

Bölüm 8

Diziler

Dizi Tanımı

Dizi Elemanlarına Değer Atama

Diziler ve Göstergeler

2-Boyutlu Diziler

Değişkenler aynı anda tek bir değer tutabilen *temel değişkenler* ve birden fazla değer saklayabilen *bileşik değişkenler* olmak üzere ikiye ayrılır.

Temel değişkenler bellekte tek bir hücreyi tanımlayıp, içlerinde tek bir değeri tutabilirler.

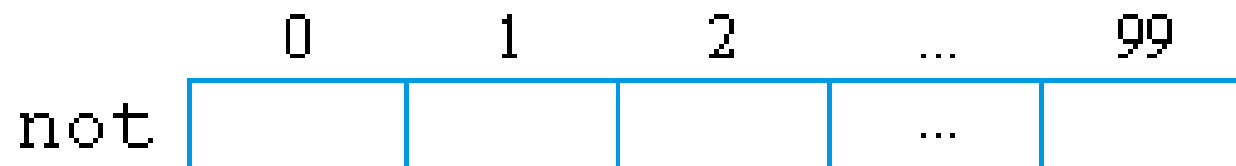
Diziler ise ardarda sıralanmış bellek hücreleridirler. Diziler bu bağlamda *bileşik değişkenlerdir* ve bellekte aynı anda birden fazla değer saklamasını mümkün kılarlar.

veri_tipi dizi_ismi [eleman_sayısı];

`int not[100];`

veri_tipi dizi_ismi eleman_sayısı

Bellek Görüntüsü:



Örnek: `int not[4];`
`int i;`

	0	1	2	...	99
not				...	

`not[0]=20;`

	0	1	2	...	99
not	20			...	

`not[2]= not[0]+10;`

	0	1	2	...	99
not	20		30	...	

`i=0;`

`not[i]=90;`

`not[++i]=70;`

	0	1	2	...	99
not	90	70	30	...	

Dizi Elemanlarına Değer Atama

```
int a[3], b;  
scanf("%d", &b);  
a[2]=b;
```

veya

```
int a[3];  
scanf("%d", &a[2]);
```

Dizi Elemanlarına Değer Atama

Örnek: Kullanıcıdan alınan 5 tamsayı değerini bir dizide saklayan ve bu değerlerin ortalamasını bulan bir program yazalım

```
#include <stdio.h>
int main(void)
{
    int a[5];
    int i, toplam;
    double orta;
    /*Kullanici degerlerinin dizide saklanmasi*/
    printf("Bes tamsayi giriniz:");
    for(i=0; i<5; ++i)
        scanf("%d", &a[i]);
    /*Toplam ve ortalamanın bulunmasi*/
    toplam=0;
    for(i=0; i<5; ++i)
        toplam=toplam+a[i];
    orta= toplam/5.0;
    printf("Ortalama=%5.2f", orta);
    return (0);
}
```

Diziler ve Göstergeler

Bir dizi ismi aslında dizinin ilk elemanını gösteren sabit bir gösterge olarak yaratılır.

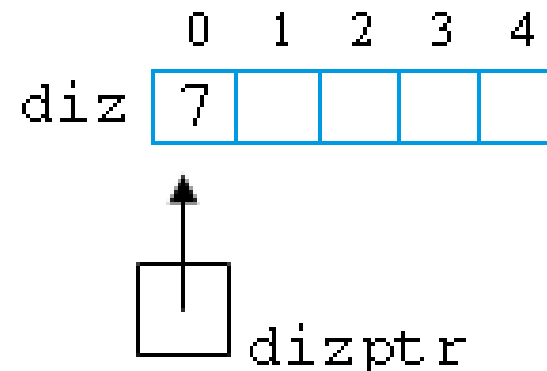
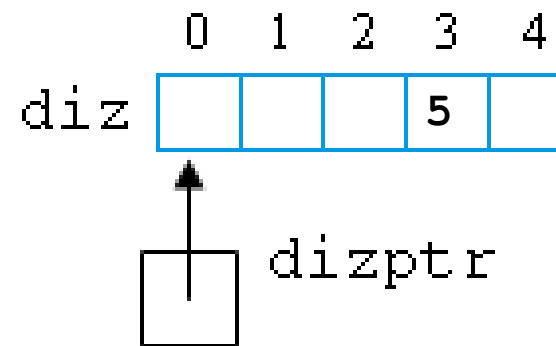
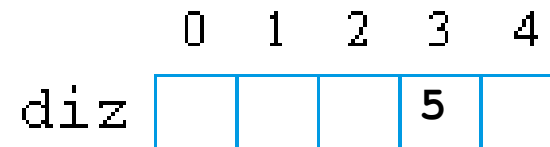
```
int diz[5];
```

```
diz[3] = 5;
```

```
int *dizptr;
```

```
dizptr = &diz[0];
```

```
*dizptr = 7;
```



