( a_n + \frac{x}{a_n} \right), n=0,1,2,3, \dots$$

formülü kullanılır ve belirlenen bir hata payına kadar tekrarlanarak bulunur.

Buna göre

a)  $\sqrt{150}$  sayısının yaklaşık değerini Babil metodu kullanarak bulunuz.

b) Babil metodu kullanılarak  $\sqrt{x}$  sayısının yaklaşık değerini bulan algoritmayı doğal dille yazınız.

a)  $\sqrt{150}$  için başlangıç değeri  $a_0 = 12$  olsun

$$a_1 = \frac{1}{2} \left( a_0 + \frac{150}{a_0} \right) = \frac{1}{2} \left( 12 + \frac{150}{12} \right) = 12,25$$

$$a_2 = \frac{1}{2} \left( a_1 + \frac{150}{a_1} \right) = \frac{1}{2} \left( 12,25 + \frac{150}{12,25} \right) = 12,24745..$$

$$a_3 = 12,24745..$$

$a_2$  ve  $a_3$  neredeyse aynı olduğundan

$$\sqrt{150} \approx 12,247 \text{ yazabiliriz.}$$

b)

1. adım: Başla

2. adım:

Karekökü alınacak pozitif sayı  $x$  ve başlangıç tahmini  $a_0 > 0$  değerlerini gir

3. adım:

Hata payı  $E$  değerini belirle (Yaklaşık değer ile gerçek değer farkının ne kadar yakın olmasını istiyorsak, örneğin gerçek değer ile yaklaşık değer farkı 0,0000001 den küçük olsun diyebiliriz)

4. adım:

Yeni tahmini hesapla:

$$a_{n+1} = \frac{1}{2} \cdot \left( a_n + \frac{x}{a_n} \right)$$

5. adım:

$|a_{n+1} - a_n| < E$  koşulunu kontrol et

6. adım:

Eğer koşul sağlanmıyorsa

$$a_n = a_{n+1} \text{ yap ve 4. adıma git}$$

Eğer koşul sağlanıyorsa

sonucu kabul et

7. adım:

$a_{n+1}$  değerini  $\sqrt{x}$ 'in yaklaşık değeri olarak yazdır

8. adım: Bitir

**Örnek...8 :**

Tam kare ifadelerden yararlanarak irrasyonel sayının yaklaşık değerini bulma yöntemi:  
 $a$  bir irrasyonel sayı  $b$ ,  $a$ 'ya en yakın tam kare sayı olmak üzere,  $a$  irrasyonel sayısının yaklaşık değeri

$$\frac{a+b}{2\sqrt{b}} \approx \sqrt{a} \text{ ile bulunur.}$$

Buna göre ,

a)  $\sqrt{150}$  sayısının yaklaşık değerini tam kare ifadelerden yararlanarak irrasyonel sayının yaklaşık değerini bulma yöntemi ile bulunuz.

b)  $\sqrt{x}$  sayısının yaklaşık değerini bulan algoritmayı doğal dille yazınız.

$$a) \quad 12^2 = 144 \quad 13^2 = 169$$

$$b = 144 \quad \sqrt{b} = 12$$

$$\sqrt{150} \approx \frac{150+144}{2 \cdot \sqrt{144}} = \frac{294}{24} = 12,25$$

$$\sqrt{150} \approx 12,25$$

b)

1. adım: Başla

2. adım:

Karekökü alınacak irrasyonel sayı  $x$  değerini gir

3. adım:

$x$ 'e en yakın tam kare sayı  $b$  değerini belirle

4. adım:

Yaklaşık karekök değerini hesapla:

$$\sqrt{x} \approx (x + b) / (2\sqrt{b})$$

5. adım:

Hesaplanan yaklaşık değeri yazdır

7. adım: Bitir

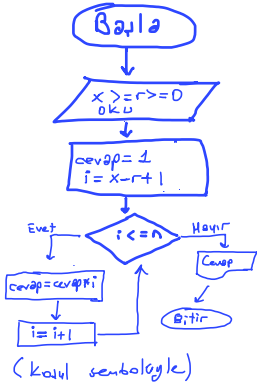
**Örnek...9 :**

$x$  kişiden oluşan bir gruptaki kişilerin yan yana kaç farklı şekilde sıralanabileceğini hesaplayan algoritmanın işleyişini

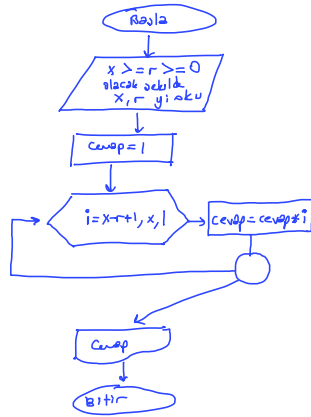
- iki farklı akış şeması (koşul ve döngü)
- algoritmik doğal dil
- sözde kod ile ifade ediniz.

a)  $x \geq r \geq 0$  koşuluyla  $x$  kişiden  $r$  tanesinin sıralanması veya algoritmanın akış şemaları

Koşul ile



Döngü ile



b)  $P(x,r)$  Hesaplama algoritmik doğal dil ile

1.adım Başla

2.adım

$n \geq r \geq 0$  olacak şekilde  $n$  ve  $r$  doğal sayılarını kullanıcıdan al.

3.adım Cevap değişkenine 1 değerini ata.

4.adım  $i = n - r + 1$ 'den başlayarak  $n$ 'e kadar (1'er 1'er artarak) işlemi yap:  
Cevap = Cevap \*  $i$

5.adım Hesaplanan Cevap değerini ekrana yaz.

6.adım Bitir

c)  $P(x,r)$  sözde kod ile

Girdi  $x \geq r \geq 0$  olmak üzere  $x$  ve  $r$  doğal sayılarını gir

Çıktı :  $x$  nesnenin  $r$  tanesinin sıralama sayısı

Başla

cevap=1

Döngü  $i=x - r+1$  den  $x$  e kadar ve dahil (adım +1)

cevap= cevap\*i

döngü sonu

Yazdır cevap

Bitir

**Örnek...10 :**

Verilen bir doğal sayının kaç basamaklı olduğunu bulan algoritmanın işleyişinin akış şemasını çizip doğal dil ile yazınız

**Algoritmanın Doğal Dil ile İşleyişi**

1.adım : Başla

2.adım : Kullanıcıdan bir doğal sayı alınır.

3.adım : Basamak sayısını tutmak için bir  $k$  değişkenini oluştur ve başlangıçta  $k=0$  ata.

4.adım : Aşağıdaki işlemleri tekrarla

Sayıyı 10'a böl ve tam kısmı al sıfırdan farklı her tam kısım için  $k$  sayısını 1 arttır

Döngüde bölüm sonrası sayının tam kısmı 0 olduğunda işlemi sonlandır ve 5.adıma git

5.adım : Elde edilen basamak sayısı ( $k$ ) ekrana yazdır ve 6.adıma git.

6.adım : Bitir

