



# **Bölüm 1: Algoritmalar**

## **JAVA ile Nesne Yönelimli Programlama**



# Önce Algoritma, Sonra Uygulama

- **Adım 1** - Sorunu net bir şekilde tanımlamak, başarının anahtarıdır.
- **Adım 2** - Bir algoritma, sorunu çözmek için adım adım yönergeler sunar.
  - Doğru algoritmayı seçmek, programın verimli çalışmasını sağlar.
- **Adım 3** - Algoritma belirlendikten sonra, seçilen programlama dilinde kodlama gerçekleştirilir.
- Test senaryoları oluşturularak, program doğrulanır, hatalar belirlenir.



# Sorunu Tanımlama

- Programın girdi ve çıktıları doğal bir dilde açıklanır.
- Programın doğru çalıştığını doğrulamak için test durumları oluşturulur.
- **Sorun Tanımı:** Bir kullanıcının girdiği iki sayının toplamını hesaplamak.
  - Test Senaryosu 1: Girdi : 3, 5 - Beklenen Çıktı : 8
  - Test Senaryosu 2: Girdi : -2, 7 - Beklenen Çıktı : 5
- **Sorun Tanımı:** Bir metin belgesinin içindeki kelime sayısını bulmak.
  - Test Senaryosu 1: Girdi : "Merhaba, dünya!" - Beklenen Çıktı : 2
  - Test Senaryosu 2: Girdi : "Bu bir denemedir." - Beklenen Çıktı : 3



# Algoritma Belirleme

- Sorunu mantıklı bir şekilde çözmek için gerekli adımlar belirlenir.
- **Sorun:** İki sayının toplamını hesaplama.
- **Algoritma:**
  - İlk sayıyı al.
  - İkinci sayıyı al.
  - İki sayıyı topla.
  - Toplamı ekrana yazdır.



# Algoritma Belirleme

- Algoritma oluşturulduktan sonraki adım test etmektir.
- Küçük ama anlamlı test senaryoları oluşturulur.
- Algoritma adım adım çalıştırılır ve her adımın sonucu yazılır.



# Koda Dönüştürme (Implementation)

- Algoritmayı oluşturduktan sonraki adım kodlamadır.
- Programlama dili, algoritmayı bilgisayarın anlayacağı şekilde ifade eder.
- Bir programı yönetilebilir ve anlaşılır hale getirmek için,
  - Kod mantıksal parçalara ayrılır.



# Örnek Kod Parçacıkları

```
public class SumCalculator {  
    public static void main(String[] args) {  
        // Create a Scanner object to read user input  
        Scanner scanner = new Scanner(System.in);  
        // Prompt the user to enter the number  
        System.out.print("Enter the first number: ");  
        double num1 = scanner.nextDouble();  
        System.out.print("Enter the second number: ");  
        double num2 = scanner.nextDouble();  
        // Calculate the sum of the two numbers  
        double sum = num1 + num2;  
        // Display the result  
        System.out.println("The sum of " + num1 + " and " + num2 + " is: " + sum);  
        // Close the Scanner  
        scanner.close();  
    }  
}
```



# Örnek Kod Parçacıkları

```
public class ToplamaHesaplama {  
    public static void main(String[] args) {  
        // Kullanıcı girdisini okumak için bir Scanner nesnesi oluşturun  
        Scanner tarayici = new Scanner(System.in);  
        // Kullanıcıdan sayıyı girmesini isteyin  
        System.out.print("Birinci sayıyı giriniz: ");  
        double sayi1 = tarayici.nextDouble();  
        System.out.print("İkinci sayıyı giriniz: ");  
        double sayi2 = tarayici.nextDouble();  
        // İki sayının toplamını hesaplayın  
        double toplam = sayi1 + sayi2;  
        // Sonucu görüntüleyin  
        System.out.println(sayi1 + " ile " + sayi2 + " toplamı: " + toplam);  
        // Tarayıcıyı kapatın  
        tarayici.close();  
    }  
}
```





# Fonksiyonlar

- Programları daha anlaşılır, sürdürülebilir ve verimli hale getirir.
- Bir grup ifadeyi (*statements*) adlandırma fırsatı sunar.
- Tekrarlı kodları ortadan kaldırarak programı küçük ve düzenli hale getirir.
- Uzun bir programı fonksiyonlara bölmek,
  - her parçanın ayrı ayrı test edilebilmesine ve
  - sonrasında çalışabilir bir bütün haline getirmeye olanak tanır.
- İyi tasarlanmış fonksiyonlar, başka projelerde yeniden kullanılabilir.



# Örnek Algoritmalar

- İki Tamsayıyı Küçükten Büyüğe Doğru Yazdırma
- Üç Tamsayıyı Küçükten Büyüğe Doğru Yazdırma
- Faktöriyel Bulma
- *Fibonacci* Serisi Bulma
- *Palindrom* Kontrolü
- Asal Sayı Kontrolü
- Üs Alma



# İki Tamsayıyı Küçükten Büyüğe Doğru Yazdırma

Başla

```
Yaz "Birinci tamsayıyı girin: "
```

```
Oku birinciTamsayı
```

```
Yaz "İkinci tamsayıyı girin: "
```

```
Oku ikinciTamsayı
```

```
Eğer birinciTamsayı < ikinciTamsayı ise
```

```
    Yaz birinciTamsayı, " ", ikinciTamsayı
```

```
Değilse
```

```
    Yaz ikinciTamsayı, " ", birinciTamsayı
```

Bitir



# İki Tamsayıyı Küçükten Büyüğe Doğru Yazdırma

Başla

Yaz "Birinci tamsayıyı girin: "

Oku **birinciTamsayı**

Yaz "İkinci tamsayıyı girin: "

Oku **ikinciTamsayı**

birinci	ikinci
4	

Eğer **birinciTamsayı < ikinciTamsayı** ise

Yaz **birinciTamsayı, " ", ikinciTamsayı**

Değilse

Yaz **ikinciTamsayı, " ", birinciTamsayı**

Bitir



# İki Tamsayıyı Küçükten Büyüğe Doğru Yazdırma

Başla

Yaz "Birinci tamsayıyı girin: "  
Oku birinciTamsayı

Yaz "İkinci tamsayıyı girin: "  
**Oku ikinciTamsayı**

birinci	ikinci
4	11

Eğer birinciTamsayı < ikinciTamsayı ise  
Yaz birinciTamsayı, " ", ikinciTamsayı  
Değilse  
Yaz ikinciTamsayı, " ", birinciTamsayı

Bitir



# İki Tamsayıyı Küçükten Büyüğe Doğru Yazdırma

Başla

Yaz "Birinci tamsayıyı girin: "  
Oku birinciTamsayı

Yaz "İkinci tamsayıyı girin: "  
Oku ikinciTamsayı

birinci	ikinci
4	11

**Eğer birinciTamsayı < ikinciTamsayı ise**

Yaz birinciTamsayı, " ", ikinciTamsayı

Değilse

Yaz ikinciTamsayı, " ", birinciTamsayı

Bitir



# İki Tamsayıyı Küçükten Büyüğe Doğru Yazdırma

Başla

Yaz "Birinci tamsayıyı girin: "  
Oku birinciTamsayı

Yaz "İkinci tamsayıyı girin: "  
Oku ikinciTamsayı

birinci	ikinci
4	11
4 11	

Eğer birinciTamsayı < ikinciTamsayı ise  
Yaz birinciTamsayı, " ", ikinciTamsayı  
Değilse  
Yaz ikinciTamsayı, " ", birinciTamsayı

Bitir



# İki Tamsayıyı Küçükten Büyüğe Doğru Yazdırma

Başla

Yaz "Birinci tamsayıyı girin: "

Oku **birinciTamsayı**

Yaz "İkinci tamsayıyı girin: "

Oku **ikinciTamsayı**

birinci	ikinci
13	

Eğer **birinciTamsayı < ikinciTamsayı** ise

Yaz **birinciTamsayı, " ", ikinciTamsayı**

Değilse

Yaz **ikinciTamsayı, " ", birinciTamsayı**

Bitir





# İki Tamsayıyı Küçükten Büyüğe Doğru Yazdırma

Başla

Yaz "Birinci tamsayıyı girin: "  
Oku birinciTamsayı

Yaz "İkinci tamsayıyı girin: "  
**Oku ikinciTamsayı**

birinci	ikinci
13	11

Eğer birinciTamsayı < ikinciTamsayı ise  
Yaz birinciTamsayı, " ", ikinciTamsayı  
Değilse  
Yaz ikinciTamsayı, " ", birinciTamsayı

Bitir



# İki Tamsayıyı Küçükten Büyüğe Doğru Yazdırma

Başla

Yaz "Birinci tamsayıyı girin: "  
Oku birinciTamsayı

Yaz "İkinci tamsayıyı girin: "  
Oku ikinciTamsayı

birinci	ikinci
13	11

**Eğer birinciTamsayı < ikinciTamsayı ise**

Yaz birinciTamsayı, " ", ikinciTamsayı

Değilse

Yaz ikinciTamsayı, " ", birinciTamsayı

Bitir



# İki Tamsayıyı Küçükten Büyüğe Doğru Yazdırma

Başla

Yaz "Birinci tamsayıyı girin: "  
Oku birinciTamsayı

Yaz "İkinci tamsayıyı girin: "  
Oku ikinciTamsayı

birinci	ikinci
13	11

Eğer birinciTamsayı < ikinciTamsayı ise  
Yaz birinciTamsayı, " ", ikinciTamsayı

**Değilse**

Yaz ikinciTamsayı, " ", birinciTamsayı

Bitir



# İki Tamsayıyı Küçükten Büyüğe Doğru Yazdırma

Başla

Yaz "Birinci tamsayıyı girin: "  
Oku birinciTamsayı

Yaz "İkinci tamsayıyı girin: "  
Oku ikinciTamsayı

birinci	ikinci
13	11
11 13	

Eğer birinciTamsayı < ikinciTamsayı ise  
Yaz birinciTamsayı, " ", ikinciTamsayı  
Değilse  
**Yaz ikinciTamsayı, " ", birinciTamsayı**