Gösterge Gösterimi

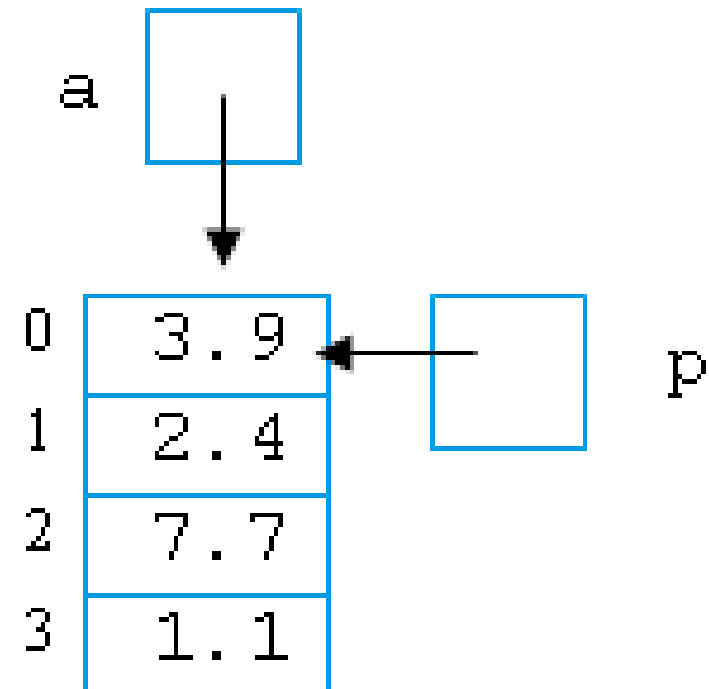
```
*dizptr = 5;  
* (dizptr + 1) = 6;  
* (dizptr + 2) = 7;  
.  
.  
.  
* (dizptr + i) = 8;
```

Dizi Gösterimi

```
diz[0] = 5;  
diz[1] = 6;  
diz[2] = 7;  
.  
.  
.  
diz[i] = 8;
```


Diziler ve Göstergeler

```
1  #include <stdio.h>
2  int main(void)
3  {   double a[]={3.9,2.4,7.7,1.1};
4      double *p;
5      p=a;
6      printf(" %f ", *a);
7      printf(" %f ", a[0]);
8      printf(" %f ", *p);
9      printf(" %f \n", p[0]);
10     printf(" %f ", p[1]);
11     printf(" %f ", a[1]);
12     return (0);
13 }
```



Dizi Elemanının Fonksiyona Gönderilmesi

```
int topla(int x, int y)
{
    return(x+y);
}
```

```
int a[5]={1,2,3,4,5};
printf("%d", topla(a[0],a[4]));
```

Dizinin Fonksiyona Gönderilmesi

```
int a[5]={10,20,30,40,50};
```

a dizisini **f()** fonksiyonuna yollamak için aşağıdaki komutu kullanmalıyız.

```
f(..., a, ...);
```

Bu fonksiyonun başlığı ise iki şekilde olabilir.

```
void f( , int x[5], )
```

```
void f( , int x[], )
```

Fonksiyon parametresinde tanımlanan **x** dizisi yoluyla yapılan değişiklikler, gerçek parametre olan **a** dizisini etkileyecektir. Çünkü dizi ismi dizinin ilk elemanının yerini gösteren bir göstergedir ve kaynak parametresi gibi davranacaktır.

Örnek:

```
#include <stdio.h>
void f1(int b[], int n);
int main(void)
{
    int i, a[]={1,2,3};
    f1(a,3);
    printf("\nmain fonksiyonu ");
    for(i=0;i<3;++i)
        printf("%d", a[i]);
    return(0);
}
void f1(int b[], int n)
{
    int i;
    printf("f1 fonksiyonu ");
    for(i=0;i<n;++i)
    { printf("%d", b[i]);
      b[i]=8;
    }
}
```

Çıktı:

```
f1 fonksiyonu 123
main fonksiyonu 888
```

Dizi Yerine Gösterge Kullanımı

```
float toplama (float apar[])  
{  
    int i;  
    float toplam=0.0;  
    for (i=0;i<4; i++)  
        toplam+=apar[i];  
    return (toplam);  
}
```

