

BİLGİSAYAR BİLİMLERİNE GİRİŞ I

-9-

Dizi Nedir?

2

- Veri yapısı türlerinden bir tanesidir (Array, Struct, Pointer, Class...).
- İçerisinde birden fazla,
 - ▣ **aynı tip** ve
 - ▣ **aynı isimdeki**veriyi **bellekte** depolayabilen değişkenlere **dizi** denir.

Örneğin: Bir sınıftaki öğrencilerin notlarını saklamak için dizileri kullanmak mümkündür.

Dizilerin Bildirimi

3

- Bir dizi çok sayıda değişken barındırdığından, bunları bir birinden ayırt etmek için **indis** adı verilen bilgiler kullanılır.
- C Programlama Dili'nde, bir dizi hangi tipte tanımlanmış olursa olsun **başlangıç indisi** her zaman **0'dır**.
- Diziler tanımlanırken,
 - ▣ dizinin **adı**,
 - ▣ dizinin **boyutu**,
 - ▣ dizi elemanların hangi **tipte** olacağı belirtilmelidir.

Dizilerin Bildirimi

4

- Bir dizinin bildirim işleminin genel biçimi aşağıdaki gibidir:

veriTipi dizi_adı[eleman_sayısı];

- Örneğin, double türündeki 8 adet öğrenci notunu bellekte tutmak için aşağıdaki gibi bir dizi tanımlayabiliriz.

double ogrenci_notu[8];

Örnek: Dizilerin Bildirimi

5

`ogrenci_notu`
dizinin ismidir.

Dizinin 8 elemanı

<code>ogrenci_notu[0]</code>	45	1. eleman
<code>ogrenci_notu[1]</code>	56	2. eleman
<code>ogrenci_notu[2]</code>	78	3. eleman
<code>ogrenci_notu[3]</code>	93	4. eleman
<code>ogrenci_notu[4]</code>	78	5. eleman
<code>ogrenci_notu[5]</code>	69	6. eleman
<code>ogrenci_notu[6]</code>	77	7. eleman
<code>ogrenci_notu[7]</code>	90	8. eleman

Örnek: Dizi elemanlarına değer atama ve okuma

6

- Tek boyutlu, **5 elemanlı**, sayısal (int) bir dizi tanımlanarak:
 - Dizi elemanlarına dizi indislerinin 3 katı for döngüsü içerisinde değer olarak atanacaktır.
 - Yine başka bir for döngüsü içerisinde bu değerler ekrana yazdırılacaktır.

```
#include<stdio.h>
#include<conio.h>
```

```
main()
{
int dizi[5],i;
```

```
for(i=0;i<5;i++){
    dizi[i]=i*3;
}
```

```
for(i=0;i<5;i++){
    printf("%d. dizi eleman degeri: %d\n",i,dizi[i]);
}
getch();
}
```

0. dizi eleman degeri: 0
1. dizi eleman degeri: 3
2. dizi eleman degeri: 6
3. dizi eleman degeri: 9
4. dizi eleman degeri: 12

Örnek: Dizi atamaları

8

- ✓ `int x[5];` `// 5 elemanlı sayısal x dizisi`
- ✓ `x[0] = 20;` `// Geçerli atama`
- ✓ `x[2.3]=5;` `// Geçersiz atama`

Örnek: Dizi elemanı değeri kadar ekrana * karakteri yazdırma *

9

- 10 elemanlı {19, 3, 15, 7, 11, 9, 13, 5, 17, 1} grafik isimli bir dizi oluşturulacaktır.
- Dizideki elemanlar tek tek okunarak her dizi elemanının sayısı kadar ekrana * karakteri yazdırılacaktır.
- Ekran çıktısı aşağıdaki gibi olacaktır.

Eleman	Deger	Histogram
0	19	*****
1	3	***
2	15	*****
3	7	*****
4	11	*****
5	9	*****
6	13	*****
7	5	*****
8	17	*****
9	1	*

```
#include <stdio.h>
#include <conio.h>
#define BOYUT 10
```

```
main()
{
    int n[BOYUT] = { 19, 3, 15, 7, 11, 9, 13, 5, 17, 1 };
    int i, j;
    printf( "%s%13s%17s\n", "Eleman", "Deger", "Histogram" );
    for ( i = 0; i <= BOYUT-1; i++ ) {
        printf( "%7d%13d      ", i, n[ i ] );
        for ( j = 1; j <= n[ i ]; j++ )
            printf( "%c", '*' );
        printf( "\n" );
    }
    getch();
}
```

