



# Bölüm 2. Bağlı Liste

Olcay Taner Yıldız

2014



## Giriş

### Bağlı Liste Tanımı

### Temel Bağlı Liste İşlemleri

### İki Listeyi Birleştirme

### Çift Bağlı Liste Tanımı

### Temel Çift Bağlı Liste İşlemleri

### Dairesel Bağlı Liste

### Uygulama: Polinom Aritmetiği

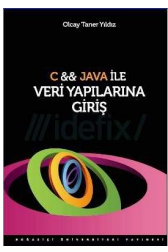
# Giriş



# Bir dizinin $k$ 'nıncı elemanını silen algoritma

<u>Giriş</u>	1
<u>Bağlı Liste Tanımı</u>	2
<u>Temel Bağlı Liste İşlemleri</u>	3
<u>İki Listeyi Birleştirme</u>	5
<u>Çift Bağlı Liste Tanımı</u>	6
<u>Temel Çift Bağlı Liste İşlemleri</u>	
<u>Dairesel Bağlı Liste</u>	
<u>Uygulama: Polinom Aritmetiği</u>	

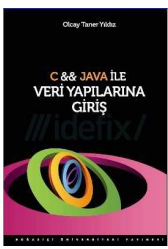
```
void dizidenSil(int[] dizi, int k){
    int i;
    for (i = k; i < dizi.length - 1; i++){
        dizi[i] = dizi[i + 1];
    }
}
```



# Bir dizinin $k$ 'nıncı yerine yeni eleman ekleyen algoritma

<u>Giriş</u>	1
<u>Bağlı Liste Tanımı</u>	2
<u>Temel Bağlı Liste İşlemleri</u>	3
<u>İki Listeyi Birleştirme</u>	4
<u>Çift Bağlı Liste Tanımı</u>	5
<u>Temel Çift Bağlı Liste İşlemleri</u>	6
<u>Dairesel Bağlı Liste</u>	7
<u>Uygulama: Polinom Aritmetiği</u>	

```
void diziyeEkle(int[] dizi, int k, int yeni){
    int i;
    for (i = dizi.length - 2; i >= k; i --){
        dizi [ i + 1 ] = dizi [ i ];
    }
    dizi [ k ] = yeni;
}
```



# Sıralı bir dizide ikili arama yöntemiyle $k$ sayısını arayan algoritma

Giriş	1
Bağlı Liste Tanımı	2
Temel Bağlı Liste İşlemleri	3
İki Listeyi Birleştirme	4
Çift Bağlı Liste Tanımı	5
Temel Çift Bağlı Liste İşlemleri	6
Dairesel Bağlı Liste	7
Uygulama: Polinom Aritmetiği	8
	9
	10
	11
	12
	13
	14
	15

```
void dizideAra(int[] dizi, int k){
    int sol = 0, sag = dizi.length - 1, orta;
    orta = (sol + sag) / 2;
    while (sol <= sag){
        if (k < dizi[orta])
            sag = orta - 1;
        else
            if (k > dizi[orta])
                sol = orta + 1;
            else
                return orta;
        orta = (sol + sag) / 2;
    }
    return -1;
}
```



[Giriş](#)

[Bağlı Liste Tanımı](#)

[Temel Bağlı Liste İşlemleri](#)

[İki Listeyi Birleştirme](#)

[Çift Bağlı Liste Tanımı](#)

[Temel Çift Bağlı Liste İşlemleri](#)

[Dairesel Bağlı Liste](#)

[Uygulama: Polinom Aritmetiği](#)

# Bağlı Liste Tanımı



# Tam sayı içeren eleman veri yapısı

Giriş

Bağlı Liste Tanımı

Temel Bağlı Liste İşlemleri

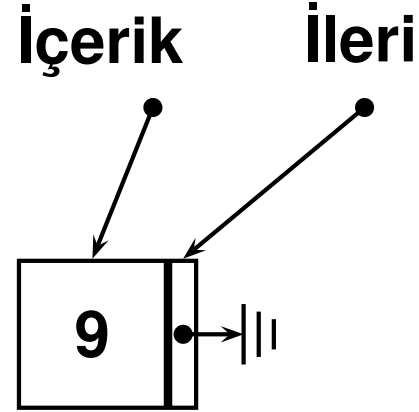
İki Listeyi Birleştirme

Çift Bağlı Liste Tanımı

Temel Çift Bağlı Liste İşlemleri

Dairesel Bağlı Liste

Uygulama: Polinom Aritmetiği





# Tam sayı içeren eleman veri yapısı tanımı

<u>Giriş</u>	1
<u>Bağlı Liste Tanımı</u>	2
<u>Temel Bağlı Liste İşlemleri</u>	3
<u>İki Listeyi Birleştirme</u>	4
<u>Çift Bağlı Liste Tanımı</u>	5
<u>Temel Çift Bağlı Liste İşlemleri</u>	6
<u>Dairesel Bağlı Liste</u>	7
<u>Uygulama: Polinom Aritmetiği</u>	8

```
public class Eleman{
    int icerik;
    Eleman ileri;
    public Eleman(int icerik){
        this.icerik = icerik ;
        ileri = null;
    }
}
```





# Bir bağılı liste ve dizi yapısı

Giriş

Bağılı Liste Tanımı

Temel Bağılı Liste İşlemleri

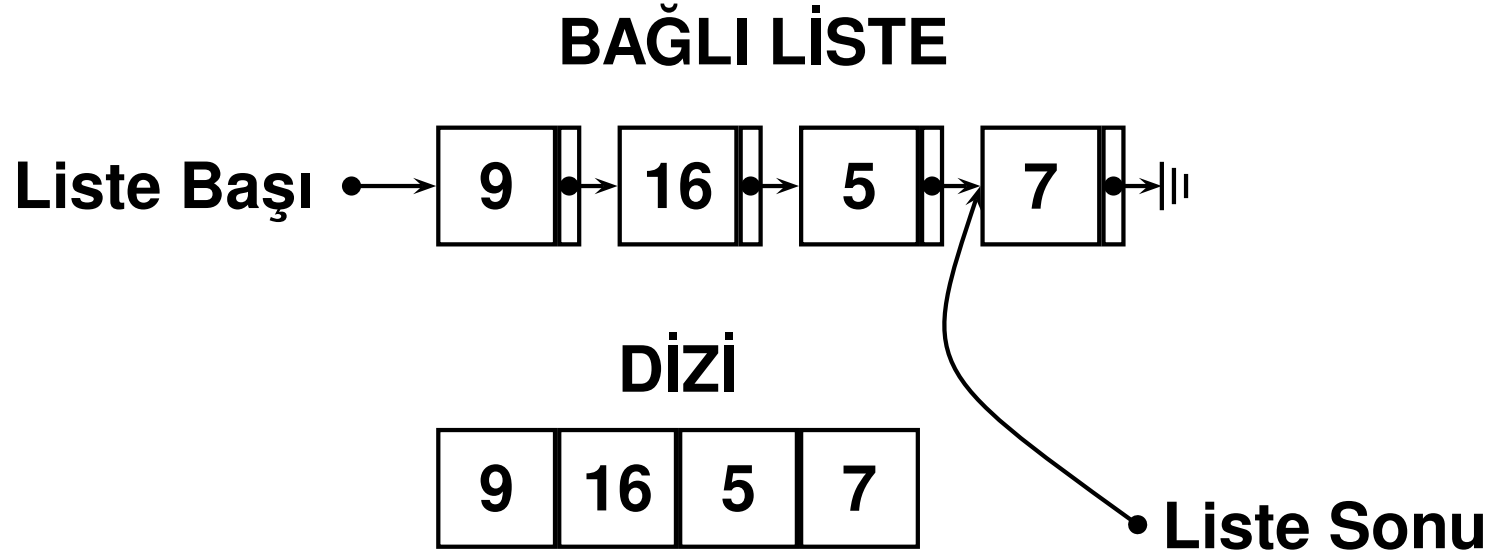
İki Listeyi Birleştirme

Çift Bağılı Liste Tanımı

Temel Çift Bağılı Liste İşlemleri

Dairesel Bağılı Liste

Uygulama: Polinom Aritmetiği





# Tam sayılar içeren bağlı liste tanımı

<u>Giriş</u>	1
<u>Bağlı Liste Tanımı</u>	2
<u>Temel Bağlı Liste İşlemleri</u>	3
<u>İki Listeyi Birleştirme</u>	4
<u>Çift Bağlı Liste Tanımı</u>	5
<u>Temel Çift Bağlı Liste İşlemleri</u>	6
<u>Dairesel Bağlı Liste</u>	7
<u>Uygulama: Polinom Aritmetiği</u>	8

```
public class Liste{
    Eleman bas;
    Eleman son;
    public Liste(){
        bas = null;
        son = null;
    }
}
```



[Giriş](#)

[Bağlı Liste Tanımı](#)

[Temel Bağlı Liste İşlemleri](#)

[İki Listeyi Birleştirme](#)

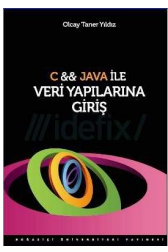
[Çift Bağlı Liste Tanımı](#)

[Temel Çift Bağlı Liste İşlemleri](#)

[Dairesel Bağlı Liste](#)

[Uygulama: Polinom Aritmetiği](#)

# Temel Bağlı Liste İşlemleri



# Bağlı listenin başına ekleme

Giriş

Bağlı Liste Tanımı

Temel Bağlı Liste İşlemleri

İki Listeyi Birleştirme

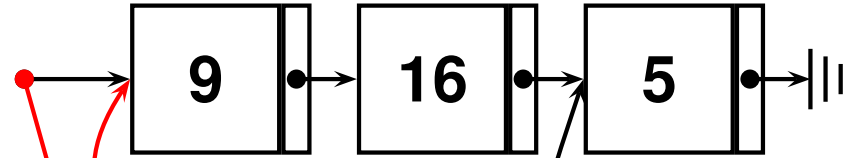
Çift Bağlı Liste Tanımı

Temel Çift Bağlı Liste İşlemleri

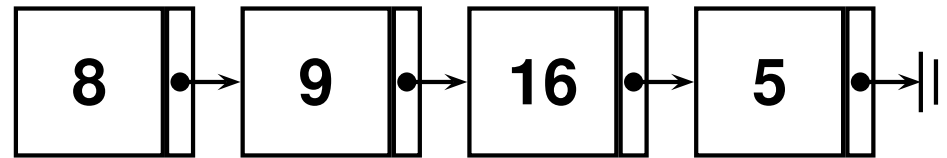
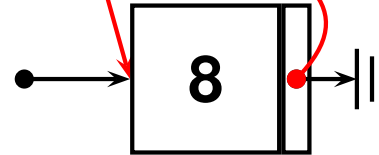
Dairesel Bağlı Liste

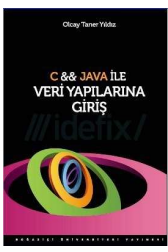
Uygulama: Polinom Aritmetiği

**Liste Başı**



**Yeni**

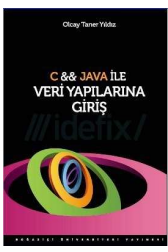




# Bağlı listenin başına eleman ekleyen algoritma

<u>Giriş</u>	1
<u>Bağlı Liste Tanımı</u>	2
<u>Temel Bağlı Liste İşlemleri</u>	3
<u>İki Listeyi Birleştirme</u>	4
<u>Çift Bağlı Liste Tanımı</u>	5
<u>Temel Çift Bağlı Liste İşlemleri</u>	6
<u>Dairesel Bağlı Liste</u>	
<u>Uygulama: Polinom Aritmetiği</u>	

```
void listeBasinaEkle(Eleman yeni){  
    if (son == null)  
        son = yeni;  
    yeni. ileri = bas;  
    bas = yeni;  
}
```