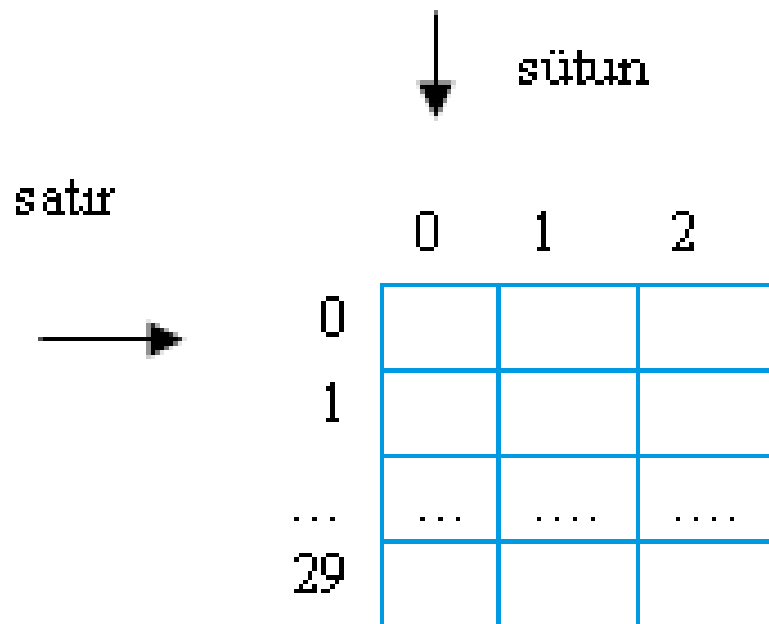
```
float toplama (float *aptr)  
{  
    int i;  
    float toplam=0.0;  
    for (i=0;i<4; aptr++,i++)  
        toplam+=*aptr;  
    return (toplam);  
}
```

2-Boyutlu Diziler

Her elemanı bir boyutlu dizi olan yapılara ise *çok boyutlu diziler* denir. Bu bölümde *matris* veya *tablo* olarak da bilinen 2-boyutlu dizilere değineceğiz.

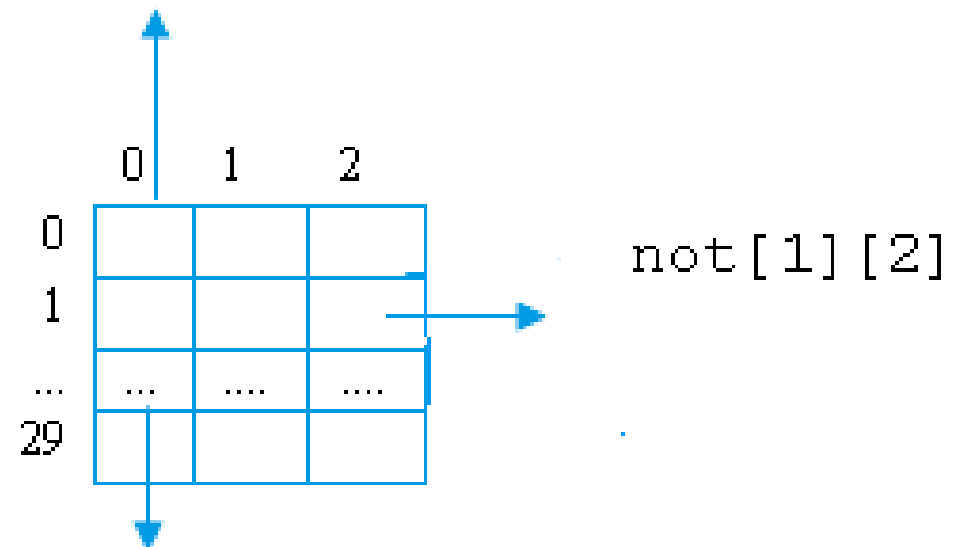
veri_tipi dizi_ismi[satır_büyüklüğü][sütun_büyüklüğü];

int not[3][30];



not

not[0][0]



not[29][0]

2-Boyutlu Dizilere Değer Atama

```
scanf ("%d", &not [29] [0]);
```

```
not [1] [1]=90;
```

Tanımlama Sırasında Değer Atama

```
int b[4][3]={ {15, 30, 39}, {23, 65, 30}, {32, 61, 12}, {48, 34, 11} };  
               1. satır      2. satır      3. satır      4. satır
```

```
int b[][3]={ {15, 30, 39}, {23, 65, 30}, {32, 61, 12}, {48, 34, 11} };  
             1. satır      2. satır      3. satır      4. satır
```