Bitir



# Örnek Algoritmalar

- İki Tamsayıyı Küçükten Büyüğe Doğru Yazdırma
- **Üç Tamsayıyı Küçükten Büyüğe Doğru Yazdırma**
- Faktöriyel Bulma
- Fibonacci Serisi Bulma
- Palindrom Kontrolü
- Asal Sayı Kontrolü
- Üs Alma



# Üç Tamsayıyı Küçükten Büyüğe Doğru Yazdırma

Başla

```
// İlk tamsayıyı kullanıcıdan al  
Yaz "Birinci tamsayıyı girin: "  
Oku birinciTamsayı
```

```
// İkinci tamsayıyı kullanıcıdan al  
Yaz "İkinci tamsayıyı girin: "  
Oku ikinciTamsayı
```

```
// Üçüncü tamsayıyı kullanıcıdan al  
Yaz "Üçüncü tamsayıyı girin: "  
Oku üçüncüTamsayı
```



# Üç Tamsayıyı Küçükten Büyüğe Doğru Yazdırma

```
// Tamsayıları küçükten büyüğe sırala ve yazdır
Eğer birinciTamsayı <= ikinciTamsayı ve
                               birinciTamsayı <= üçüncüTamsayı ise
    Yaz birinciTamsayı
    Eğer ikinciTamsayı <= üçüncüTamsayı ise
        Yaz ikinciTamsayı, " ", üçüncüTamsayı
    Değilse
        Yaz üçüncüTamsayı, " ", ikinciTamsayı
    End Eğer
```



# Üç Tamsayıyı Küçükten Büyüğe Doğru Yazdırma

```
Değilse Eğer ikinciTamsayı <= birinciTamsayı ve  
                ikinciTamsayı <= üçüncüTamsayı ise
```

```
Yaz ikinciTamsayı
```

```
Eğer birinciTamsayı <= üçüncüTamsayı ise
```

```
    Yaz birinciTamsayı, " ", üçüncüTamsayı
```

```
Değilse
```

```
    Yaz üçüncüTamsayı, " ", birinciTamsayı
```

```
End Eğer
```





# Üç Tamsayıyı Küçükten Büyüğe Doğru Yazdırma

Değilse

Yaz üçüncüTamsayı

Eğer birinciTamsayı  $\leq$  ikinciTamsayı ise

Yaz birinciTamsayı, " ", ikinciTamsayı

Değilse

Yaz ikinciTamsayı, " ", birinciTamsayı

End Eğer

End Eğer

Bitir



# Örnek Algoritmalar

- İki Tamsayıyı Küçükten Büyüğe Doğru Yazdırma
- Üç Tamsayıyı Küçükten Büyüğe Doğru Yazdırma
- **Faktöriyel Bulma**
- Fibonacci Serisi Bulma
- Palindrom Kontrolü
- Asal Sayı Kontrolü
- Üs Alma



# Faktöriyel Bulma

Başla

```
// Bir sayıyı girin
```

```
Input: sayı
```

```
// Sonucu saklamak için bir değişkeni başlatın  
faktoriyel = 1
```

```
// Faktöriyel hesabı için döngü  
Döngü i = 1'den başlayarak sayı'ya kadar  
    faktoriyel = faktoriyel * i  
Döngüyü Bitir
```

```
Yaz "Girilen sayının faktöriyeli: ", faktoriyel
```

Bitir



# Faktöriyel Bulma

Başla

```
// Bir sayıyı girin
```

**Input: sayı**

sayı	faktoriyel	i
5		

```
// Sonucu saklamak için bir değişkeni başlatın  
faktoriyel = 1
```

```
// Faktöriyel hesabı için döngü  
Döngü i = 1'den başlayarak sayı'ya kadar  
    faktoriyel = faktoriyel * i  
Döngüyü Bitir
```

```
Yaz "Girilen sayının faktöriyeli: ", faktoriyel
```

Bitir



# Faktöriyel Bulma

Başla

```
// Bir sayıyı girin
```

```
Input: sayı
```

sayı	faktoriyel	i
5	1	

```
// Sonucu saklamak için bir değişkeni başlatın
```

```
faktoriyel = 1
```

```
// Faktöriyel hesabı için döngü
```

```
Döngü i = 1'den başlayarak sayı'ya kadar
```

```
    faktoriyel = faktoriyel * i
```

```
Döngüyü Bitir
```

```
Yaz "Girilen sayının faktöriyeli: ", faktoriyel
```

Bitir



# Faktöriyel Bulma

Başla

```
// Bir sayıyı girin
```

```
Input: sayı
```

sayı	faktoriyel	i
5	1	1

```
// Sonucu saklamak için bir değişkeni başlatın  
faktoriyel = 1
```

```
// Faktöriyel hesabı için döngü
```

**Döngü i = 1'den başlayarak sayı'ya kadar**

```
    faktoriyel = faktoriyel * i
```

```
Döngüyü Bitir
```

```
Yaz "Girilen sayının faktöriyeli: ", faktoriyel
```

Bitir



# Faktöriyel Bulma

Başla

```
// Bir sayıyı girin
```

```
Input: sayı
```

sayı	faktoriyel	i
5	1	1

```
// Sonucu saklamak için bir değişkeni başlatın  
faktoriyel = 1
```

```
// Faktöriyel hesabı için döngü  
Döngü i = 1'den başlayarak sayı'ya kadar  
    faktoriyel = faktoriyel * i  
Döngüyü Bitir
```

```
Yaz "Girilen sayının faktöriyeli: ", faktoriyel
```

Bitir



# Faktöriyel Bulma

Başla

```
// Bir sayıyı girin
```

```
Input: sayı
```

sayı	faktoriyel	i
5	1	2

```
// Sonucu saklamak için bir değişkeni başlatın  
faktoriyel = 1
```

```
// Faktöriyel hesabı için döngü
```

**Döngü i = 1'den başlayarak sayı'ya kadar**

```
    faktoriyel = faktoriyel * i
```

```
Döngüyü Bitir
```

```
Yaz "Girilen sayının faktöriyeli: ", faktoriyel
```

Bitir





# Faktöriyel Bulma

Başla

```
// Bir sayıyı girin
```

```
Input: sayı
```

sayı	faktoriyel	i
5	2	2

```
// Sonucu saklamak için bir değişkeni başlatın  
faktoriyel = 1
```

```
// Faktöriyel hesabı için döngü  
Döngü i = 1'den başlayarak sayı'ya kadar  
    faktoriyel = faktoriyel * i  
Döngüyü Bitir
```

```
Yaz "Girilen sayının faktöriyeli: ", faktoriyel
```

Bitir



# Faktöriyel Bulma

Başla

```
// Bir sayıyı girin
```

```
Input: sayı
```

sayı	faktoriyel	i
5	2	3

```
// Sonucu saklamak için bir değişkeni başlatın  
faktoriyel = 1
```

```
// Faktöriyel hesabı için döngü
```

**Döngü i = 1'den başlayarak sayı'ya kadar**

```
    faktoriyel = faktoriyel * i
```

```
Döngüyü Bitir
```

```
Yaz "Girilen sayının faktöriyeli: ", faktoriyel
```

Bitir



# Faktöriyel Bulma

Başla

```
// Bir sayıyı girin
```

```
Input: sayı
```

sayı	faktoriyel	i
5	6	3

```
// Sonucu saklamak için bir değişkeni başlatın  
faktoriyel = 1
```

```
// Faktöriyel hesabı için döngü  
Döngü i = 1'den başlayarak sayı'ya kadar  
    faktoriyel = faktoriyel * i  
Döngüyü Bitir
```

```
Yaz "Girilen sayının faktöriyeli: ", faktoriyel
```

Bitir



# Faktöriyel Bulma

Başla

```
// Bir sayıyı girin
```

```
Input: sayı
```

sayı	faktoriyel	i
5	6	4

```
// Sonucu saklamak için bir değişkeni başlatın  
faktoriyel = 1
```

```
// Faktöriyel hesabı için döngü
```

**Döngü i = 1'den başlayarak sayı'ya kadar**

```
    faktoriyel = faktoriyel * i
```

```
Döngüyü Bitir
```

```
Yaz "Girilen sayının faktöriyeli: ", faktoriyel
```

Bitir



# Faktöriyel Bulma

Başla

```
// Bir sayıyı girin
```

```
Input: sayı
```

sayı	faktoriyel	i
5	24	4

```
// Sonucu saklamak için bir değişkeni başlatın  
faktoriyel = 1
```

```
// Faktöriyel hesabı için döngü  
Döngü i = 1'den başlayarak sayı'ya kadar  
    faktoriyel = faktoriyel * i  
Döngüyü Bitir
```

```
Yaz "Girilen sayının faktöriyeli: ", faktoriyel
```

Bitir



# Faktöriyel Bulma

Başla

```
// Bir sayıyı girin
```

```
Input: sayı
```

sayı	faktoriyel	i
5	24	5

```
// Sonucu saklamak için bir değişkeni başlatın  
faktoriyel = 1
```

```
// Faktöriyel hesabı için döngü
```

**Döngü i = 1'den başlayarak sayı'ya kadar**

```
    faktoriyel = faktoriyel * i
```

```
Döngüyü Bitir
```

```
Yaz "Girilen sayının faktöriyeli: ", faktoriyel
```

Bitir



# Faktöriyel Bulma

Başla

```
// Bir sayıyı girin
```

```
Input: sayı
```

sayı	faktoriyel	i
5	120	5

```
// Sonucu saklamak için bir değişkeni başlatın  
faktoriyel = 1
```

```
// Faktöriyel hesabı için döngü  
Döngü i = 1'den başlayarak sayı'ya kadar  
    faktoriyel = faktoriyel * i  
Döngüyü Bitir
```

```
Yaz "Girilen sayının faktöriyeli: ", faktoriyel
```