# Bağlı listenin sonuna ekleme

Giriş

Bağlı Liste Tanımı

Temel Bağlı Liste İşlemleri

İki Listeyi Birleştirme

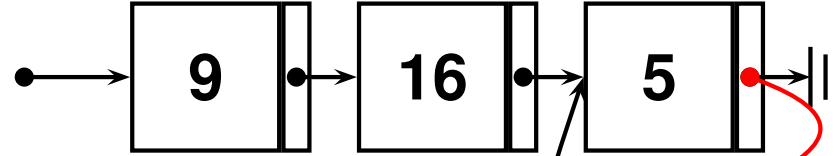
Çift Bağlı Liste Tanımı

Temel Çift Bağlı Liste İşlemleri

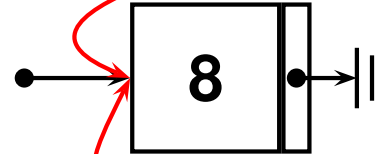
Dairesel Bağlı Liste

Uygulama: Polinom Aritmetiği

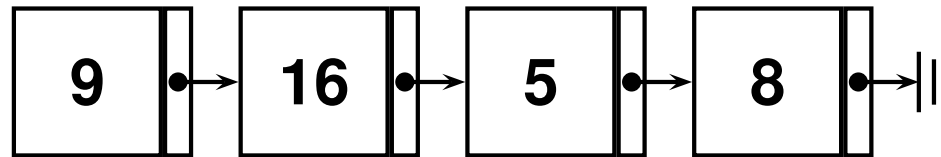
**Liste Başı**

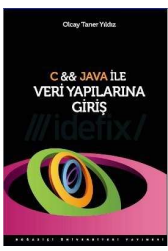


**Yeni**



**Liste Sonu**

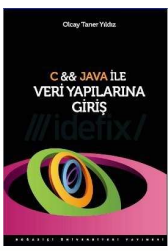




# Bağlı listenin sonuna eleman ekleyen algoritma

<u>Giriş</u>	1
<u>Bağlı Liste Tanımı</u>	2
<u>Temel Bağlı Liste İşlemleri</u>	3
<u>İki Listeyi Birleştirme</u>	4
<u>Çift Bağlı Liste Tanımı</u>	5
<u>Temel Çift Bağlı Liste İşlemleri</u>	6
<u>Dairesel Bağlı Liste</u>	7
<u>Uygulama: Polinom Aritmetiği</u>	

```
void listeyeEkle(Eleman yeni){  
    if (bas == null)  
        bas = yeni;  
    else  
        son. ileri = yeni;  
    son = yeni;  
}
```



# Bağlı listenin ortasına ekleme

Giriş

Bağlı Liste Tanımı

Temel Bağlı Liste İşlemleri

İki Listeyi Birleştirme

Çift Bağlı Liste Tanımı

Temel Çift Bağlı Liste İşlemleri

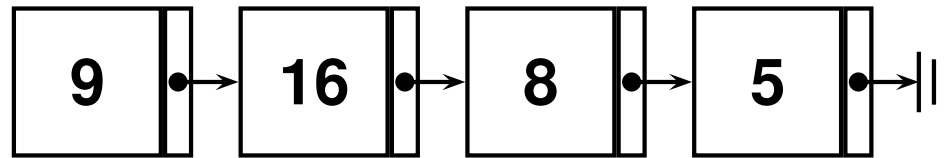
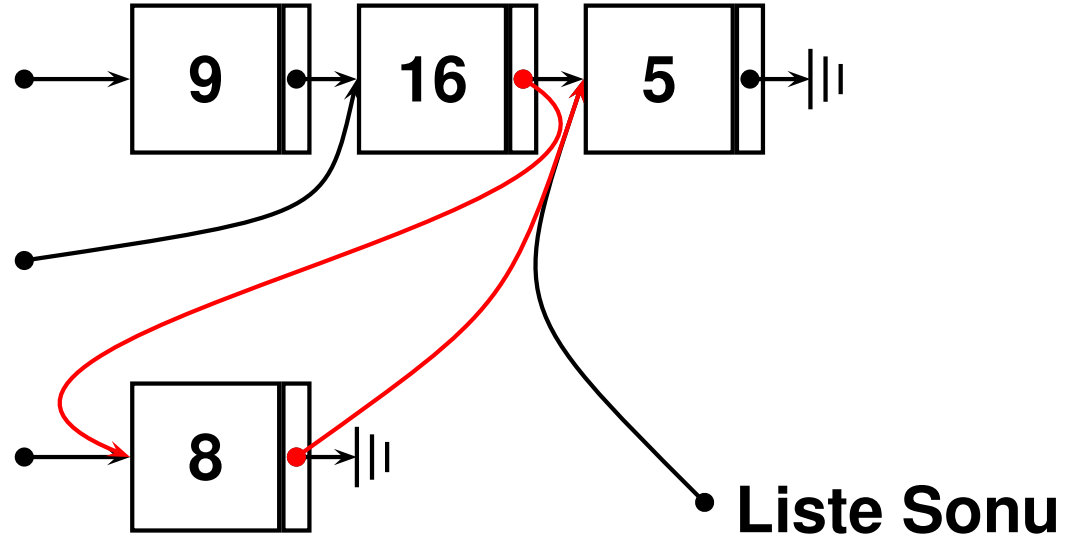
Dairesel Bağlı Liste

Uygulama: Polinom Aritmetiği

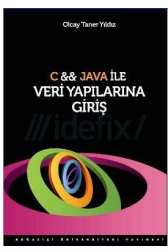
**Liste Başı**

**Önce**

**Yeni**







# Bağlı listenin ortasına eleman ekleyen algoritma

<u>Giriş</u>	1
<u>Bağlı Liste Tanımı</u>	2
<u>Temel Bağlı Liste İşlemleri</u>	3
<u>İki Listeyi Birleştirme</u>	4
<u>Çift Bağlı Liste Tanımı</u>	
<u>Temel Çift Bağlı Liste İşlemleri</u>	
<u>Dairesel Bağlı Liste</u>	
<u>Uygulama: Polinom Aritmetiği</u>	

```
void listeOrtaEkle(Eleman yeni, Eleman once){
    yeni. ileri = once. ileri ;
    once. ileri = yeni;
}
```



# Bağlı Liste İşlemleri

Giriş

Bağlı Liste Tanımı

Temel Bağlı Liste İşlemleri

İki Listeyi Birleştirme

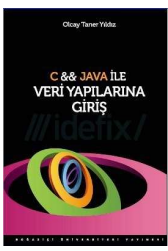
Çift Bağlı Liste Tanımı

Temel Çift Bağlı Liste İşlemleri

Dairesel Bağlı Liste

Uygulama: Polinom Aritmetiği

- Liste başına ekleme:  $O(1)$
- Liste sonuna ekleme:  $O(1)$
- Liste ortasına ekleme:  $O(1)$



# Bağlı listede verilen bir değeri arama

<u>Giriş</u>	1
<u>Bağlı Liste Tanımı</u>	2
<u>Temel Bağlı Liste İşlemleri</u>	3
<u>İki Listeyi Birleştirme</u>	4
<u>Çift Bağlı Liste Tanımı</u>	5
<u>Temel Çift Bağlı Liste İşlemleri</u>	6
<u>Dairesel Bağlı Liste</u>	7
<u>Uygulama: Polinom Aritmetiği</u>	8
	9
	10

```
Eleman listeAra(int deger){
    Eleman tmp;
    tmp = bas;
    while (tmp != null){
        if (tmp.icerik == deger)
            return tmp;
        tmp = tmp.ileri ;
    }
    return null;
}
```



# Bağlı listenin $i$ 'ninci elemanını döndürme

<u>Giriş</u>	1
<u>Bağlı Liste Tanımı</u>	2
<u>Temel Bağlı Liste İşlemleri</u>	3
<u>İki Listeyi Birleştirme</u>	4
<u>Çift Bağlı Liste Tanımı</u>	5
<u>Temel Çift Bağlı Liste İşlemleri</u>	6
<u>Dairesel Bağlı Liste</u>	7
<u>Uygulama: Polinom Aritmetiği</u>	8
	9

```
Eleman elemanl(int i){
    Eleman tmp = bas;
    int j = 0;
    while (tmp != null && j < i){
        j++;
        tmp = tmp.ileri ;
    }
    return tmp;
}
```



# Bağlı Liste İşlemleri

Giriş

Bağlı Liste Tanımı

Temel Bağlı Liste İşlemleri

İki Listeyi Birleştirme

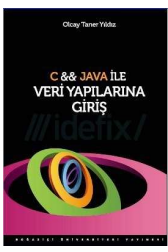
Çift Bağlı Liste Tanımı

Temel Çift Bağlı Liste İşlemleri

Dairesel Bağlı Liste

Uygulama: Polinom Aritmetiği

- Listede arama:  $\mathcal{O}(N)$
- $i$ 'ninci elemanı getirme:  $\mathcal{O}(N)$



# Bağlı listenin ilk elemanını silme

Giriş

Bağlı Liste Tanımı

Temel Bağlı Liste İşlemleri

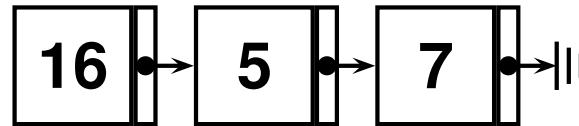
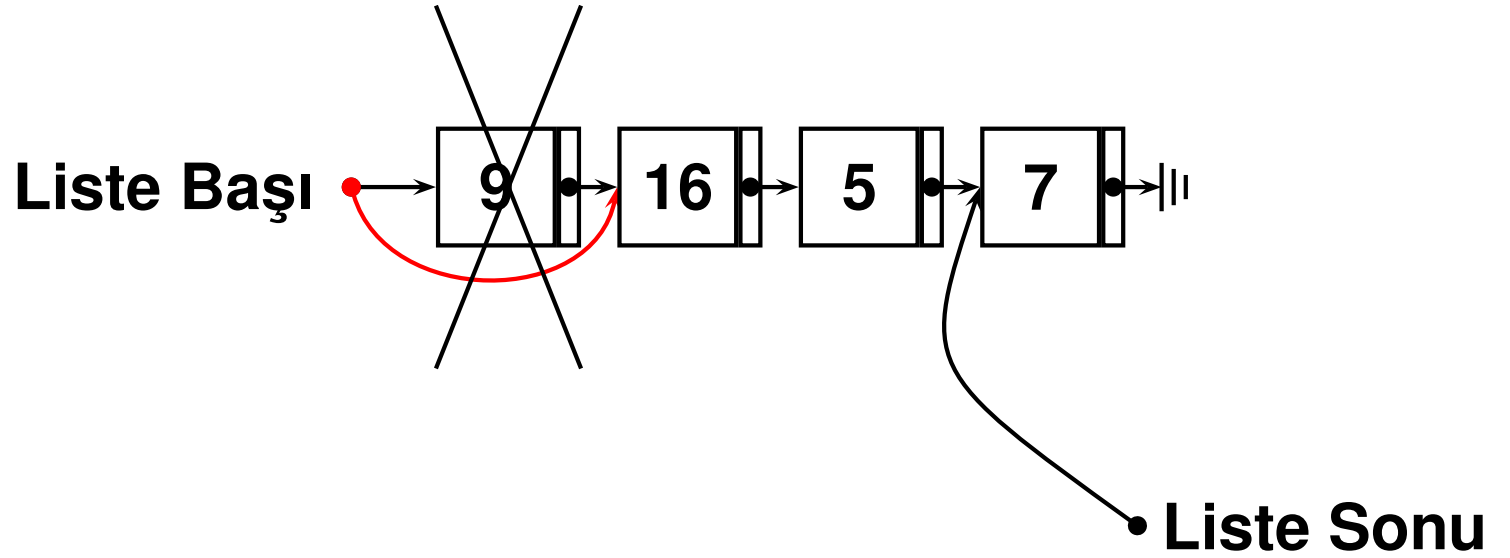
İki Listeyi Birleştirme

Çift Bağlı Liste Tanımı

Temel Çift Bağlı Liste İşlemleri

Dairesel Bağlı Liste

Uygulama: Polinom Aritmetiği





# Bağlı listenin ilk elemanını silme

- Giriş 1
- Bağlı Liste Tanımı 2
- Temel Bağlı Liste İşlemleri 3
- İki Listeyi Birleştirme 5

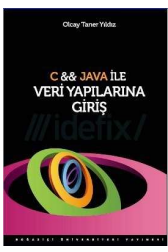
Çift Bağlı Liste Tanımı

Temel Çift Bağlı Liste İşlemleri

Dairesel Bağlı Liste

Uygulama: Polinom Aritmetiği

```
void listeBasiSil (){\n    bas = bas. ileri ;\n    if (bas == null)\n        son = null;\n}
```