	0	1	2
0	15	30	39
1	23	65	30
2	32	61	12
3	48	34	11

b

satır yönünde atama

Tanımlama Sırasında Değer Atama

```
int b[4][3]= { {15}, {23}, {32}, {48}};
```

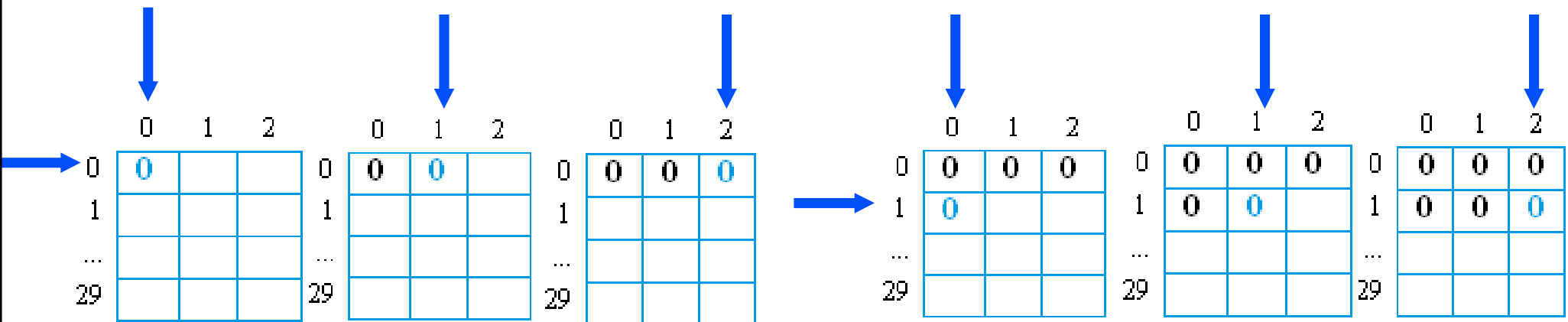
1. satır 2. satır 3. satır 4. satır

	0	1	2
0	15	30	39
1	23	0	0
2	0	0	0
3	0	0	0

b

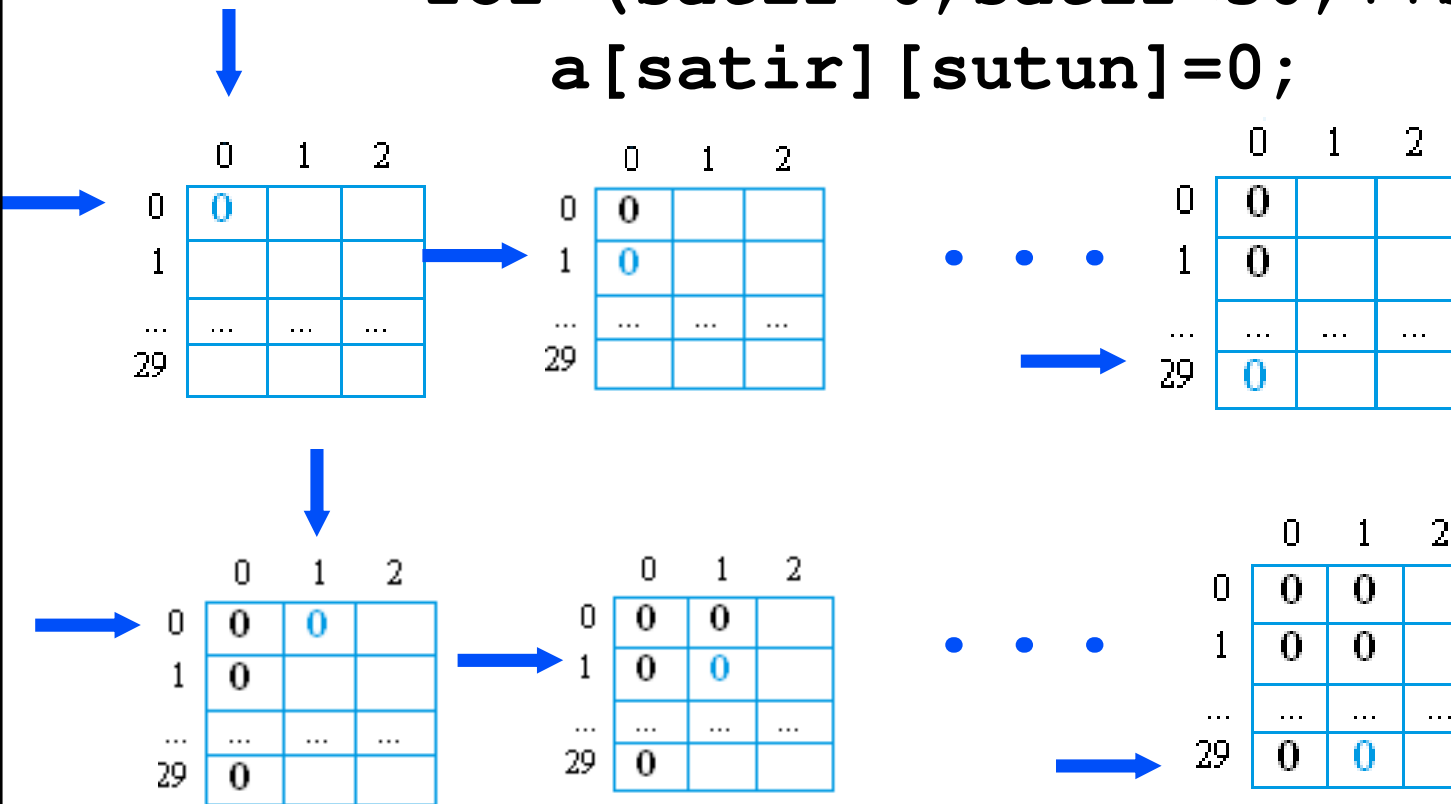
Tanımlama Sonrasında Satır Yönünde Değer Atama

```
int a[30][3];  
for(satir=0; satir<30; ++satir)  
{ for (sutun=0;sutun<3; ++sutun)  
  a[satir][sutun]=0;  
}
```