Bitir



# Faktöriyel Bulma

Başla

```
// Bir sayıyı girin
```

```
Input: sayı
```

sayı	faktoriyel	i
5	120	5

```
// Sonucu saklamak için bir değişkeni başlatın  
faktoriyel = 1
```

```
// Faktöriyel hesabı için döngü  
Döngü i = 1'den başlayarak sayı'ya kadar  
    faktoriyel = faktoriyel * i
```

**Döngüyü Bitir**

```
Yaz "Girilen sayının faktöriyeli: ", faktoriyel
```

Bitir





# Örnek Algoritmalar

- İki Tamsayıyı Küçükten Büyüğe Doğru Yazdırma
- Üç Tamsayıyı Küçükten Büyüğe Doğru Yazdırma
- Faktöriyel Bulma
- **Fibonacci Serisi Bulma**
- Palindrom Kontrolü
- Asal Sayı Kontrolü
- Üs Alma



# Fibonacci Serisi Bulma

Başla

```
// Bir sayıyı girin
```

```
Input: n
```

```
// İlk iki Fibonacci sayısını başlatın
```

```
a = 0
```

```
b = 1
```

```
// Sonucu saklamak için bir değişkeni başlatın
```

```
sonuc = 0
```

```
Eğer n = 0 ise
```

```
Yaz 0
```

```
Değilse
```



# Fibonacci Serisi Bulma

Değilse

// İlk n Fibonacci sayısını hesaplayın

Yaz a

Yaz b

Döngü i = 2'den başlayarak n - 1'e kadar

sonuc = a + b

a = b

b = sonuc

Yaz sonuc

Döngüyü Bitir

End Eğer

Bitir



# Fibonacci Serisi Bulma

Değilse

// İlk n Fibonacci sayısını hesaplayın

Yaz a

Yaz b

Döngü i = 2'den başlayarak n - 1'e kadar

sonuc = a + b

a = b

b = sonuc

Yaz sonuc

Döngüyü Bitir

End Eğer

n	a	b	i	sonuc
5	0	1		

Bitir



# Fibonacci Serisi Bulma

Değilse

// İlk n Fibonacci sayısını hesaplayın

**Yaz a**

Yaz b

Döngü i = 2'den başlayarak n - 1'e kadar

sonuc = a + b

a = b

b = sonuc

Yaz sonuc

Döngüyü Bitir

End Eğer

n	a	b	i	sonuc
5	0	1		
0				

Bitir



# Fibonacci Serisi Bulma

Değilse

// İlk n Fibonacci sayısını hesaplayın

Yaz a

**Yaz b**

Döngü i = 2'den başlayarak n - 1'e kadar

sonuc = a + b

a = b

b = sonuc

Yaz sonuc

Döngüyü Bitir

End Eğer

n	a	b	i	sonuc
<b>5</b>	<b>0</b>	<b>1</b>		
<b>0 1</b>				

Bitir



# Fibonacci Serisi Bulma

Değilse

// İlk n Fibonacci sayısını hesaplayın

Yaz a

Yaz b

**Döngü i = 2'den başlayarak n - 1'e kadar**

sonuc = a + b

a = b

b = sonuc

Yaz sonuc

Döngüyü Bitir

End Eğer

n	a	b	i	sonuc
5	0	1	2	
0 1				

Bitir



# Fibonacci Serisi Bulma

Değilse

// İlk n Fibonacci sayısını hesaplayın

Yaz a

Yaz b

Döngü i = 2'den başlayarak n - 1'e kadar

**sonuc = a + b**

a = b

b = sonuc

Yaz sonuc

Döngüyü Bitir

End Eğer

n	a	b	i	sonuc
5	0	1	2	1
0 1				

Bitir





# Fibonacci Serisi Bulma

Değilse

// İlk n Fibonacci sayısını hesaplayın

Yaz a

Yaz b

Döngü i = 2'den başlayarak n - 1'e kadar

sonuc = a + b

**a = b**

b = sonuc

Yaz sonuc

Döngüyü Bitir

End Eğer

n	a	b	i	sonuc
5	1	1	2	1
0 1				

Bitir



# Fibonacci Serisi Bulma

Değilse

```
// İlk n Fibonacci sayısını hesaplayın
```

```
Yaz a
```

```
Yaz b
```

```
Döngü i = 2'den başlayarak n - 1'e kadar
```

```
    sonuc = a + b
```

```
    a = b
```

```
    b = sonuc
```

```
    Yaz sonuc
```

```
Döngüyü Bitir
```

```
End Eğer
```

n	a	b	i	sonuc
5	1	1	2	1
0 1				

Bitir



# Fibonacci Serisi Bulma

Değilse

// İlk n Fibonacci sayısını hesaplayın

Yaz a

Yaz b

Döngü i = 2'den başlayarak n - 1'e kadar

sonuc = a + b

a = b

b = sonuc

**Yaz sonuc**

Döngüyü Bitir

End Eğer

n	a	b	i	sonuc
5	1	1	2	1
0 1 1				

Bitir



# Fibonacci Serisi Bulma

Değilse

// İlk n Fibonacci sayısını hesaplayın

Yaz a

Yaz b

**Döngü i = 2'den başlayarak n - 1'e kadar**

sonuc = a + b

a = b

b = sonuc

Yaz sonuc

Döngüyü Bitir

End Eğer

n	a	b	i	sonuc
5	1	1	3	1
0 1 1				

Bitir



# Fibonacci Serisi Bulma

Değilse

// İlk n Fibonacci sayısını hesaplayın

Yaz a

Yaz b

Döngü i = 2'den başlayarak n - 1'e kadar

**sonuc = a + b**

a = b

b = sonuc

Yaz sonuc

Döngüyü Bitir

End Eğer

n	a	b	i	sonuc
5	1	1	3	2
0 1 1				

Bitir



# Fibonacci Serisi Bulma

Değilse

// İlk n Fibonacci sayısını hesaplayın

Yaz a

Yaz b

Döngü i = 2'den başlayarak n - 1'e kadar

sonuc = a + b

**a = b**

b = sonuc

Yaz sonuc

Döngüyü Bitir

End Eğer

n	a	b	i	sonuc
5	1	1	3	2
0 1 1				

Bitir



# Fibonacci Serisi Bulma

Değilse

```
// İlk n Fibonacci sayısını hesaplayın
```

```
Yaz a
```

```
Yaz b
```

```
Döngü i = 2'den başlayarak n - 1'e kadar
```

```
    sonuc = a + b
```

```
    a = b
```

```
    b = sonuc
```

```
    Yaz sonuc
```

```
Döngüyü Bitir
```

```
End Eğer
```

n	a	b	i	sonuc
5	1	2	3	2
0 1 1				

Bitir



# Fibonacci Serisi Bulma

Değilse

// İlk n Fibonacci sayısını hesaplayın

Yaz a

Yaz b

Döngü i = 2'den başlayarak n - 1'e kadar

sonuc = a + b

a = b

b = sonuc

**Yaz sonuc**

Döngüyü Bitir

End Eğer

n	a	b	i	sonuc
5	1	2	3	2
0 1 1 2				

Bitir





# Fibonacci Serisi Bulma

Değilse

// İlk n Fibonacci sayısını hesaplayın

Yaz a

Yaz b

**Döngü i = 2'den başlayarak n - 1'e kadar**

sonuc = a + b

a = b

b = sonuc

Yaz sonuc

Döngüyü Bitir

End Eğer

n	a	b	i	sonuc
5	1	2	4	2
0 1 1 2				

Bitir



# Fibonacci Serisi Bulma

Değilse

// İlk n Fibonacci sayısını hesaplayın

Yaz a

Yaz b

Döngü i = 2'den başlayarak n - 1'e kadar

**sonuc = a + b**

a = b

b = sonuc

Yaz sonuc

Döngüyü Bitir

End Eğer

n	a	b	i	sonuc
5	1	2	4	3
0 1 1 2				

Bitir



# Fibonacci Serisi Bulma

Değilse

// İlk n Fibonacci sayısını hesaplayın

Yaz a

Yaz b

Döngü i = 2'den başlayarak n - 1'e kadar

sonuc = a + b

**a = b**

b = sonuc

Yaz sonuc

Döngüyü Bitir

End Eğer

n	a	b	i	sonuc
5	2	2	4	3
0 1 1 2				

Bitir



# Fibonacci Serisi Bulma

Değilse

```
// İlk n Fibonacci sayısını hesaplayın
```

```
Yaz a
```

```
Yaz b
```

```
Döngü i = 2'den başlayarak n - 1'e kadar
```

```
    sonuc = a + b
```

```
    a = b
```

```
    b = sonuc
```

```
    Yaz sonuc
```

```
Döngüyü Bitir
```

```
End Eğer
```

n	a	b	i	sonuc
5	2	3	4	3
0 1 1 2				

Bitir



# Fibonacci Serisi Bulma

Değilse

```
// İlk n Fibonacci sayısını hesaplayın
```

```
Yaz a
```

```
Yaz b
```

```
Döngü i = 2'den başlayarak n - 1'e kadar
```

```
    sonuc = a + b
```

```
    a = b
```

```
    b = sonuc
```

```
    Yaz sonuc
```

```
Döngüyü Bitir
```

```
End Eğer
```