# Bağlı listenin son elemanını silme

Giriş

Bağlı Liste Tanımı

Temel Bağlı Liste İşlemleri

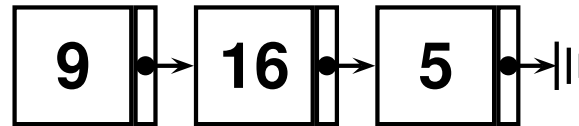
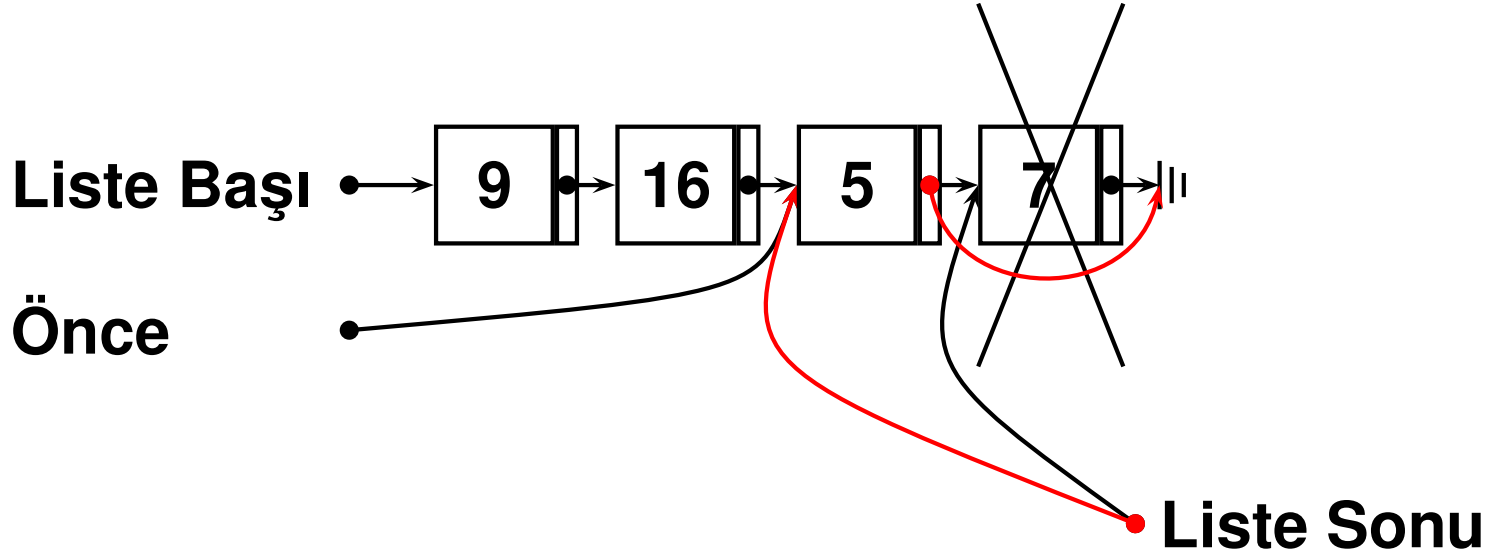
İki Listeyi Birleştirme

Çift Bağlı Liste Tanımı

Temel Çift Bağlı Liste İşlemleri

Dairesel Bağlı Liste

Uygulama: Polinom Aritmetiği



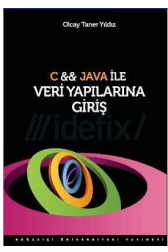




# Bağlı listenin son elemanını silme

<u>Giriş</u>	1
<u>Bağlı Liste Tanımı</u>	2
<u>Temel Bağlı Liste İşlemleri</u>	3
<u>İki Listeyi Birleştirme</u>	4
<u>Çift Bağlı Liste Tanımı</u>	6
<u>Temel Çift Bağlı Liste İşlemleri</u>	7
<u>Dairesel Bağlı Liste</u>	8
<u>Uygulama: Polinom Aritmetiği</u>	9
	10
	11
	12
	13
	14

```
void listeSonuSil(){
    Eleman tmp, once;
    tmp = bas;
    once = null;
    while (tmp != son){
        once = tmp;
        tmp = tmp. ileri ;
    }
    if (once == null)
        bas = null;
    else
        once. ileri = null;
    son = once;
}
```



# Bağlı listenin ortasından silme

Giriş

Bağlı Liste Tanımı

Temel Bağlı Liste İşlemleri

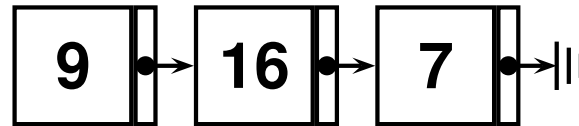
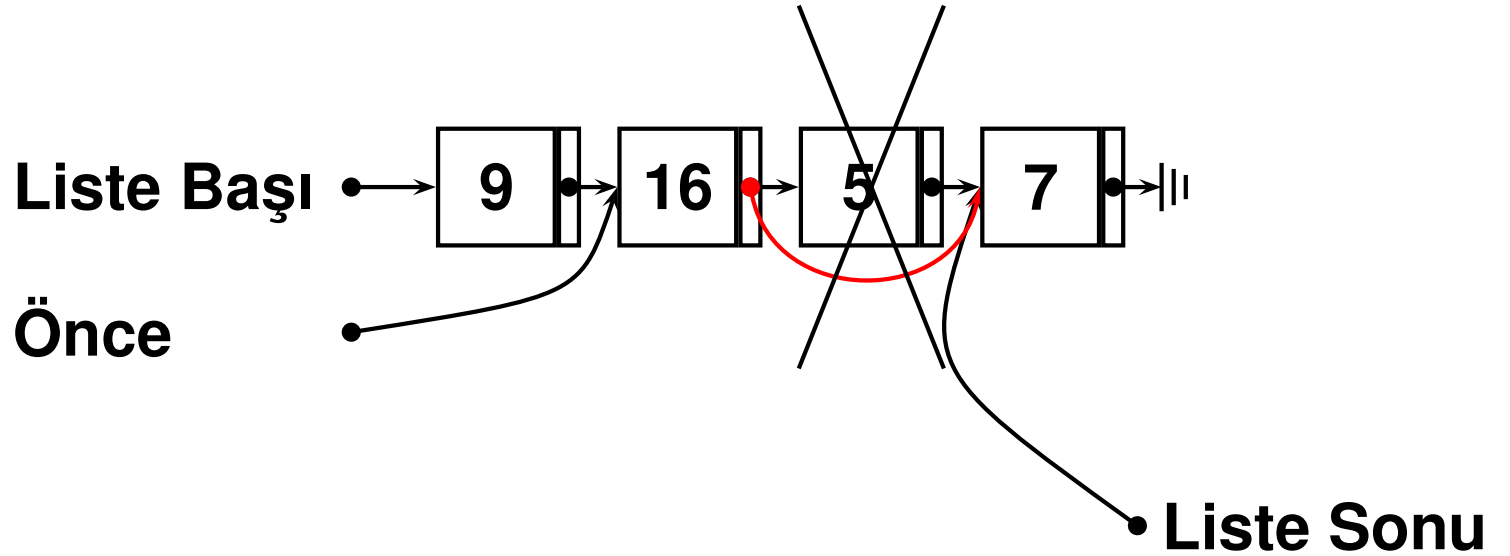
İki Listeyi Birleştirme

Çift Bağlı Liste Tanımı

Temel Çift Bağlı Liste İşlemleri

Dairesel Bağlı Liste

Uygulama: Polinom Aritmetiği





# Bağlı listenin ortasından bir elemanı silme

<u>Giriş</u>	1
<u>Bağlı Liste Tanımı</u>	2
<u>Temel Bağlı Liste İşlemleri</u>	3
<u>İki Listeyi Birleştirme</u>	4
<u>Çift Bağlı Liste Tanımı</u>	5
<u>Temel Çift Bağlı Liste İşlemleri</u>	6
<u>Dairesel Bağlı Liste</u>	7
<u>Uygulama: Polinom Aritmetiği</u>	8
	9
	10

```
void listedenSil(Eleman s){
    Eleman tmp, elemanonce;
    tmp = bas;
    elemanonce = null;
    while (tmp != s){
        elemanonce = tmp;
        tmp = tmp.ileri ;
    }
    elemanonce.ileri = s. ileri ;
}
```



# Bağlı Liste İşlemleri

Giriş

Bağlı Liste Tanımı

Temel Bağlı Liste İşlemleri

İki Listeyi Birleştirme

Çift Bağlı Liste Tanımı

Temel Çift Bağlı Liste İşlemleri

Dairesel Bağlı Liste

Uygulama: Polinom Aritmetiği

- Listenin ilk elemanını silme:  $O(1)$
- Listenin son elemanını silme:  $O(N)$
- Listenin ortasından silme:  $O(N)$



# Bağlı listedeki eleman sayısını bulma

<u>Giriş</u>	1
<u>Bağlı Liste Tanımı</u>	2
<u>Temel Bağlı Liste İşlemleri</u>	3
<u>İki Listeyi Birleştirme</u>	4
<u>Çift Bağlı Liste Tanımı</u>	5
<u>Temel Çift Bağlı Liste İşlemleri</u>	6
<u>Dairesel Bağlı Liste</u>	7
<u>Uygulama: Polinom Aritmetiği</u>	8
	9
	10

```
int elemanSayisi(){
    int sayac = 0;
    Eleman tmp;
    tmp = bas;
    while (tmp != null){
        tmp = tmp.ileri ;
        sayac++;
    }
    return sayac;
}
```



[Giriş](#)

[Bağlı Liste Tanımı](#)

[Temel Bağlı Liste İşlemleri](#)

[İki Listeyi Birleştirme](#)

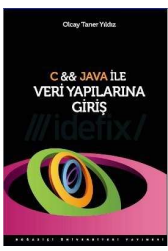
[Çift Bağlı Liste Tanımı](#)

[Temel Çift Bağlı Liste İşlemleri](#)

[Dairesel Bağlı Liste](#)

[Uygulama: Polinom Aritmetiği](#)

# İki Listeyi Birleştirme



# Bağlı listeleri birleştirme

Giriş

Bağlı Liste Tanımı

Temel Bağlı Liste İşlemleri

İki Listeyi Birleştirme

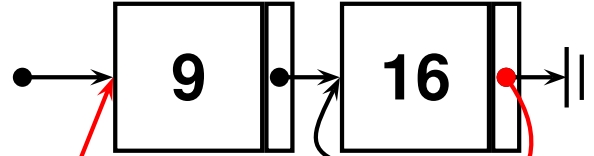
Çift Bağlı Liste Tanımı

Temel Çift Bağlı Liste İşlemleri

Dairesel Bağlı Liste

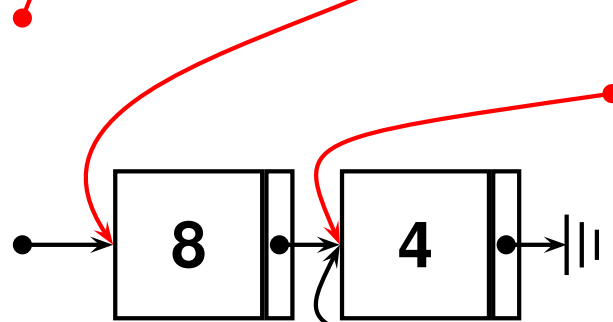
Uygulama: Polinom Aritmetiği

**1. Liste Başı**



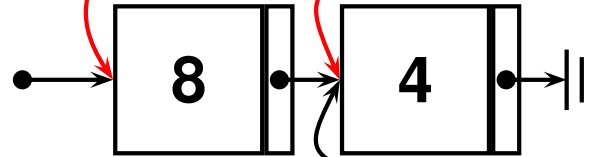
**1. Liste Sonu**

**3. Liste Başı**

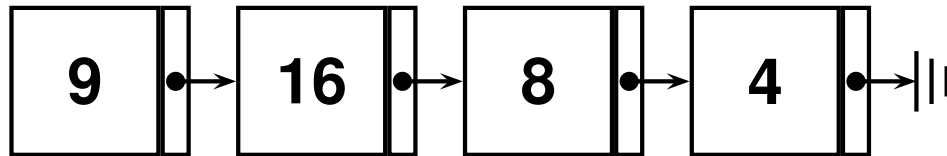


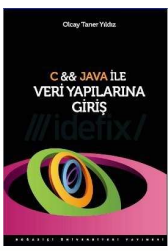
**3. Liste Sonu**

**2. Liste Başı**



**2. Liste Sonu**





# İki bağlı listeyi birleştirme

Giriş	1
Bağlı Liste Tanımı	2
Temel Bağlı Liste İşlemleri	3
İki Listeyi Birleştirme	4
Çift Bağlı Liste Tanımı	5
Temel Çift Bağlı Liste İşlemleri	6
Dairesel Bağlı Liste	7
Uygulama: Polinom Aritmetiği	8
	9
	10
	11
	12

```
Liste birlestir (Liste l1 , Liste l2){
    Liste tmp;
    if (l1.bas == null)
        return l2;
    if (l2.bas == null)
        return l1;
    tmp = new Liste();
    tmp.bas = l1.bas;
    tmp.son = l2.son;
    l1.son.ileri = l2.bas;
    return tmp;
}
```