Tanımlama Sonrasında Sütun Yönünde Değer Atama

```
int a[30][3];  
for(sutun=0; sutun <3; ++sutun)  
    for (satir=0; satir<30; ++satir)  
        a[satir][sutun]=0;
```



2-Boyutlu Diziler

Örnek: Kullanıcının, 30 kişilik bir sınıftaki her öğrenci için 3'er sınav notu gireceği ve her sınav için sınıf ortalamasının ekranda gösterileceği bir program yazınız.

```
#include <stdio.h>
int main(void)
{ int not[30][3];
  int i,j,toplam;
  double orta;
  /* Kullanıcıdan notların alınması*/
  for(i=0;i<30;++i)
  { printf("%d. öğrenci notları:",i+1);
    for(j=0;j<3;++j)
      scanf("%d", &not[i][j]);
  }
  /*Her sınavın ortalamasının bulunması*/
  for(j=0;j<3;++j)
  { toplam=0;
    for(i=0;i<30;++i)
      toplam+=not[i][j];
    orta=toplam/30.0;
    printf("%d. sınav ortalaması: %5.2f\n",j+1,orta);
  }
  return (0);
}
```

2-Boyutlu Diziler ve Fonksiyonlar

Dizi tanımı

```
int a[2][3]={1,2,3,4,5,6};
```

Fonksiyon çağırma

```
f1( , a, );
```

Fonksiyon başlığı

```
f1( , int b[][3], );
```

2-Boyutlu Diziler

Örnek: Öğrencilerin sınav ortalamalarını bulacak bir program yazalım. Bu program için `ortalama()` isimli bir fonksiyon kullanalım. `ortalama()` fonksiyonu çağırıldığı yerden 2-boyutlu bir dizi alır, dizinin her satırının ortalamasını bulup, tek boyutlu bir başka dizide saklar ve ortalamaları tutan bu tek boyutlu diziyi çağırıldığı yere geri döndürür.

```
#include <stdio.h>

void ortalama(int ogr_sa,int sinav_sa,
              int not[][3],double orta[])
{
    double toplam;
    int i,j;
    for(i=0;i<ogr_sa;++i)
    {
        toplam=0;
        for(j=0;j<sinav_sa;++j)
            toplam+=not[i][j];
        orta[i]=toplam/3.0;
    }
}
```

2-Boyutlu Diziler

```
int main(void)
{
    int test[30][3];
    int i,j,toplam;
    double averaj[30];
    /*Kullanıcıdan notların alınması*/
    for(i=0;i<30;++i)
    {
        printf("%d. ogrenci notlari:", i+1);
        for(j=0;j<3;++j)
            scanf("%d",&test[i][j]);
    }
    /*Her ogrencinin ortalamasının bulunması*/
    ortalama(30,3,test,averaj) ;
    for(i=0;i<30;++i)
        printf("%d.ogrenci ortalamasi:%5.2f\n",i+1,
            averaj[i]);
    return (0);
}
```