n	a	b	i	sonuc
5	2	3	4	3
0 1 1 2 3				

Bitir



# Fibonacci Serisi Bulma

Değilse

// İlk n Fibonacci sayısını hesaplayın

Yaz a

Yaz b

**Döngü i = 2'den başlayarak n - 1'e kadar**

sonuc = a + b

a = b

b = sonuc

Yaz sonuc

Döngüyü Bitir

End Eğer

n	a	b	i	sonuc
5	2	3	5	3
0 1 1 2 3				

Bitir



# Fibonacci Serisi Bulma

Değilse

// İlk n Fibonacci sayısını hesaplayın

Yaz a

Yaz b

Döngü i = 2'den başlayarak n - 1'e kadar

**sonuc = a + b**

a = b

b = sonuc

Yaz sonuc

Döngüyü Bitir

End Eğer

n	a	b	i	sonuc
<b>5</b>	<b>2</b>	<b>3</b>	<b>5</b>	<b>5</b>
0	1	1	2	3

Bitir



# Fibonacci Serisi Bulma

Değilse

```
// İlk n Fibonacci sayısını hesaplayın
```

```
Yaz a
```

```
Yaz b
```

```
Döngü i = 2'den başlayarak n - 1'e kadar
```

```
    sonuc = a + b
```

```
    a = b
```

```
    b = sonuc
```

```
    Yaz sonuc
```

```
Döngüyü Bitir
```

```
End Eğer
```

n	a	b	i	sonuc
5	3	3	5	5
0 1 1 2 3				

Bitir





# Fibonacci Serisi Bulma

Değilse

```
// İlk n Fibonacci sayısını hesaplayın
```

```
Yaz a
```

```
Yaz b
```

```
Döngü i = 2'den başlayarak n - 1'e kadar
```

```
    sonuc = a + b
```

```
    a = b
```

```
    b = sonuc
```

```
    Yaz sonuc
```

```
Döngüyü Bitir
```

```
End Eğer
```

n	a	b	i	sonuc
5	3	5	5	5
0 1 1 2 3				

Bitir



# Fibonacci Serisi Bulma

Değilse

```
// İlk n Fibonacci sayısını hesaplayın
```

```
Yaz a
```

```
Yaz b
```

```
Döngü i = 2'den başlayarak n - 1'e kadar
```

```
    sonuc = a + b
```

```
    a = b
```

```
    b = sonuc
```

```
    Yaz sonuc
```

```
Döngüyü Bitir
```

```
End Eğer
```

n	a	b	i	sonuc
5	3	5	5	5
0 1 1 2 3 5				

Bitir



# Fibonacci Serisi Bulma

Değilse

```
// İlk n Fibonacci sayısını hesaplayın
```

```
Yaz a
```

```
Yaz b
```

```
Döngü i = 2'den başlayarak n - 1'e kadar
```

```
    sonuc = a + b
```

```
    a = b
```

```
    b = sonuc
```

```
    Yaz sonuc
```

**Döngüyü Bitir**

```
End Eğer
```

n	a	b	i	sonuc
5	3	5	5	5
0 1 1 2 3 5				

Bitir



# Örnek Algoritmalar

- İki Tamsayıyı Küçükten Büyüğe Doğru Yazdırma
- Üç Tamsayıyı Küçükten Büyüğe Doğru Yazdırma
- Faktöriyel Bulma
- Fibonacci Serisi Bulma
- **Palindrom Kontrolü**
- Asal Sayı Kontrolü
- Üs Alma



# Palindrom Kontrolü

Başla

```
// Bir kelime veya cümlenin girilmesini isteyin
```

```
Input: girdiMetin
```

```
// Temizlenmiş girdiyi ters çevirin
```

```
tersGirdi = TersÇevir(girdiMetin)
```

```
// Temizlenmiş girdi ile ters girdiyi karşılaştırın
```

```
Eğer girdiMetin = tersGirdi ise
```

```
    Yaz "Girilen metin bir palindromdur."
```

```
Değilse
```

```
    Yaz "Girilen metin bir palindrom değildir."
```

```
End Eğer
```

Bitir



# Palindrom Kontrolü

Başla

```
// Bir kelime veya cümlenin girilmesini isteyin
```

```
Input: girdiMetin
```

```
// Temizlenmiş girdiyi ters çevirin  
tersGirdi = TersÇevir(girdiMetin)
```

girdiMetin	tersGirdi

```
// Temizlenmiş girdi ile ters girdiyi karşılaştırın
```

```
Eğer girdiMetin = tersGirdi ise
```

```
Yaz "Girilen metin bir palindromdur."
```

```
Değilse
```

```
Yaz "Girilen metin bir palindrom değildir."
```

```
End Eğer
```

Bitir



# Palindrom Kontrolü

Başla

```
// Bir kelime veya cümlenin girilmesini isteyin
```

**Input: girdiMetin**

```
// Temizlenmiş girdiyi ters çevirin  
tersGirdi = TersÇevir(girdiMetin)
```

girdiMetin	tersGirdi
abcba	

```
// Temizlenmiş girdi ile ters girdiyi karşılaştırın
```

```
Eğer girdiMetin = tersGirdi ise
```

```
Yaz "Girilen metin bir palindromdur."
```

```
Değilse
```

```
Yaz "Girilen metin bir palindrom değildir."
```

```
End Eğer
```

Bitir



# Palindrom Kontrolü

Başla

```
// Bir kelime veya cümlenin girilmesini isteyin
```

```
Input: girdiMetin
```

```
// Temizlenmiş girdiyi ters çevirin
```

```
tersGirdi = TersÇevir(girdiMetin)
```

girdiMetin	tersGirdi
<b>abcba</b>	<b>abcba</b>

```
// Temizlenmiş girdi ile ters girdiyi karşılaştırın
```

```
Eğer girdiMetin = tersGirdi ise
```

```
Yaz "Girilen metin bir palindromdur."
```

```
Değilse
```

```
Yaz "Girilen metin bir palindrom değildir."
```

```
End Eğer
```

Bitir





# Palindrom Kontrolü

Başla

```
// Bir kelime veya cümlenin girilmesini isteyin
```

```
Input: girdiMetin
```

```
// Temizlenmiş girdiyi ters çevirin  
tersGirdi = TersÇevir(girdiMetin)
```

girdiMetin	tersGirdi
abcba	abcba

```
// Temizlenmiş girdi ile ters girdiyi karşılaştırın
```

```
Eğer girdiMetin = tersGirdi ise
```

```
    Yaz "Girilen metin bir palindromdur."
```

```
Değilse
```

```
    Yaz "Girilen metin bir palindrom değildir."
```

```
End Eğer
```

Bitir



# Palindrom Kontrolü

Başla

```
// Bir kelime veya cümlenin girilmesini isteyin
```

```
Input: girdiMetin
```

```
// Temizlenmiş girdiyi ters çevirin  
tersGirdi = TersÇevir(girdiMetin)
```

girdiMetin	tersGirdi
abcba	abcba
Girilen metin bir palindromdur	

```
// Temizlenmiş girdi ile ters girdiyi karşılaştırın  
Eğer girdiMetin = tersGirdi ise
```

```
    Yaz "Girilen metin bir palindromdur."
```

```
Değilse
```

```
    Yaz "Girilen metin bir palindrom değildir."
```

```
End Eğer
```

Bitir



# Örnek Algoritmalar

- İki Tamsayıyı Küçükten Büyüğe Doğru Yazdırma
- Üç Tamsayıyı Küçükten Büyüğe Doğru Yazdırma
- Faktöriyel Bulma
- Fibonacci Serisi Bulma
- Palindrom Kontrolü
- **Asal Sayı Kontrolü**
- Üs Alma



# Asal Sayı Kontrolü

Başla

```
// Bir sayıyı girin
```

```
Input: sayı
```

```
// Sayının asal olup olmadığını kontrol etmek için bir bayrak (flag) başlatın  
asalMi = True
```

```
// 2'den başlayarak sayının yarısına kadar olan tüm bölenleri kontrol et
```

```
Döngü bölen = 2'den başlayarak sayı / 2'e kadar
```

```
    Eğer sayı % bölen = 0 ise
```

```
        // Bölünüyorsa, sayı asal değildir
```

```
        asalMi = False
```

```
        Döngüyü Kır
```

```
    End Eğer
```

```
Döngüyü Bitir
```



# Asal Sayı Kontrolü

```
// Bayrağa (flag) göre sonucu yazdırın  
Eğer asalMi = True ise  
    Yaz sayı, " bir asal sayıdır."  
Değilse  
    Yaz sayı, " bir asal sayı değildir."  
End Eğer
```

Bitir



# Asal Sayı Kontrolü

Başla

// Bir sayıyı girin

**Input: sayı**

sayı	asalMi	bölen
7		

// Sayının asal olup olmadığını kontrol etmek için bir bayrak (flag) başlatın  
asalMi = True

// 2'den başlayarak sayının yarısına kadar olan tüm bölenleri kontrol et

Döngü bölen = 2'den başlayarak sayı / 2'e kadar

Eğer sayı % bölen = 0 ise

// Bölünüyorsa, sayı asal değildir

asalMi = False

Döngüyü Kır

End Eğer

Döngüyü Bitir



# Asal Sayı Kontrolü

Başla

```
// Bir sayıyı girin
```

Input: sayı

sayı	asalMi	bölen
7	True	

```
// Sayının asal olup olmadığını kontrol etmek için bir bayrak (flag) başlatın
```

```
asalMi = True
```

```
// 2'den başlayarak sayının yarısına kadar olan tüm bölenleri kontrol et
```

```
Döngü bölen = 2'den başlayarak sayı / 2'e kadar
```

```
    Eğer sayı % bölen = 0 ise
```

```
        // Bölünüyorsa, sayı asal değildir
```

```
        asalMi = False
```

```
        Döngüyü Kır
```

```
    End Eğer
```

```
Döngüyü Bitir
```