[Giriş](#)

[Bağlı Liste Tanımı](#)

[Temel Bağlı Liste İşlemleri](#)

[İki Listeyi Birleştirme](#)

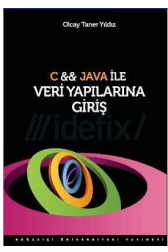
[Çift Bağlı Liste Tanımı](#)

[Temel Çift Bağlı Liste İşlemleri](#)

[Dairesel Bağlı Liste](#)

[Uygulama: Polinom Aritmetiği](#)

## Çift Bağlı Liste Tanımı



# Tam sayı içeren eleman veri yapısı (Çift bağlı liste için)

Giriş

Bağlı Liste Tanımı

Temel Bağlı Liste İşlemleri

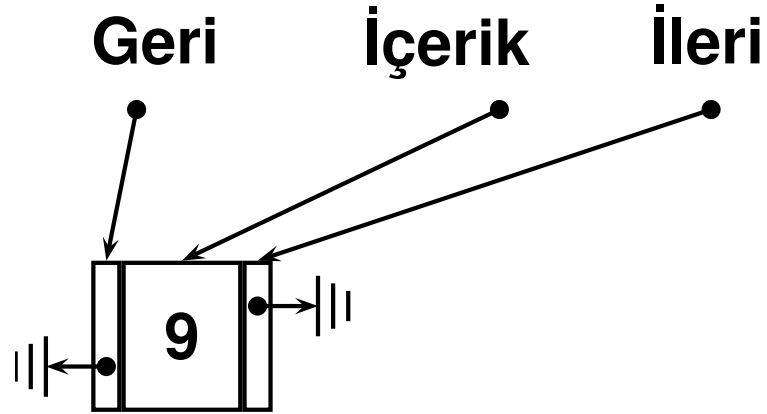
İki Listeyi Birleştirme

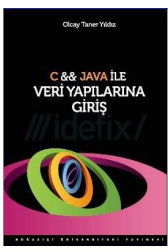
Çift Bağlı Liste Tanımı

Temel Çift Bağlı Liste İşlemleri

Dairesel Bağlı Liste

Uygulama: Polinom Aritmetiği





# Tam sayı içeren bir eleman tanımı (Çift bağlı liste için)

<u>Giriş</u>	1
<u>Bağlı Liste Tanımı</u>	2
<u>Temel Bağlı Liste İşlemleri</u>	3
<u>İki Listeyi Birleştirme</u>	4
<u>Çift Bağlı Liste Tanımı</u>	5
<u>Temel Çift Bağlı Liste İşlemleri</u>	6
<u>Dairesel Bağlı Liste</u>	7
<u>Uygulama: Polinom Aritmetiği</u>	8
	9
	10

```
public class CiftEleman{
    int icerik;
    CiftEleman ileri ;
    CiftEleman geri;
    public CiftEleman(int icerik){
        this.icerik = icerik ;
        ileri = null;
        geri = null;
    }
}
```



# Bir çift bağılı liste ve dizi yapısı

Giriş

Bağılı Liste Tanımı

Temel Bağılı Liste İşlemleri

İki Listeyi Birleştirme

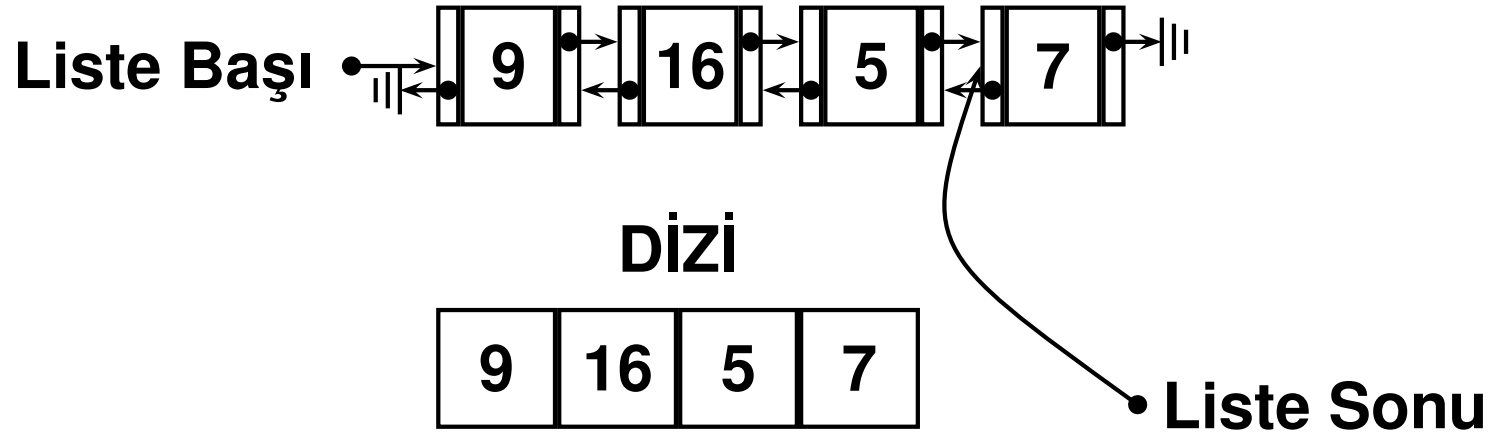
Çift Bağılı Liste Tanımı

Temel Çift Bağılı Liste İşlemleri

Dairesel Bağılı Liste

Uygulama: Polinom Aritmetiği

## ÇİFT BAĞLI LİSTE





# Tam sayılar içeren çift bağlı liste tanımı

<u>Giriş</u>	1
<u>Bağlı Liste Tanımı</u>	2
<u>Temel Bağlı Liste İşlemleri</u>	3
<u>İki Listeyi Birleştirme</u>	4
<u>Çift Bağlı Liste Tanımı</u>	5
<u>Temel Çift Bağlı Liste İşlemleri</u>	6
<u>Dairesel Bağlı Liste</u>	7
<u>Uygulama: Polinom Aritmetiği</u>	8

```
public class CiftListe{
    CiftEleman bas;
    CiftEleman son;
    public CiftListe(){
        bas = null;
        son = null;
    }
}
```



[Giriş](#)

[Bağlı Liste Tanımı](#)

[Temel Bağlı Liste İşlemleri](#)

[İki Listeyi Birleştirme](#)

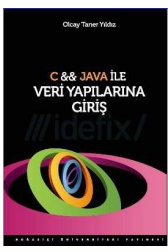
[Çift Bağlı Liste Tanımı](#)

[Temel Çift Bağlı Liste İşlemleri](#)

[Dairesel Bağlı Liste](#)

[Uygulama: Polinom Aritmetiği](#)

# Temel Çift Bağlı Liste İşlemleri



# Çift bağılı listenin başına eleman ekleme

Giriş

Bağılı Liste Tanımı

Temel Bağılı Liste İşlemleri

İki Listeyi Birleştirme

Çift Bağılı Liste Tanımı

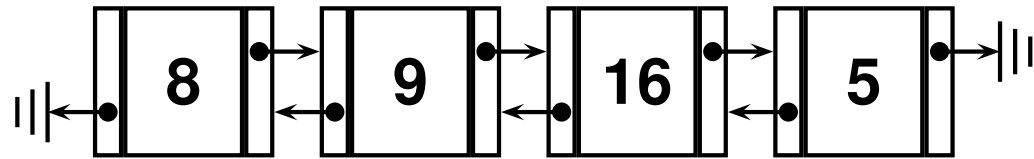
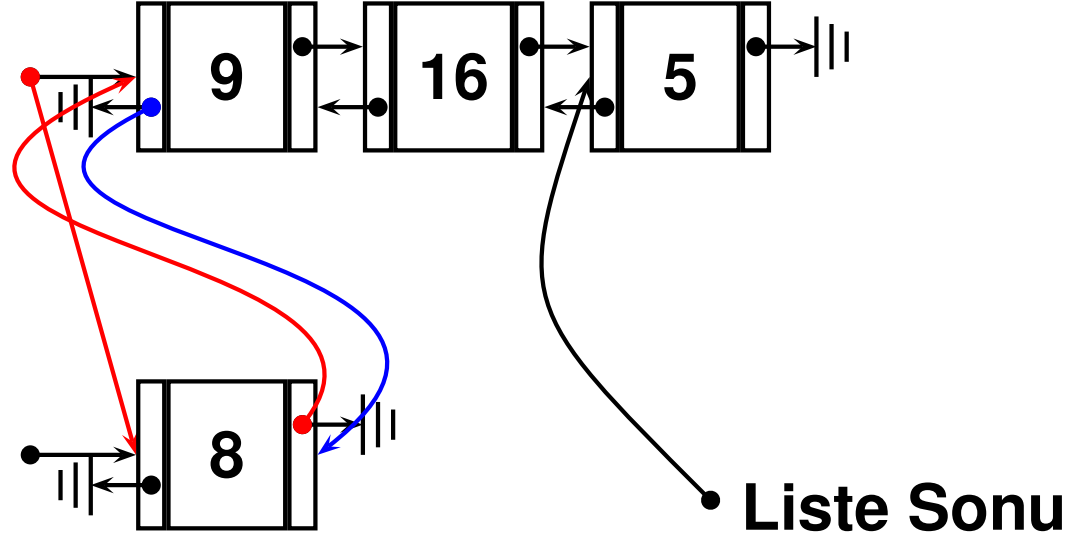
Temel Çift Bağılı Liste İşlemleri

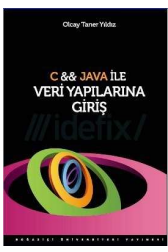
Dairesel Bağılı Liste

Uygulama: Polinom Aritmetiği

**Liste Başı**

**Yeni**



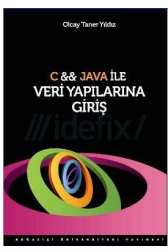


# Çift bağlı listenin başına eleman ekleyen algoritma

<u>Giriş</u>	1
<u>Bağlı Liste Tanımı</u>	2
<u>Temel Bağlı Liste İşlemleri</u>	3
<u>İki Listeyi Birleştirme</u>	4
<u>Çift Bağlı Liste Tanımı</u>	5
<u>Temel Çift Bağlı Liste İşlemleri</u>	6
<u>Dairesel Bağlı Liste</u>	7
<u>Uygulama: Polinom Aritmetiği</u>	8

```
void listeBasinaEkle(CiftEleman yeni){
    if (son == null)
        son = yeni;
    else
        bas.geri = yeni;
    yeni. ileri = bas;
    bas = yeni;
}
```