# Asal Sayı Kontrolü

Başla

```
// Bir sayıyı girin
```

```
Input: sayı
```

sayı	asalMi	bölen
7	True	2

```
// Sayının asal olup olmadığını kontrol etmek için bir bayrak (flag) başlatın  
asalMi = True
```

```
// 2'den başlayarak sayının yarısına kadar olan tüm bölenleri kontrol et
```

```
Döngü bölen = 2'den başlayarak sayı / 2'e kadar
```

```
    Eğer sayı % bölen = 0 ise
```

```
        // Bölünüyorsa, sayı asal değildir
```

```
        asalMi = False
```

```
        Döngüyü Kır
```

```
    End Eğer
```

```
Döngüyü Bitir
```





# Asal Sayı Kontrolü

Başla

```
// Bir sayıyı girin
```

```
Input: sayı
```

sayı	asalMi	bölen
7	True	2

```
// Sayının asal olup olmadığını kontrol etmek için bir bayrak (flag) başlatın  
asalMi = True
```

```
// 2'den başlayarak sayının yarısına kadar olan tüm bölenleri kontrol et  
Döngü bölen = 2'den başlayarak sayı / 2'e kadar
```

```
Eğer sayı % bölen = 0 ise
```

```
    // Bölünüyorsa, sayı asal değildir
```

```
    asalMi = False
```

```
    Döngüyü Kır
```

```
End Eğer
```

```
Döngüyü Bitir
```



# Asal Sayı Kontrolü

Başla

```
// Bir sayıyı girin
```

```
Input: sayı
```

sayı	asalMi	bölen
7	True	3

```
// Sayının asal olup olmadığını kontrol etmek için bir bayrak (flag) başlatın  
asalMi = True
```

```
// 2'den başlayarak sayının yarısına kadar olan tüm bölenleri kontrol et
```

```
Döngü bölen = 2'den başlayarak sayı / 2'e kadar
```

```
    Eğer sayı % bölen = 0 ise
```

```
        // Bölünüyorsa, sayı asal değildir
```

```
        asalMi = False
```

```
        Döngüyü Kır
```

```
    End Eğer
```

```
Döngüyü Bitir
```



# Asal Sayı Kontrolü

Başla

```
// Bir sayıyı girin
```

```
Input: sayı
```

sayı	asalMi	bölen
7	True	3

```
// Sayının asal olup olmadığını kontrol etmek için bir bayrak (flag) başlatın  
asalMi = True
```

```
// 2'den başlayarak sayının yarısına kadar olan tüm bölenleri kontrol et  
Döngü bölen = 2'den başlayarak sayı / 2'e kadar
```

```
Eğer sayı % bölen = 0 ise
```

```
    // Bölünüyorsa, sayı asal değildir
```

```
    asalMi = False
```

```
    Döngüyü Kır
```

```
End Eğer
```

```
Döngüyü Bitir
```



# Asal Sayı Kontrolü

Başla

// Bir sayıyı girin

Input: sayı

sayı	asalMi	bölen
7	True	3

// Sayının asal olup olmadığını kontrol etmek için bir bayrak (flag) başlatın  
asalMi = True

// 2'den başlayarak sayının yarısına kadar olan tüm bölenleri kontrol et

Döngü bölen = 2'den başlayarak sayı / 2'e kadar

Eğer sayı % bölen = 0 ise

// Bölünüyorsa, sayı asal değildir

asalMi = False

Döngüyü Kır

End Eğer

**Döngüyü Bitir**



# Asal Sayı Kontrolü

Başla

// Bir sayıyı girin

**Input: sayı**

sayı	asalMi	bölen
21		

// Sayının asal olup olmadığını kontrol etmek için bir bayrak (flag) başlatın  
asalMi = True

// 2'den başlayarak sayının yarısına kadar olan tüm bölenleri kontrol et

Döngü bölen = 2'den başlayarak sayı / 2'e kadar

Eğer sayı % bölen = 0 ise

// Bölünüyorsa, sayı asal değildir

asalMi = False

Döngüyü Kır

End Eğer

Döngüyü Bitir



# Asal Sayı Kontrolü

Başla

// Bir sayıyı girin

Input: sayı

sayı	asalMi	bölen
21	True	

// Sayının asal olup olmadığını kontrol etmek için bir bayrak (flag) başlatın

**asalMi = True**

// 2'den başlayarak sayının yarısına kadar olan tüm bölenleri kontrol et

Döngü bölen = 2'den başlayarak sayı / 2'e kadar

Eğer sayı % bölen = 0 ise

// Bölünüyorsa, sayı asal değildir

asalMi = False

Döngüyü Kır

End Eğer

Döngüyü Bitir



# Asal Sayı Kontrolü

Başla

```
// Bir sayıyı girin
```

```
Input: sayı
```

sayı	asalMi	bölen
21	True	2

```
// Sayının asal olup olmadığını kontrol etmek için bir bayrak (flag) başlatın  
asalMi = True
```

```
// 2'den başlayarak sayının yarısına kadar olan tüm bölenleri kontrol et
```

```
Döngü bölen = 2'den başlayarak sayı / 2'e kadar
```

```
    Eğer sayı % bölen = 0 ise
```

```
        // Bölünüyorsa, sayı asal değildir
```

```
        asalMi = False
```

```
        Döngüyü Kır
```

```
    End Eğer
```

```
Döngüyü Bitir
```



# Asal Sayı Kontrolü

Başla

```
// Bir sayıyı girin
```

```
Input: sayı
```

sayı	asalMi	bölen
21	True	2

```
// Sayının asal olup olmadığını kontrol etmek için bir bayrak (flag) başlatın  
asalMi = True
```

```
// 2'den başlayarak sayının yarısına kadar olan tüm bölenleri kontrol et  
Döngü bölen = 2'den başlayarak sayı / 2'e kadar
```

```
Eğer sayı % bölen = 0 ise
```

```
    // Bölünüyorsa, sayı asal değildir
```

```
    asalMi = False
```

```
    Döngüyü Kır
```

```
End Eğer
```

```
Döngüyü Bitir
```





# Asal Sayı Kontrolü

Başla

```
// Bir sayıyı girin
```

```
Input: sayı
```

sayı	asalMi	bölen
21	True	3

```
// Sayının asal olup olmadığını kontrol etmek için bir bayrak (flag) başlatın  
asalMi = True
```

```
// 2'den başlayarak sayının yarısına kadar olan tüm bölenleri kontrol et
```

```
Döngü bölen = 2'den başlayarak sayı / 2'e kadar
```

```
    Eğer sayı % bölen = 0 ise
```

```
        // Bölünüyorsa, sayı asal değildir
```

```
        asalMi = False
```

```
        Döngüyü Kır
```

```
    End Eğer
```

```
Döngüyü Bitir
```



# Asal Sayı Kontrolü

Başla

```
// Bir sayıyı girin
```

```
Input: sayı
```

sayı	asalMi	bölen
21	True	3

```
// Sayının asal olup olmadığını kontrol etmek için bir bayrak (flag) başlatın  
asalMi = True
```

```
// 2'den başlayarak sayının yarısına kadar olan tüm bölenleri kontrol et  
Döngü bölen = 2'den başlayarak sayı / 2'e kadar
```

```
Eğer sayı % bölen = 0 ise
```

```
    // Bölünüyorsa, sayı asal değildir
```

```
    asalMi = False
```

```
    Döngüyü Kır
```

```
End Eğer
```

```
Döngüyü Bitir
```



# Asal Sayı Kontrolü

Başla

```
// Bir sayıyı girin
```

```
Input: sayı
```

sayı	asalMi	bölen
21	False	3

```
// Sayının asal olup olmadığını kontrol etmek için bir bayrak (flag) başlatın  
asalMi = True
```

```
// 2'den başlayarak sayının yarısına kadar olan tüm bölenleri kontrol et
```

```
Döngü bölen = 2'den başlayarak sayı / 2'e kadar
```

```
    Eğer sayı % bölen = 0 ise
```

```
        // Bölünüyorsa, sayı asal değildir
```

```
        asalMi = False
```

```
        Döngüyü Kır
```

```
    End Eğer
```

```
Döngüyü Bitir
```



# Asal Sayı Kontrolü

Başla

```
// Bir sayıyı girin
```

```
Input: sayı
```

sayı	asalMi	bölen
7	False	3

```
// Sayının asal olup olmadığını kontrol etmek için bir bayrak (flag) başlatın  
asalMi = True
```

```
// 2'den başlayarak sayının yarısına kadar olan tüm bölenleri kontrol et
```

```
Döngü bölen = 2'den başlayarak sayı / 2'e kadar
```

```
    Eğer sayı % bölen = 0 ise
```

```
        // Bölünüyorsa, sayı asal değildir
```

```
        asalMi = False
```

```
        Döngüyü Kır
```

```
    End Eğer
```

```
Döngüyü Bitir
```



# Örnek Algoritmalar

- İki Tamsayıyı Küçükten Büyüğe Doğru Yazdırma
- Üç Tamsayıyı Küçükten Büyüğe Doğru Yazdırma
- Faktöriyel Bulma
- Fibonacci Serisi Bulma
- Palindrom Kontrolü
- Asal Sayı Kontrolü
- **Üs Alma**

# Üs Alma



Başla

```
// Bir sayıyı girin
```

```
Input: taban
```

```
// Bir üssü girin
```

```
Input: üs
```

```
// Sonucu saklamak için bir değişkeni başlatın
```

```
sonuc = 1
```

```
// Üssü kullanarak sonucu hesaplayın
```

```
Döngü i = 1'den başlayarak üs kadar
```

```
    sonuc = sonuc * taban
```

```
Yaz "Sonuç: ", sonuc
```

Bitir

# Üs Alma



Başla

```
// Bir sayıyı girin
```

**Input: taban**

```
// Bir üssü girin
```

Input: üs

```
// Sonucu saklamak için bir değişkeni başlatın
```

```
sonuc = 1
```

```
// Üssü kullanarak sonucu hesaplayın
```

```
Döngü i = 1'den başlayarak üs kadar
```

```
sonuc = sonuc * taban
```

```
Yaz "Sonuç: ", sonuc
```

Bitir

taban	üs	sonuc	i
3			

# Üs Alma



Başla

```
// Bir sayıyı girin
```

```
Input: taban
```

```
// Bir üssü girin
```

```
Input: üs
```

```
// Sonucu saklamak için bir değişkeni başlatın
```

```
sonuc = 1
```

```
// Üssü kullanarak sonucu hesaplayın
```

```
Döngü i = 1'den başlayarak üs kadar
```

```
sonuc = sonuc * taban
```

```
Yaz "Sonuç: ", sonuc
```

Bitir

taban	üs	sonuc	i
3	4		



# Üs Alma



Başla

```
// Bir sayıyı girin
```

```
Input: taban
```

```
// Bir üssü girin
```

```
Input: üs
```

taban	üs	sonuc	i
3	4	1	

```
// Sonucu saklamak için bir değişkeni başlatın
```

```
sonuc = 1
```

```
// Üssü kullanarak sonucu hesaplayın
```

```
Döngü i = 1'den başlayarak üs kadar
```

```
sonuc = sonuc * taban
```

```
Yaz "Sonuç: ", sonuc
```

Bitir

# Üs Alma



Başla

```
// Bir sayıyı girin
```

```
Input: taban
```

```
// Bir üssü girin
```

```
Input: üs
```

```
// Sonucu saklamak için bir değişkeni başlatın
```

```
sonuc = 1
```

```
// Üssü kullanarak sonucu hesaplayın
```

```
Döngü i = 1'den başlayarak üs kadar
```

```
sonuc = sonuc * taban
```

```
Yaz "Sonuç: ", sonuc
```

Bitir

taban	üs	sonuc	i
3	4	1	1

# Üs Alma



Başla

```
// Bir sayıyı girin
```

```
Input: taban
```

```
// Bir üssü girin
```

```
Input: üs
```

```
// Sonucu saklamak için bir değişkeni başlatın
```

```
sonuc = 1
```

```
// Üssü kullanarak sonucu hesaplayın
```

```
Döngü i = 1'den başlayarak üs kadar
```

```
sonuc = sonuc * taban
```

```
Yaz "Sonuç: ", sonuc
```

Bitir

taban	üs	sonuc	i
3	4	3	1

# Üs Alma



Başla

```
// Bir sayıyı girin
```

```
Input: taban
```

```
// Bir üssü girin
```

```
Input: üs
```

```
// Sonucu saklamak için bir değişkeni başlatın
```

```
sonuc = 1
```

```
// Üssü kullanarak sonucu hesaplayın
```

```
Döngü i = 1'den başlayarak üs kadar
```

```
sonuc = sonuc * taban
```

```
Yaz "Sonuç: ", sonuc
```

Bitir

taban	üs	sonuc	i
3	4	3	2

# Üs Alma



Başla

```
// Bir sayıyı girin
```

```
Input: taban
```

```
// Bir üssü girin
```

```
Input: üs
```

```
// Sonucu saklamak için bir değişkeni başlatın
```

```
sonuc = 1
```

```
// Üssü kullanarak sonucu hesaplayın
```

```
Döngü i = 1'den başlayarak üs kadar
```

```
sonuc = sonuc * taban
```

```
Yaz "Sonuç: ", sonuc
```

Bitir

taban	üs	sonuc	i
3	4	9	2

# Üs Alma



Başla

```
// Bir sayıyı girin
```

```
Input: taban
```

```
// Bir üssü girin
```

```
Input: üs
```

```
// Sonucu saklamak için bir değişkeni başlatın
```

```
sonuc = 1
```

```
// Üssü kullanarak sonucu hesaplayın
```

```
Döngü i = 1'den başlayarak üs kadar
```

```
sonuc = sonuc * taban
```

```
Yaz "Sonuç: ", sonuc
```

Bitir

taban	üs	sonuc	i
3	4	9	3

# Üs Alma



Başla

```
// Bir sayıyı girin
```

```
Input: taban
```

```
// Bir üssü girin
```

```
Input: üs
```

```
// Sonucu saklamak için bir değişkeni başlatın
```

```
sonuc = 1
```

```
// Üssü kullanarak sonucu hesaplayın
```

```
Döngü i = 1'den başlayarak üs kadar
```

```
sonuc = sonuc * taban
```

```
Yaz "Sonuç: ", sonuc
```

Bitir

taban	üs	sonuc	i
3	4	27	3

# Üs Alma



Başla

```
// Bir sayıyı girin
```

```
Input: taban
```

```
// Bir üssü girin
```

```
Input: üs
```

```
// Sonucu saklamak için bir değişkeni başlatın
```

```
sonuc = 1
```

```
// Üssü kullanarak sonucu hesaplayın
```

```
Döngü i = 1'den başlayarak üs kadar
```

```
sonuc = sonuc * taban
```

```
Yaz "Sonuç: ", sonuc
```

Bitir

taban	üs	sonuc	i
3	4	27	4



# Üs Alma



Başla

```
// Bir sayıyı girin
```

```
Input: taban
```

```
// Bir üssü girin
```

```
Input: üs
```

```
// Sonucu saklamak için bir değişkeni başlatın
```

```
sonuc = 1
```

```
// Üssü kullanarak sonucu hesaplayın
```

```
Döngü i = 1'den başlayarak üs kadar
```

```
sonuc = sonuc * taban
```

```
Yaz "Sonuç: ", sonuc
```

Bitir

taban	üs	sonuc	i
3	4	81	4

# Üs Alma



Başla

```
// Bir sayıyı girin
```

```
Input: taban
```

```
// Bir üssü girin
```

```
Input: üs
```

```
// Sonucu saklamak için bir değişkeni başlatın
```

```
sonuc = 1
```

```
// Üssü kullanarak sonucu hesaplayın
```

```
Döngü i = 1'den başlayarak üs kadar
```

```
sonuc = sonuc * taban
```

```
Yaz "Sonuç: ", sonuc
```

Bitir

taban	üs	sonuc	i
3	4	81	4



# Arama Algoritmaları

- Bir koleksiyon içerisinde belirli bir öğenin varlığını sorgular.
- Veri işlemek ve bilgi edinmek için kullanılan temel bir işlemdir.
- Belirli öğelerin hızlı ve doğru bir şekilde bulunmasına yardımcı olur.
- **Kullanım Alanları:**
  - Kitapların kütüphanedeki yerini bulma,
  - İnternet üzerinde bir web sitesinde belirli bir sayfayı bulma,
  - Veri tabanında belirli bir kaydı arama.



# Arama Algoritmaları

- **Problem Tanımı**

- **Girdi:**

- Nesnelerin bir koleksiyonu, «Sepet» diyelim.
- Belirli bir nesne, «Elma» diyelim.

- **Çıktı:**

- Eğer «Elma», "Sepet" içindeyse "True".
- Eğer «Elma», "Sepet" içinde değilse "False".



# Arama Algoritmaları

- Rastgele (Random)
- Doğrusal (Linear)
- İkili (Binary)



# Rastgele Arama

- Koleksiyon içerisinde öğeleri rastgele kontrol eder.
- Her adımda rastgele bir öğe seçer.
- Hedefi bulana veya arama alanı tükenene kadar çalışır.
- Büyük veri koleksiyonlarında hızlı arama yapmak için kullanılabilir.
- **Avantajlar:**
  - Basit ve hızlıdır, veri koleksiyonunun sırası önemli değildir.
- **Dezavantajlar:**
  - En kötü durumda veri koleksiyonunun tamamını taramak gerekir.
  - Verimli olmayabilir.



# Rastgele Arama

- Rastgele bir öge seç.
- Seçilen ögeyi hedefle karşılaştır.
- Eğer hedef bulunduysa işlemi sonlandır.
- Eğer hedef bulunamazsa adımları tekrarla.



# Rastgele Arama

- Her zaman etkili bir çözüm sağlamaz.
- Öğeyi bulma olasılığını artırabilir, kesin bir sonuç garantisi vermez.





# Rastgele Arama

- Elimizde rastgele bir dizi olsun: [77, 42, 64, 12, **23**, 55, 7, 48, 31]
- Aramak istediğimiz öge: 48
- Adım 1:
  - Dizi üzerinde rastgele bir eleman seçin ve aranan öge (48) ile karşılaştırın.
  - Örneğin, ilk seçilen eleman 23'tür, bu nedenle 23 ile 48'i karşılaştırın.



# Rastgele Arama

- Elimizde rastgele bir dizi olsun: [77, **42**, 64, 12, 23, 55, 7, 48, 31]
- Aramak istediğimiz öge: 48
- Adım 2:
  - Aranan öge (48) henüz bulunamadı, çünkü seçilen eleman 23'tür.
  - Dizi üzerinde bir başka rastgele eleman seçin ve tekrar karşılaştırın.
  - Bu sefer seçilen eleman 42'dir.



# Rastgele Arama

- Elimizde rastgele bir dizi olsun: [77, 42, 64, 12, 23, 55, **7**, 48, 31]
- Aramak istediğimiz öge: 48
- Adım 3:
  - 42 ile 48'i karşılaştırın.
  - Aranan öge (48) henüz bulunamadı, çünkü seçilen eleman 42'dir.
  - Yeni bir rastgele eleman seçin.



# Rastgele Arama

- Elimizde rastgele bir dizi olsun: [77, 42, 64, 12, 23, 55, 7, **48**, 31]
- Aramak istediğimiz öge: 48
- Adım 4:
- Bu adımları tekrarlayarak aranan öğeyi (48) bulana kadar devam edin.
- Örneğin, bir sonraki seçilen eleman 48 olduğunda aranan öge bulundu.