# Çift bağılı listenin sonuna eleman ekleme

Giriş

Bağılı Liste Tanımı

Temel Bağılı Liste İşlemleri

İki Listeyi Birleştirme

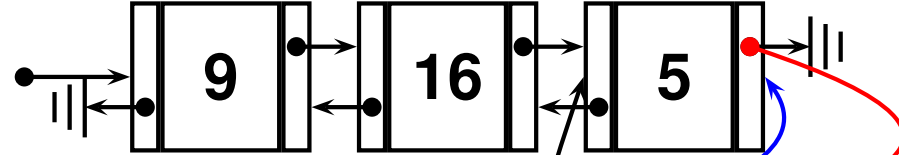
Çift Bağılı Liste Tanımı

Temel Çift Bağılı Liste İşlemleri

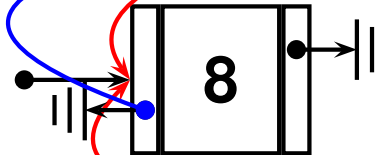
Dairesel Bağılı Liste

Uygulama: Polinom Aritmetiği

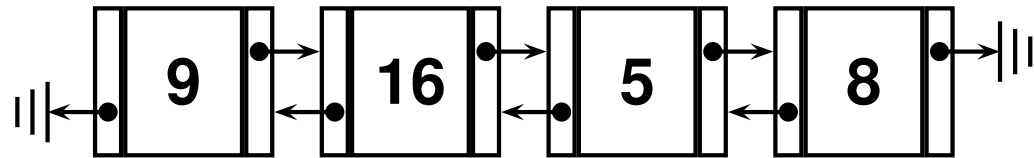
**Liste Başı**

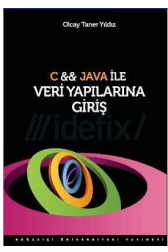


**Yeni**



**Liste Sonu**

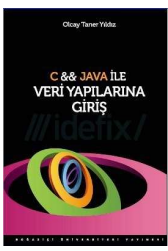




# Çift bağılı listenin sonuna eleman ekleyen algoritma

<u>Giriş</u>	1
<u>Bağılı Liste Tanımı</u>	2
<u>Temel Bağılı Liste İşlemleri</u>	3
<u>İki Listeyi Birleştirme</u>	4
<u>Çift Bağılı Liste Tanımı</u>	5
<u>Temel Çift Bağılı Liste İşlemleri</u>	6
<u>Dairesel Bağılı Liste</u>	7
<u>Uygulama: Polinom Aritmetiği</u>	8

```
void listeyeEkle(CiftEleman yeni){  
    if (bas == null)  
        bas = yeni;  
    else  
        son. ileri = yeni;  
    yeni.geri = son;  
    son = yeni;  
}
```



# Çift bağılı listenin ortasına eleman ekleme

Giriş

Bağılı Liste Tanımı

Temel Bağılı Liste İşlemleri

İki Listeyi Birleştirme

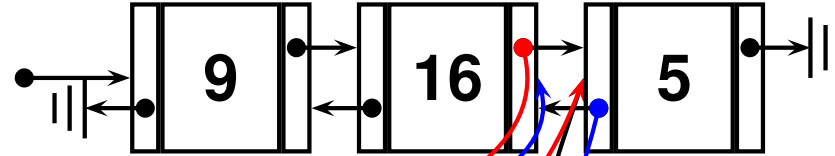
Çift Bağılı Liste Tanımı

Temel Çift Bağılı Liste İşlemleri

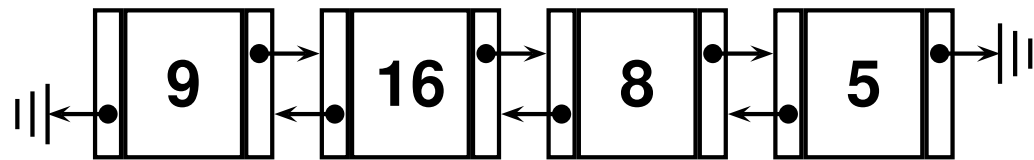
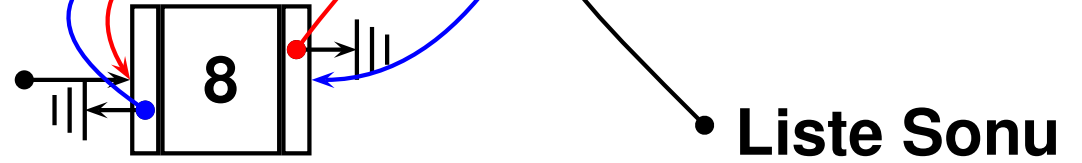
Dairesel Bağılı Liste

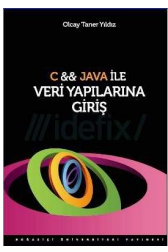
Uygulama: Polinom Aritmetiği

**Liste Başı**



**Yeni**





# Çift bağılı listenin ortasına eleman ekleyen algoritma

<u>Giriş</u>	1
<u>Bağılı Liste Tanımı</u>	2
<u>Temel Bağılı Liste İşlemleri</u>	3
<u>İki Listeyi Birleştirme</u>	4
<u>Çift Bağılı Liste Tanımı</u>	5
<u>Temel Çift Bağılı Liste İşlemleri</u>	6
<u>Dairesel Bağılı Liste</u>	
<u>Uygulama: Polinom Aritmetiği</u>	

```
void listeOrtaEkle(CiftEleman yeni, CiftEleman once){
    yeni.ileri = once.ileri ;
    yeni.geri = once;
    once.ileri .geri = yeni;
    once.ileri = yeni;
}
```



# Çift Bağlı Liste İşlemleri

Giriş

Bağlı Liste Tanımı

Temel Bağlı Liste İşlemleri

İki Listeyi Birleştirme

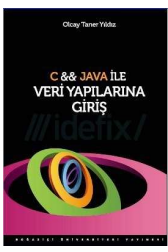
Çift Bağlı Liste Tanımı

Temel Çift Bağlı Liste İşlemleri

Dairesel Bağlı Liste

Uygulama: Polinom Aritmetiği

- Listenin başına ekleme:  $O(1)$
- Listenin sonuna ekleme:  $O(1)$
- Listenin ortasına ekleme:  $O(1)$



# Çift bağlı listenin ilk elemanını silme

Giriş

Bağlı Liste Tanımı

Temel Bağlı Liste İşlemleri

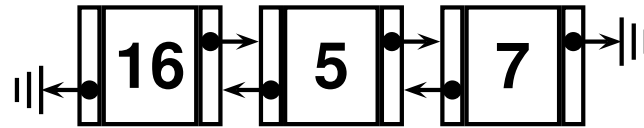
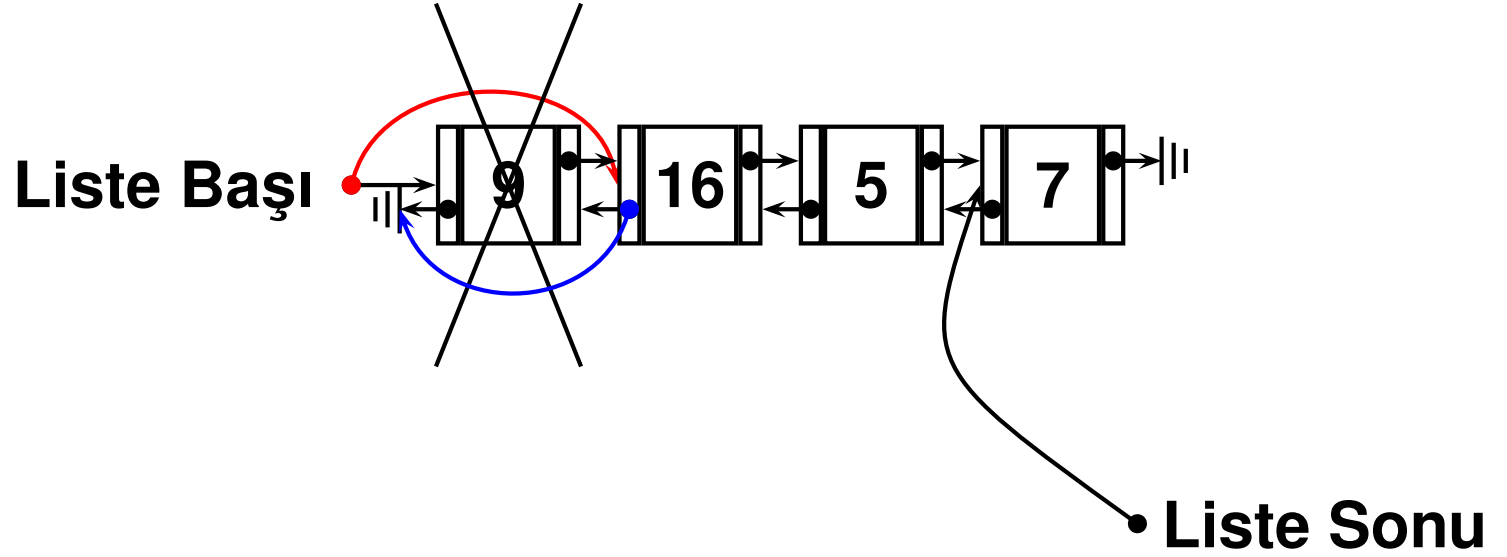
İki Listeyi Birleştirme

Çift Bağlı Liste Tanımı

Temel Çift Bağlı Liste İşlemleri

Dairesel Bağlı Liste

Uygulama: Polinom Aritmetiği

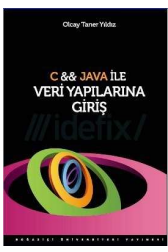




# Çift bağılı listenin ilk elemanını silme

<u>Giriş</u>	1
<u>Bağılı Liste Tanımı</u>	2
<u>Temel Bağılı Liste İşlemleri</u>	3
<u>İki Listeyi Birleştirme</u>	4
<u>Çift Bağılı Liste Tanımı</u>	5
<u>Temel Çift Bağılı Liste İşlemleri</u>	6
<u>Dairesel Bağılı Liste</u>	7
<u>Uygulama: Polinom Aritmetiği</u>	

```
void listeBasiSil (){\n    bas = bas. ileri ;\n    if (bas == null)\n        son = null;\n    else\n        bas.geri = null;\n}
```



# Çift bağılı listenin son elemanını silme

Giriş

Bağılı Liste Tanımı

Temel Bağılı Liste İşlemleri

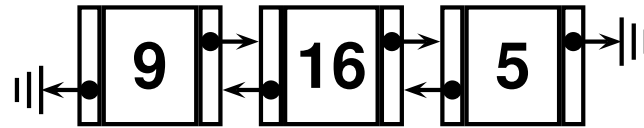
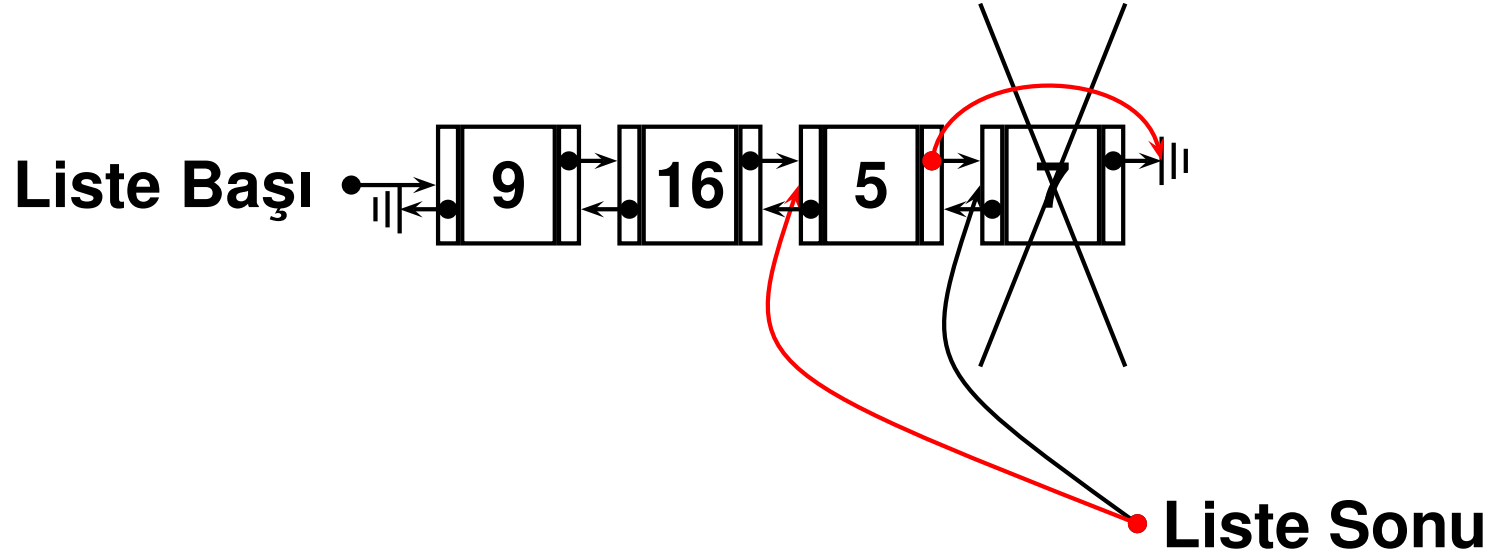
İki Listeyi Birleştirme