# Arama Algoritmaları

- Rastgele (Random)
- **Doğrusal (Linear)**
- İkili (Binary)



# Doğrusal Arama

- Koleksiyon içinde bir öğeyi bulmanın en temel yoludur.
- Öğeleri sırayla kontrol ederek aranan öğeyi bulmaya çalışır.
- Küçük koleksiyonlarda etkili bir çözümdür.
- Öğelerin sıralı olması gerekmez, herhangi bir sırada bulunabilirler.
- **Avantajlar:**
  - Basit ve anlaşılır, herhangi bir sırayla çalışabilir.
- **Dezavantajlar:**
  - Büyük koleksiyonlarda yavaş, her öğeyi kontrol etmesi gerekebilir.



# Doğrusal Arama

- Öğeleri bir listeye koyun
- İlk öğeden başlayarak sırayla hedef öğe ile karşılaştırın.
- Eğer hedef öğe bulunursa işlemi sonlandırın ve bulunan öğeyi dönün.
- Eğer hedef öğe bulunmazsa bir sonraki öğeye geçin.
- Listenin sonuna kadar devam edin.



# Doğrusal Arama

- Elimizde sıralanmamış bir dizi olsun: [42, 17, 8, 23, 31]
- Adım 1:
  - Aranacak öğeyi belirleyin, örneğin 23'ü aranıyor.
  - Dizinin ilk elemanından başlayın (42) ve aranan öğe ile karşılaştırın.





# Doğrusal Arama

- Elimizde sıralanmamış bir dizi olsun: [42, **17**, 8, 23, 31]
- Adım 2:
- İlk eleman (42) aranan öge (23) ile eşleşmez.
- Bir sonraki elemana geçin ve 17 ile karşılaştırın.



# Doğrusal Arama

- Elimizde sıralanmamış bir dizi olsun: [42, 17, **8**, 23, 31]
- Adım 3:
- İkinci eleman (17) de aranan öge ile eşleşmez.
- Bir sonraki elemana geçin ve 8 ile karşılaştırın.



# Doğrusal Arama

- Elimizde sıralanmamış bir dizi olsun: [42, 17, 8, **23**, 31]
- Adım 4:
- Üçüncü eleman (8) de aranan öge ile eşleşmez.
- Bir sonraki elemana geçin ve 23 ile karşılaştırın.



# Doğrusal Arama

- Elimizde sıralanmamış bir dizi olsun: [42, 17, 8, **23**, 31]
- Adım 5:
- Dördüncü eleman (23), aranan öge ile eşleşir.
- Aranan öğeyi bulduk ve bu adımda arama işlemi sona erer.



# Arama Algoritmaları

- Rastgele (Random)
- Doğrusal (Linear)
- **İkili (Binary)**



# İkili Arama

- Bir koleksiyon içerisindeki bir öğeyi bulmanın hızlı ve etkili bir yoludur.
- Sıralı veriler (küçükten büyüğe veya büyükten küçüğe) üzerinde çalışır.
- Büyük veri koleksiyonlarında hızlı bir şekilde arama yapmak için idealdir.
- Her adımda listenin yarısı kadar öğeyi elemeyi sağlar.
- **Avantajlar:**
  - Hızlı, büyük veri koleksiyonlarında etkilidir.
- **Dezavantajlar:**
  - Veriler sıralı olmalıdır.



# İkili Arama

- Ortadaki öğeyi seçin ve hedefle karşılaştırın.
- Eğer hedef öge bulunursa işlemi sonlandırın.
- Eğer hedef öge, ortadaki öğeden küçükse, sol yarıya odaklan.
- Eğer hedef öge, ortadaki öğeden büyükse, sağ yarıya odaklan.
- Hedef öge bulunana veya arama alanı tükenene kadar tekrarla.



# İkili Arama

- Elimizde sıralanmış bir dizi olsun: [7, 12, 23, 31, **42**, 48, 55, 64, 77]
- Aramak istediğimiz öge: 48
- Adım 1:
- Dizinin ortasındaki elemanı (42) alın ve aranan öge (48) ile karşılaştırın.





# İkili Arama

- Elimizde sıralanmış bir dizi olsun: [7, 12, 23, 31, 42, 48, 55, 64, 77]
- Aramak istediğimiz öge: 48
- Adım 2:
  - 48, ortadaki elemandan (42) büyüktür.
  - Bu nedenle, aranan öğenin dizinin sağ yarısında olduğunu anlarız.
  - Sol yarı artık arama için kullanılmayacak.



# İkili Arama

- Elimizde sıralanmış bir dizi olsun: [7, 12, 23, 31, 42, 48, 55, 64, 77]
- Aramak istediğimiz öge: 48
- Adım 3:
- Sağ yarıyı ele alın: [48, 55, 64, 77]



# İkili Arama

- Elimizde sıralanmış bir dizi olsun: [7, 12, 23, 31, 42, 48, 55, **64**, 77]
- Aramak istediğimiz öge: 48
- Adım 4:
- Sağ yarının ortasındaki elemanı (64) alın ve aranan öge (48) ile karşılaştırın.



# İkili Arama

- Elimizde sıralanmış bir dizi olsun: [7, 12, 23, 31, 42, 48, 55, 64, 77]
- Aramak istediğimiz öge: 48
- Adım 5:
  - 48, ortadaki elemandan (64) küçüktür.
  - Bu nedenle, aranan ögenin dizinin sol yarısında olduğunu anlarız.
  - Sağ yarı artık arama için kullanılmayacak.



# İkili Arama

- Elimizde sıralanmış bir dizi olsun: [7, 12, 23, 31, 42, 48, 55, 64, 77]
- Aramak istediğimiz öge: 48
- Adım 6:
- Sol yarıyı ele alın: [48, 55]



# İkili Arama

- Elimizde sıralanmış bir dizi olsun: [7, 12, 23, 31, 42, **48**, 55, 64, 77]
- Aramak istediğimiz öge: 48
- Adım 7:
- Sol yarının ortasındaki elemanı (48) alın ve aranan öge (48) ile karşılaştırın.



# İkili Arama

- Elimizde sıralanmış bir dizi olsun: [7, 12, 23, 31, 42, 48, 55, 64, 77]
- Aramak istediğimiz öge: 48
- Adım 8:
- Aranan öge (48) bulundu.
- Arama işlemi başarıyla tamamlandı.



# İkili Arama ve Doğrusal Arama: Karşılaştırma

- Lineer arama
  - Koleksiyon içinde bir öğeyi sırayla kontrol ederek arar.
  - Her adımda bir öğeyi kontrol eder
  - Hedefi bulana veya koleksiyon sonuna kadar devam eder.
  - Büyük koleksiyonlarda yavaş olabilir.
- İkili arama
  - Sıralı verilerde hedef öğeyi hızlıca bulmak için kullanılır.
  - Verilerin sıralı (küçükten büyüğe veya büyükten küçüğe) olması gerekir.
  - Büyük koleksiyonlarda hızlıdır,
    - her adımda listenin yarısını eleme avantajı vardır.





# Sıralama Algoritmaları

- Veri koleksiyonundaki öğeleri belirli bir kurala göre düzenler.
- Genellikle öğeleri küçükten büyüğe veya büyükten küçüğe sıralar.
- Veri analizi ve arama işlemleri için temel bir adımdır.
- Sıralı veriler, arama işlemlerini hızlandırır.



# Sıralama Algoritmaları

- Rastgele (Random)
- Seçmeli (Selection Sort)
- Kabarcık (Bubble Sort)



# Rastgele Sıralama

- Bir veri koleksiyonunu rastgele bir düzende sıralama işlemidir.
- Veri koleksiyonunu rastgele karışık hale getirerek verileri çeşitlendirmek ve analiz etmek için kullanılır.
- Bazı algoritmaların başlangıç verilerini rastgele sıralama işlemi ile iyileştirmesinde kullanılır.
- Veri koleksiyonundaki öğeleri rastgele bir şekilde karıştırır.
- **Avantajlar:** Verileri rastgele sıralayarak daha farklı sonuçlar elde edilebilir.
- **Dezavantajlar:** Verilerin doğru bir şekilde sıralanmasını sağlamaz.



# Rastgele Sıralama

- Elimizde sıralanmamış bir dizi olsun: [42, 17, 8, 23, 31]
- Adım 1:
- Dizi elemanlarını rastgele bir sırayla düzenleyin.
- Her düzenleme farklı bir sonuç üretecektir. Örneğin, dizi aşağıdaki gibi bir düzende olabilir: [31, 17, 42, 8, 23]



# Rastgele Sıralama

- Elimizde sıralanmamış bir dizi olsun: [31, 17, 42, 8, 23]
- Adım 2:
- Dizi elemanlarını tekrar rastgele bir sırayla düzenleyin.
- Bu sefer farklı bir düzenleme olabilir: [8, 23, 31, 42, 17]



# Rastgele Sıralama

- Elimizde sıralanmamış bir dizi olsun: [8, 23, 31, 42, 17]
- Adım 3:
- Dizi elemanlarını tekrar rastgele bir sırayla düzenleyin.
- Bu sefer farklı bir düzenleme olabilir: [8, 23, 17, 31, 42]



# Sıralama Algoritmaları

- Rastgele Sıralama (Random)
- **Seçmeli Sıralama (Selection Sort)**
- Kabarcık Sıralama (Bubble Sort)



# Seçmeli Sıralama

- Öğeleri karşılaştırarak sıralar.
- Basit bir algoritmadır.
- En küçük veya en büyük öğeyi bulup sırayla yer değiştirerek çalışır.
- **Avantajlar:** Basit ve anlaşılır, küçük veri koleksiyonlarında etkili.
- **Dezavantajlar:** Büyük veri koleksiyonlarında yavaştır.





# Seçmeli Sıralama

- Veri koleksiyonundaki en küçük (veya en büyük) öğeyi bulun.
- Bu öğeyi sıralı koleksiyonun başına taşıyın.
- Koleksiyonun sonraki bölümündeki en küçük (veya en büyük) öğeyi bulun.
- Bu öğeyi sıralı koleksiyonun ikinci pozisyonuna taşıyın.
- Bu adımları koleksiyonun sonuna kadar tekrarlayın.