Çift Bağılı Liste Tanımı

Temel Çift Bağılı Liste İşlemleri

Dairesel Bağılı Liste

Uygulama: Polinom Aritmetiği



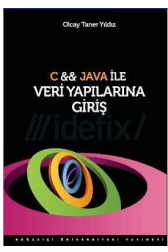




# Çift bağlı listenin son elemanını silme

<u>Giriş</u>	1
<u>Bağlı Liste Tanımı</u>	2
<u>Temel Bağlı Liste İşlemleri</u>	3
<u>İki Listeyi Birleştirme</u>	4
<u>Çift Bağlı Liste Tanımı</u>	5
<u>Temel Çift Bağlı Liste İşlemleri</u>	6
<u>Dairesel Bağlı Liste</u>	7
<u>Uygulama: Polinom Aritmetiği</u>	

```
void listeSonuSil(){
    son = son.geri;
    if (son == null)
        bas = null;
    else
        son.ileri = null;
}
```



# Çift bağılı listenin ortasından silme

Giriş

Bağılı Liste Tanımı

Temel Bağılı Liste İşlemleri

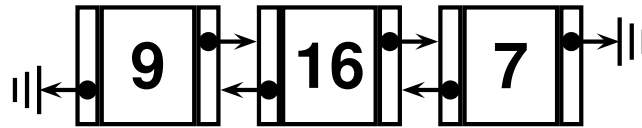
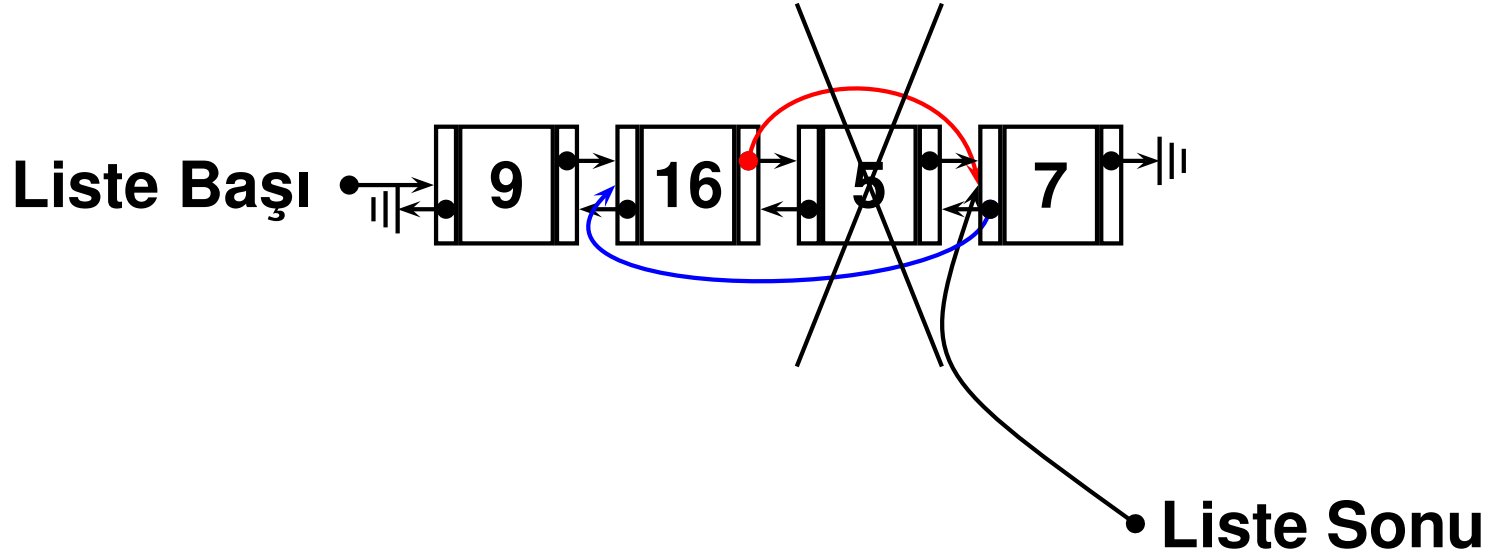
İki Listeyi Birleştirme

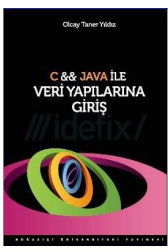
Çift Bağılı Liste Tanımı

Temel Çift Bağılı Liste İşlemleri

Dairesel Bağılı Liste

Uygulama: Polinom Aritmetiği





# Çift bağlı listenin ortasından bir elemanı silme

<u>Giriş</u>	1
<u>Bağlı Liste Tanımı</u>	2
<u>Temel Bağlı Liste İşlemleri</u>	3
<u>İki Listeyi Birleştirme</u>	4
<u>Çift Bağlı Liste Tanımı</u>	
<u>Temel Çift Bağlı Liste İşlemleri</u>	
<u>Dairesel Bağlı Liste</u>	
<u>Uygulama: Polinom Aritmetiği</u>	

```
void listedenSil(CiftEleman s){  
    s.ileri .geri = s.geri ;  
    s.geri .ileri = s.ileri ;  
}
```



# Çift Bağlı Liste İşlemleri

Giriş

Bağlı Liste Tanımı

Temel Bağlı Liste İşlemleri

İki Listeyi Birleştirme

Çift Bağlı Liste Tanımı

Temel Çift Bağlı Liste İşlemleri

Dairesel Bağlı Liste

Uygulama: Polinom Aritmetiği

- Listenin ilk elemanını silme:  $O(1)$
- Listenin son elemanını silme:  $O(1)$
- Listenin ortasından silme:  $O(1)$



[Giriş](#)

[Bağlı Liste Tanımı](#)

[Temel Bağlı Liste İşlemleri](#)

[İki Listeyi Birleştirme](#)

[Çift Bağlı Liste Tanımı](#)

[Temel Çift Bağlı Liste İşlemleri](#)

[Dairesel Bağlı Liste](#)

[Uygulama: Polinom Aritmetiği](#)

# Dairesel Bağlı Liste



# Bir dairesel bağılı liste ve dizi yapısı

Giriş

Bağılı Liste Tanımı

Temel Bağılı Liste İşlemleri

İki Listeyi Birleştirme

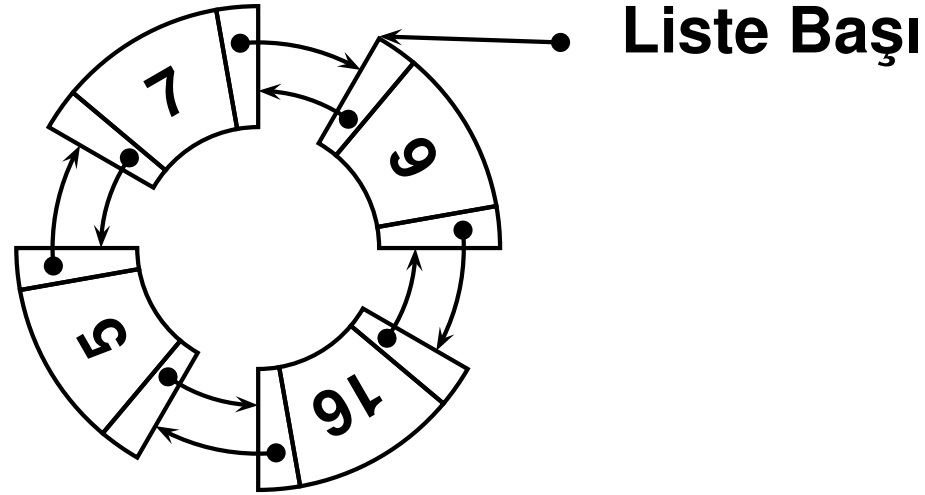
Çift Bağılı Liste Tanımı

Temel Çift Bağılı Liste İşlemleri

Dairesel Bağılı Liste

Uygulama: Polinom Aritmetiği

## DAİRESEL BAĞLI LİSTE



## DİZİ

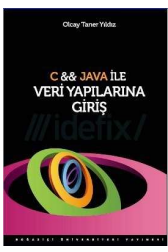
9	16	5	7
---	----	---	---



# Tam sayılar içeren dairesel bağlı liste tanımı

<u>Giriş</u>	1
<u>Bağlı Liste Tanımı</u>	2
<u>Temel Bağlı Liste İşlemleri</u>	3
<u>İki Listeyi Birleştirme</u>	4
<u>Çift Bağlı Liste Tanımı</u>	5
<u>Temel Çift Bağlı Liste İşlemleri</u>	6
<u>Dairesel Bağlı Liste</u>	
<u>Uygulama: Polinom Aritmetiği</u>	

```
public class DaireListe{  
    CiftEleman bas;  
    public DaireListe(){  
        bas = null;  
    }  
}
```



# Dairesel bağlı listenin başına 8 eklenmesi

Giriş

Bağlı Liste Tanımı

Temel Bağlı Liste İşlemleri

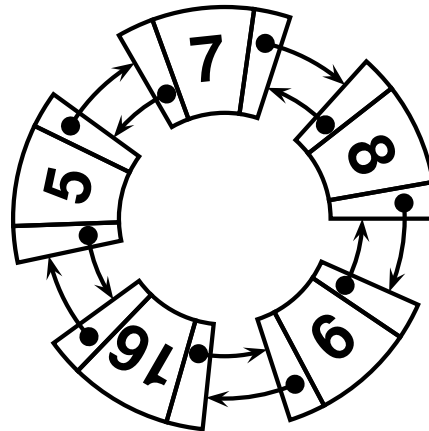
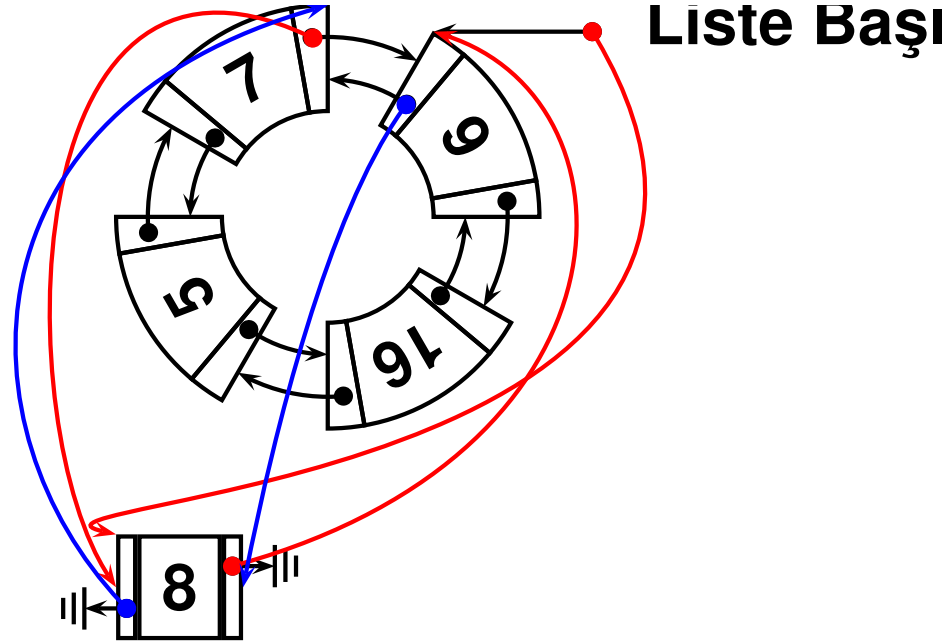
İki Listeyi Birleştirme

Çift Bağlı Liste Tanımı

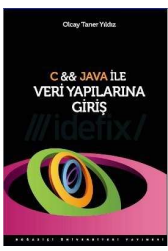
Temel Çift Bağlı Liste İşlemleri

Dairesel Bağlı Liste

Uygulama: Polinom Aritmetiği



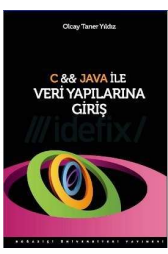




# Dairesel bağlı listenin başına eleman ekleyen algoritma

Giriş	1
Bağlı Liste Tanımı	2
Temel Bağlı Liste İşlemleri	3
İki Listeyi Birleştirme	4
Çift Bağlı Liste Tanımı	5
Temel Çift Bağlı Liste İşlemleri	6
Dairesel Bağlı Liste	7
Uygulama: Polinom Aritmetiği	8
	9
	10
	11
	12

```
void listeEkle(CiftEleman yeni){
    if (bas == null){
        yeni.ileri = yeni;
        yeni.geri = yeni;
    }else{
        yeni.ileri = bas;
        yeni.geri = bas.geri;
        bas.geri.ileri = yeni;
        bas.geri = yeni;
    }
    bas = yeni;
}
```



# Örnek bir dairesel bağlı listenin ilk elemanını silme

Giriş

Bağlı Liste Tanımı

Temel Bağlı Liste İşlemleri

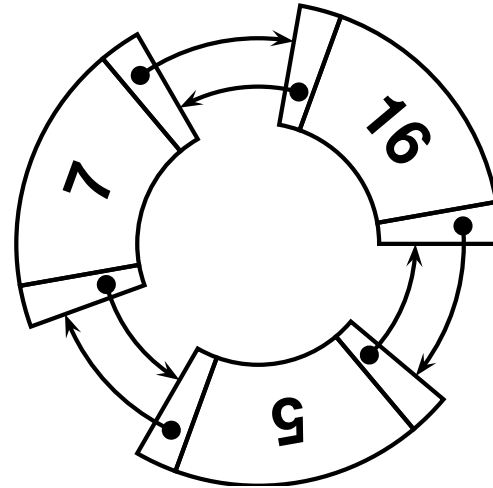
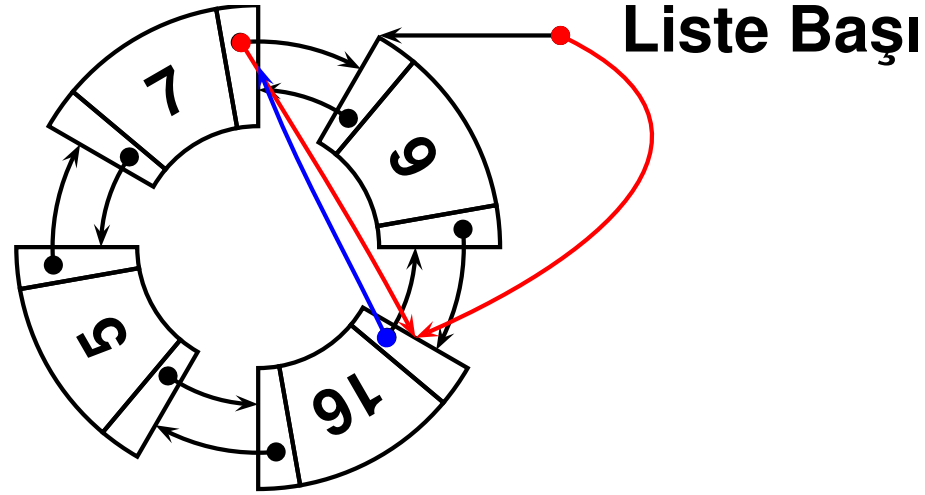
İki Listeyi Birleştirme

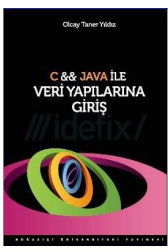
Çift Bağlı Liste Tanımı

Temel Çift Bağlı Liste İşlemleri

Dairesel Bağlı Liste

Uygulama: Polinom Aritmetiği





# Dairesel bağlı listenin ilk elemanı silen algoritma

<u>Giriş</u>	1
<u>Bağlı Liste Tanımı</u>	2
<u>Temel Bağlı Liste İşlemleri</u>	3
<u>İki Listeyi Birleştirme</u>	4
<u>Çift Bağlı Liste Tanımı</u>	5
<u>Temel Çift Bağlı Liste İşlemleri</u>	6
<u>Dairesel Bağlı Liste</u>	7
<u>Uygulama: Polinom Aritmetiği</u>	8

```
void listeSil (){
    if (bas. ileri == bas)
        bas = null;
    else{
        bas.geri. ileri = bas. ileri ;
        bas. ileri .geri = bas.geri;
        bas = bas. ileri ;
    }
}
```



[Giriş](#)

[Bağlı Liste Tanımı](#)

[Temel Bağlı Liste İşlemleri](#)

[İki Listeyi Birleştirme](#)

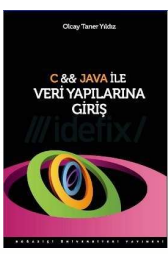
[Çift Bağlı Liste Tanımı](#)

[Temel Çift Bağlı Liste İşlemleri](#)

[Dairesel Bağlı Liste](#)

[Uygulama: Polinom Aritmetiği](#)

## Uygulama: Polinom Aritmetiği



# $4x^5 + 3x^4 + 6x^2 - 8x - 7$ polinomunun sabit dizi ile gösterimi

4	3	0	6	-8	-7
---	---	---	---	----	----

[Giriş](#)

[Bağlı Liste Tanımı](#)

[Temel Bağlı Liste İşlemleri](#)

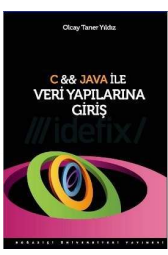
[İki Listeyi Birleştirme](#)

[Çift Bağlı Liste Tanımı](#)

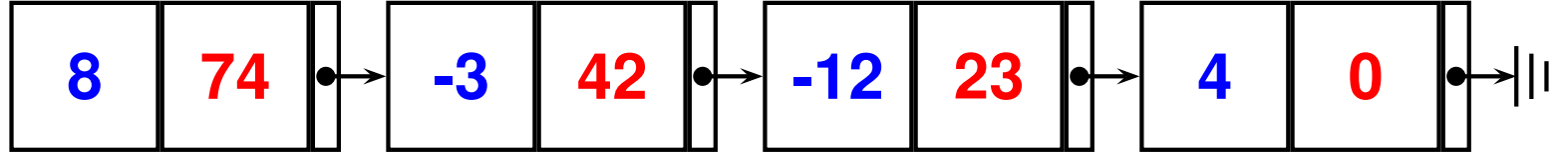
[Temel Çift Bağlı Liste İşlemleri](#)

[Dairesel Bağlı Liste](#)

[Uygulama: Polinom Aritmetiği](#)



# $8x^{74} - 3x^{42} - 12x^{23} + 4$ polinomunun bağlı liste ile gösterimi



[Giriş](#)

[Bağlı Liste Tanımı](#)

[Temel Bağlı Liste İşlemleri](#)

[İki Listeyi Birleştirme](#)

[Çift Bağlı Liste Tanımı](#)

[Temel Çift Bağlı Liste İşlemleri](#)

[Dairesel Bağlı Liste](#)

[Uygulama: Polinom Aritmetiği](#)



# Tek deęişkenli polinomun bir teriminin tanımı

<u>Giriş</u>	1
<u>Baęlı Liste Tanımı</u>	2
<u>Temel Baęlı Liste İşlemleri</u>	3
<u>İki Listeyi Birleştirme</u>	4
<u>Çift Baęlı Liste Tanımı</u>	5
<u>Temel Çift Baęlı Liste İşlemleri</u>	6
<u>Dairesel Baęlı Liste</u>	7
<u>Uygulama: Polinom Aritmetięi</u>	8

```
public class Eleman{
    int katsayi;
    int us;
    Eleman ileri ;
    public Eleman(int katsayi, int us){
        this.katsayi = katsayi;
        this.us = us;
        ileri = null;
    }
}
```



# $4x^5 + 3x^2 - 7x + 8$ ve $2x^4 + 6x^2 + 7x$ *polinomlarının toplanması*

Giriş

Bağlı Liste Tanımı

Temel Bağlı Liste İşlemleri

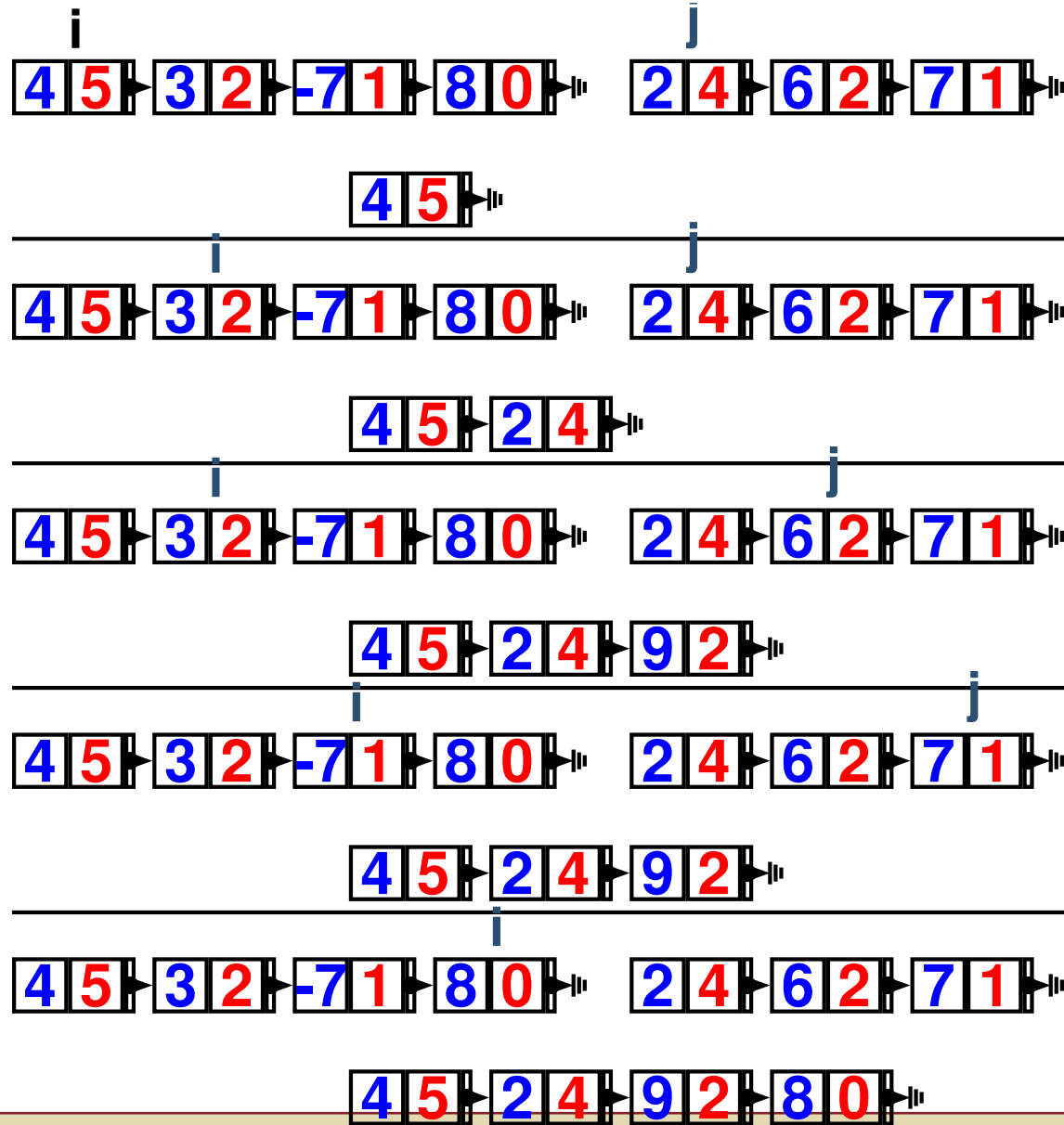
İki Listeyi Birleştirme

Çift Bağlı Liste Tanımı

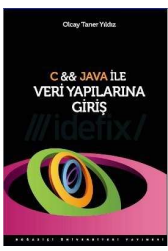
Temel Çift Bağlı Liste İşlemleri

Dairesel Bağlı Liste

Uygulama: Polinom Aritmetiği



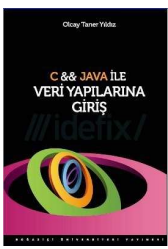




# İki polinomun toplamını bulan algoritma (1)

Giriş	1
Bağlı Liste Tanımı	2
Temel Bağlı Liste İşlemleri	3
İki Listeyi Birleştirme	4
Çift Bağlı Liste Tanımı	5
Temel Çift Bağlı Liste İşlemleri	6
Dairesel Bağlı Liste	7
Uygulama: Polinom Aritmetiği	8
	9
	10
	11
	12
	13
	14
	15
	16
	17
	18

```
Liste topla(Liste p1, Liste p2){
    Eleman i, j, k, yeni;
    Liste sonuc;
    int katsayi, us;
    i = p1.bas;
    j = p2.bas;
    sonuc = new Liste();
    while (i != null && j != null){
        if (i.us == j.us){
            katsayi = i.katsayi + j.katsayi;
            us = i.us;
            i = i.ileri ;
            j = j.ileri ;
        } else
            if (i.us > j.us){
                katsayi = i.katsayi;
                us = i.us;
                i = i.ileri ;
            }
    }
}
```