# Seçmeli Sıralama

- Elimizde sıralanmamış bir dizi olsun: [64, 25, 12, 22, **11**, 36, 48]
- Adım 1:
  - Sıralanmamış kısmın içindeki en küçük elemanı bulun (bu durumda 11).
  - İlk elemanla (64) yer değiştirin.
  - Dizi birinci adımdan sonra: [**11**, 25, 12, 22, 64, 36, 48]



# Seçmeli Sıralama

- Elimizde sıralanmamış bir dizi olsun: [11, 25, 12, 22, 64, 36, 48]
- Adım 2:
- İkinci en küçük elemanı bulun (bu durumda 12) ve ikinci elemanla yer değiştirin.
- Dizi ikinci adımdan sonra: [11, 12, 25, 22, 64, 36, 48]



# Seçmeli Sıralama

- Elimizde sıralanmamış bir dizi olsun: [11, 12, 25, 22, 64, 36, 48]
- Adım 3:
- Üçüncü en küçük elemanı bulun (bu durumda 22) ve üçüncü elemanla yer değiştirin.
- Dizi üçüncü adımdan sonra: [11, 12, 22, 25, 64, 36, 48]



# Seçmeli Sıralama

- Elimizde sıralanmamış bir dizi olsun: [11, 12, 22, 25, 64, 36, 48]
- Adım 4:
- Dördüncü en küçük elemanı bulun (bu durumda 25) ve dördüncü elemanla yer değiştirin.
- Dizi dördüncü adımdan sonra: [11, 12, 22, 25, 64, 36, 48]



# Seçmeli Sıralama

- Elimizde sıralanmamış bir dizi olsun: [11, 12, 22, 25, 64, 36, 48]
- Adım 5:
- Beşinci en küçük elemanı bulun (bu durumda 36) ve beşinci elemanla yer değiştirin.
- Dizi beşinci adımdan sonra: [11, 12, 22, 25, 36, 64, 48]



# Seçmeli Sıralama

- Elimizde sıralanmamış bir dizi olsun: [11, 12, 22, 25, 36, 64, 48]
- Adım 6:
- Altıncı en küçük elemanı bulun (bu durumda 48) ve altıncı elemanla yer değiştirin.
- Dizi altıncı adımdan sonra: [11, 12, 22, 25, 36, 48, 64]



# Seçmeli Sıralama

- Elimizde sıralanmamış bir dizi olsun: [11, 12, 22, 25, 36, 48, 64]
- Adım 7:
- Dizinin sonuna ulaşıldığı için değişiklik yok.
- Dizi yedinci adımdan sonra: [11, 12, 22, 25, 36, 48, 64]





# Sıralama Algoritmaları

- Rastgele (Random)
- Seçmeli (Selection Sort)
- **Kabarcık (Bubble Sort)**



# Kabarcık Sıralama

- Öğeleri sırayla karşılaştırarak sıralar.
- Basit bir algoritmadır.
- Yan yana duran öğeleri sırayla karşılaştırarak en büyük öğeyi bulur.
  - Koleksiyonun sonuna doğru "kabarcık" gibi taşır.
- **Avantajlar:** Basit ve anlaşılır, küçük veri koleksiyonlarında etkili.
- **Dezavantajlar:** Büyük veri koleksiyonlarında çok yavaştır.



# Kabarcık Sıralama

- Koleksiyonun başından itibaren başlayın.
- İki yan yana öğeyi karşılaştırın ve gerektiğinde yerlerini değiştirin,
  - büyük olan öğeyi sağa taşıyın.
- Koleksiyonun sonuna kadar bu adımları tekrarlayın.
- En büyük öge, koleksiyonun sonuna taşındığı için son öğeyi sıralanmış kabul edin.
- İlk dört adımı koleksiyonun sonuna kadar tekrarlayın.



# Kabarcık Sıralama

- Elimizde sıralanmamış bir dizi olsun: [64, 25, 12, 22, 11, 36, 48]
- Adım 1:
  - Dizinin ilk iki elemanını karşılaştırın (64 ve 25).
  - 25, 64'ten küçük olduğu için bu iki elemanın yerini değiştirin.
  - Dizi birinci adımdan sonra: [25, 64, 12, 22, 11, 36, 48]



# Kabarcık Sıralama

- Elimizde sıralanmamış bir dizi olsun: [25, 64, 12, 22, 11, 36, 48]
- Adım 2:
- Şimdi ikinci ve üçüncü elemanları karşılaştırın (64 ve 12).
- 12, 64'ten küçük olduğu için bu iki elemanın yerini değiştirin.
- Dizi ikinci adımdan sonra: [25, 12, 64, 22, 11, 36, 48]



# Kabarcık Sıralama

- Elimizde sıralanmamış bir dizi olsun: [25, 12, **64**, **22**, 11, 36, 48]
- Adım 3:
- Üçüncü ve dördüncü elemanları karşılaştırın (64 ve 22).
- 22, 64'ten küçük olduğu için bu iki elemanın yerini değiştirin.
- Dizi üçüncü adımdan sonra: [25, 12, **22**, **64**, 11, 36, 48]



# Kabarcık Sıralama

- Elimizde sıralanmamış bir dizi olsun: [25, 12, 22, **64**, **11**, 36, 48]
- Adım 4:
  - Dördüncü ve beşinci elemanları karşılaştırın (64 ve 11).
  - 11, 64'ten küçük olduğu için bu iki elemanın yerini değiştirin.
  - Dizi dördüncü adımdan sonra: [25, 12, 22, **11**, **64**, 36, 48]



# Kabarcık Sıralama

- Elimizde sıralanmamış bir dizi olsun: [25, 12, 22, 11, **64**, **36**, 48]
- Adım 5:
  - Beşinci ve altıncı elemanları karşılaştırın (64 ve 36).
  - 36, 64'ten küçük olduğu için bu iki elemanın yerini değiştirin.
  - Dizi beşinci adımdan sonra: [25, 12, 22, 11, **36**, **64**, 48]





# Kabarcık Sıralama

- Elimizde sıralanmamış bir dizi olsun: [25, 12, 22, 11, 36, **64**, **48**]
- Adım 6:
  - Altıncı ve yedinci elemanları karşılaştırın (64 ve 48).
  - 48, 64'ten küçük olduğu için bu iki elemanın yerini değiştirin.
  - Dizi altıncı adımdan sonra: [25, 12, 22, 11, 36, **48**, **64**]



# Kabarcık Sıralama

- Elimizde sıralanmamış bir dizi olsun: [25, 12, 22, 11, 36, 48, 64]
- Adım 7:
- Dizinin ilk iki elemanını karşılaştırın (25 ve 12).
- 12, 25'ten küçük olduğu için bu iki elemanın yerini değiştirin.
- Dizi 7. adımdan sonra: [12, 25, 22, 11, 36, 48, 64]



# Kabarcık Sıralama

- Elimizde sıralanmamış bir dizi olsun: [12, **25**, **22**, 11, 36, 48, 64]
- Adım 8:
  - Dizinin ikinci ve üçüncü elemanını karşılaştırın (25 ve 22).
  - 22, 25'ten küçük olduğu için bu iki elemanın yerini değiştirin.
  - Dizi 8. adımdan sonra: [12, **22**, **25**, 11, 36, 48, 64]



# Kabarcık Sıralama

- Elimizde sıralanmamış bir dizi olsun: [12, 22, **25**, **11**, 36, 48, 64]
- Adım 9:
  - Dizinin üçüncü ve dördüncü elemanını karşılaştırın (25 ve 11).
  - 11, 25'ten küçük olduğu için bu iki elemanın yerini değiştirin.
  - Dizi 9. adımdan sonra: [12, 22, **11**, **25**, 36, 48, 64]



# Kabarcık Sıralama

- Elimizde sıralanmamış bir dizi olsun: [12, 22, 11, **25, 36**, 48, 64]
- Adım 10:
  - Dizinin dördüncü ve beşinci elemanını karşılaştırın (25 ve 36).
  - 36, 25'ten büyük olduğu için değişiklik yok.
  - Dizi 10. adımdan sonra: [12, 22, 11, **25, 36**, 48, 64]



# Kabarcık Sıralama

- Elimizde sıralanmamış bir dizi olsun: [12, 22, 11, 25, **36, 48**, 64]
- Adım 11:
  - Dizinin beşinci ve altıncı elemanını karşılaştırsın (36 ve 48).
  - 48, 36'dan büyük olduğu için değişiklik yok.
  - Dizi 11. adımdan sonra: [12, 22, 11, 25, **36, 48**, 64]



# Kabarcık Sıralama

- Elimizde sıralanmamış bir dizi olsun: [12, **22**, **11**, 25, 36, 48, 64]
- Adım 13:
  - Dizinin ikinci ve üçüncü elemanını karşılaştırın (22 ve 11).
  - 11, 22'den küçük olduğu için iki elemanın yerini değiştirin.
  - Dizi 13. adımdan sonra: [12, **11**, **22**, 25, 36, 48, 64]



# Kabarcık Sıralama

- Elimizde sıralanmamış bir dizi olsun: [12, 11, 22, 25, 36, 48, 64]
- Adım 16:
  - Dizinin birinci ve ikinci elemanını karşılaştırın (12 ve 11).
  - 11, 12'den küçük olduğu için iki elemanın yerini değiştirin.
  - Dizi 16. adımdan sonra: [11, 12, 22, 25, 36, 48, 64]





# Kabarcık Sıralama

- Elimizde sıralanmamış bir dizi olsun: [11, 12, 22, 25, 36, 48, 64]
- Adım 21:
- Dizinin birinci ve ikinci elemanını karşılaştırın (11 ve 12).
- 12, 11'den büyük olduğu için değişiklik yok.
- Dizi 21. adımdan sonra: [11, 12, 22, 25, 36, 48, 64]



SON