# İki polinomun toplamını bulan algoritma (2)

Giriş	19
Bağlı Liste Tanımı	20
Temel Bağlı Liste İşlemleri	21
İki Listeyi Birleştirme	22
Çift Bağlı Liste Tanımı	23
Temel Çift Bağlı Liste İşlemleri	24
Dairesel Bağlı Liste	25
Uygulama: Polinom Aritmetiği	26
	27
	28
	29
	30
	31
	32
	33
	34
	35
	36
	37
	38

```
    } else {
        katsayi = j.katsayi;
        us = j.us;
        j = j.ileri ;
    }
    if (katsayi != 0){
        yeni = new Eleman(katsayi, us);
        sonuc.listeyeEkle(yeni);
    }
}
if (i == null)
    k = j;
else
    k = i;
while (k != null){
    yeni = new Eleman(k.katsayi, k.us);
    sonuc.listeyeEkle(yeni);
}
return sonuc;
}
```



# $x^2 + 2x + 1$ ve $x^2 - 2x$ polinomlarının çarpılması (1. Aşama)

Giriş

Bağlı Liste Tanımı

Temel Bağlı Liste İşlemleri

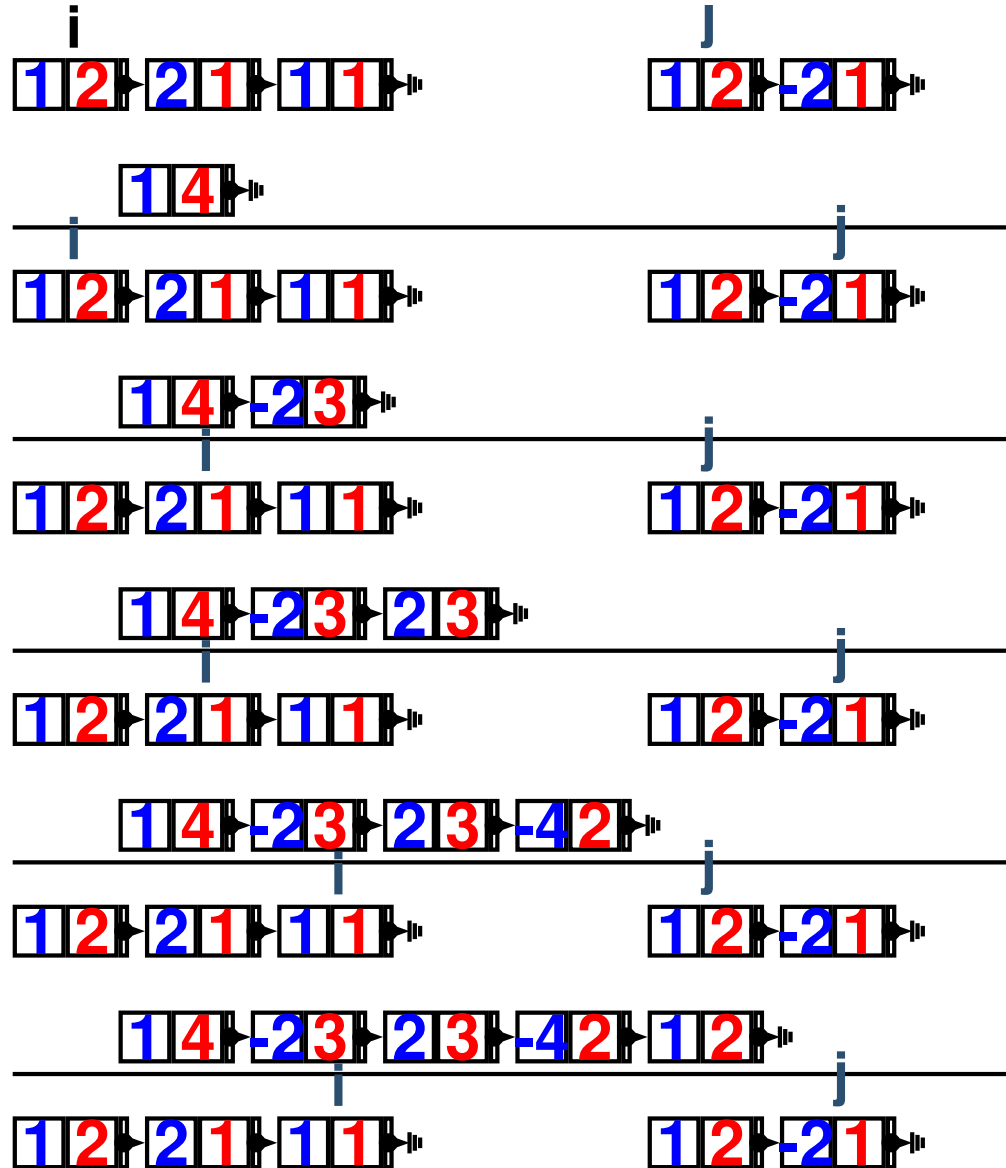
İki Listeyi Birleştirme

Çift Bağlı Liste Tanımı

Temel Çift Bağlı Liste İşlemleri

Dairesel Bağlı Liste

Uygulama: Polinom Aritmetiği





# $x^2 + 2x + 1$ ve $x^2 - 2x$ polinomlarının çarpılması (2. Aşama)

Giriş

Bağlı Liste Tanımı

Temel Bağlı Liste İşlemleri

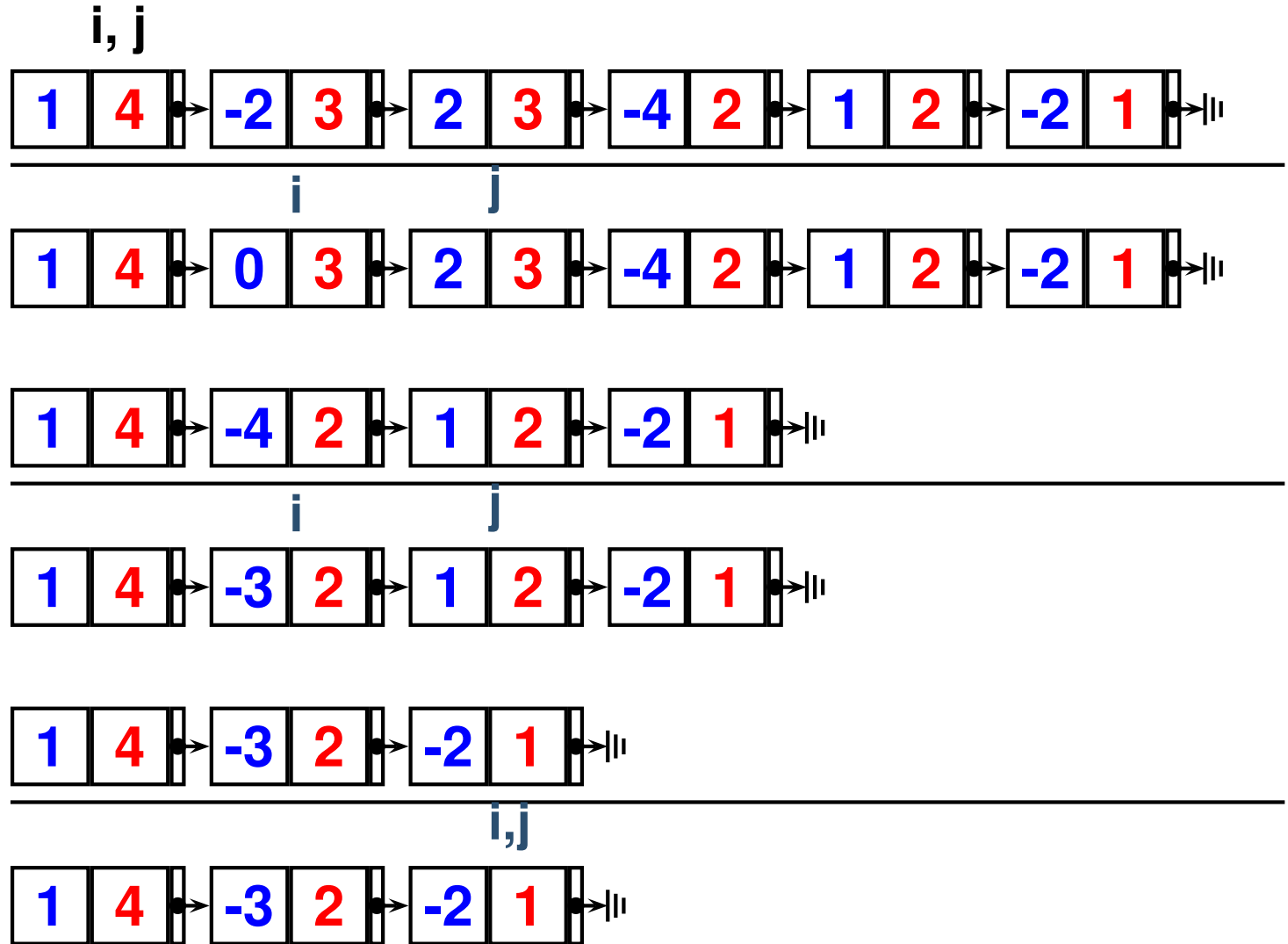
İki Listeyi Birleştirme

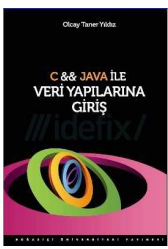
Çift Bağlı Liste Tanımı

Temel Çift Bağlı Liste İşlemleri

Dairesel Bağlı Liste

Uygulama: Polinom Aritmetiği





# İki polinomun çarpımını bulan algoritma (1)

Giriş	1
Bağlı Liste Tanımı	2
Temel Bağlı Liste İşlemleri	3
İki Listeyi Birleştirme	4
Çift Bağlı Liste Tanımı	5
Temel Çift Bağlı Liste İşlemleri	6
Dairesel Bağlı Liste	7
Uygulama: Polinom Aritmetiği	8
	9
	10
	11
	12
	13
	14
	15
	16
	17

```
Liste carp(Liste p1, Liste p2){
    Eleman i, j, yeni, once;
    Liste sonuc;
    int katsayi, us;
    sonuc = new Liste();
    i = p1.bas;
    while (i != null){
        j = p2.bas;
        while (j != null){
            katsayi = i.katsayi * j.katsayi;
            us = i.us + j.us;
            yeni = new Eleman(katsayi, us);
            sonuc.listeyeEkle(yeni);
            j = j.ileri ;
        }
        i = i.ileri ;
    }
}
```



# İki polinomun çarpımını bulan algoritma (2)

<u>Giriş</u>	18
<u>Bağlı Liste Tanımı</u>	19
<u>Temel Bağlı Liste İşlemleri</u>	20
<u>İki Listeyi Birleştirme</u>	21
<u>Çift Bağlı Liste Tanımı</u>	22
<u>Temel Çift Bağlı Liste İşlemleri</u>	23
<u>Dairesel Bağlı Liste</u>	24
<u>Uygulama: Polinom Aritmetiği</u>	25
	26
	27
	28
	29
	30
	31
	32
	33
	34
	35

```
sonuc.sirala ();
i = sonuc.bas;
once = null;
while (i != null){
    j = i.ileri ;
    while (j != null && j.us == i.us){
        i.katsayi += j.katsayi;
        i.ileri = j.ileri ;
        j = j.ileri ;
    }
    if (i.katsayi == 0)
        once.ileri = j;
    else
        once = i;
    i = j;
}
return sonuc;
}
```