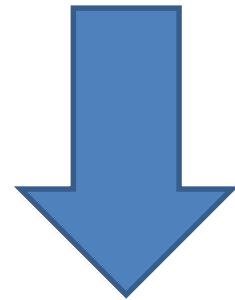
Eleman	Deger	Histogram
0	19	*****
1	3	***
2	15	*****
3	7	*****
4	11	*****
5	9	*****
6	13	*****
7	5	*****
8	17	*****
9	1	*

Örnek: Dizinin minimum elemanını bulma

Örnek: Dizinin minimum elemanını bulma

12

```
#include <stdio.h>
#include <conio.h>
main()
{
    int dizi[6];
    int enk,i;
    printf("Dizinin elemanlarini giriniz\n");
    for(i=0;i<6;i++){
        printf("Dizinin %d. elemaninini giriniz :\n",i);
        scanf("%d",&dizi[i]);
    }
```



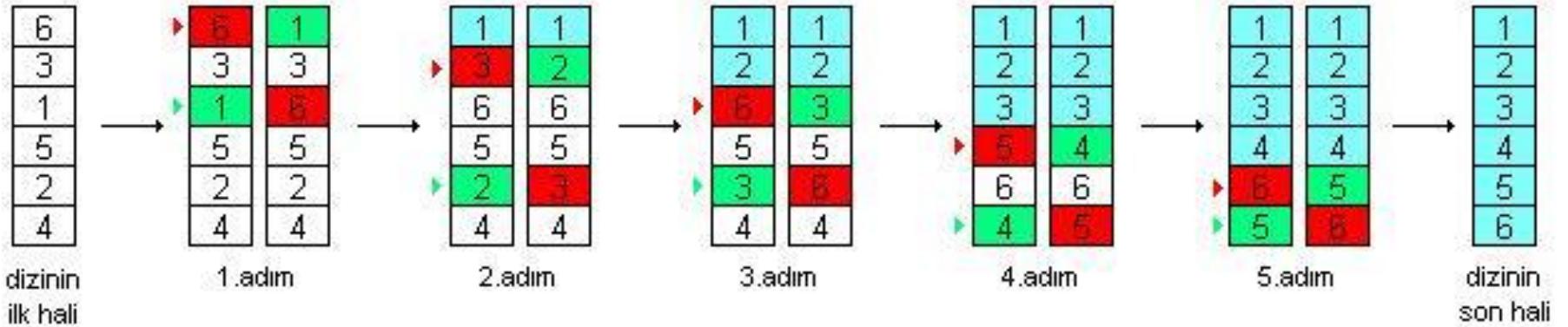
```
enk=dizi[0];  
    for(i=1;i<6;i++)  
    {  
        if(enk>dizi[i])  
        {  
            enk=dizi[i];  
        }  
    }  
    printf("minimun eleman %d dir",enk);  
    getch();  
}
```

Dizileri Sıralamak

14

- Veri sıralamak
 - ▣ Önemli bilgisayar uygulamalarından biridir.
 - ▣ Hemen hemen tüm organizasyonlar bazı verileri sıralamak zorundadır.
- Seçmeli sıralama
 1. Listedeki en küçük değerli öğeyi bul.
 2. İlk konumdaki öğeyle bulunan en küçük değerli öğenin yerini değiştir.
 3. Yukarıdaki adımları listenin ilk elemanından sonrası için (ikinci elemandan başlayarak) yinele.

Seçmeli Sıralama



	Sırayla kontrol edilen eleman
	Yerine yerleşmiş sabit eleman
	1.adımda dizideki en küçük eleman. Diğer adımlarda sabit elemandan sonra gelen en küçük eleman.

```
#include <stdio.h>
#include <conio.h>
#define BOYUT 10
int main()
{
    int dizi [BOYUT]={2,6,4,8,10,12,89,68,45,37},i;
    int enkucuk, yedek;
    for (int i = 0; i < BOYUT-1; i++)
    {
        enkucuk = i;
        for (int j = i + 1; j < BOYUT; j++)
            if (dizi[j] < dizi[enkucuk])
                enkucuk = j;
        yedek = dizi[i];
        dizi[i] = dizi[enkucuk];
        dizi[enkucuk] = yedek;
    }

    for(i=0;i<BOYUT;i++)
        printf("%d ",dizi[i]);
    getch();
    return 0;
}
```


Örnek: 10 elemanlı dizinin elemanlarını sıralama

Örnek: 10 elemanlı dizinin elemanlarını sıralama

18

```
#include<stdio.h>
```

```
int main()
{
    int a[10];
    int k,j,temp,i;
    for(i=0;i<10;i++)
    {
        printf("\nDizinin %d. elemanini
        giriniz:",i);
        scanf("%d",&a[i]);
    }
```

```
for(j=0;j<10-1;j++)
{
    for(k=j+1;k<10;k++)
    {
        if( a[j] > a[k] )
        {
            temp=a[j];
            a[j]=a[k];
            a[k]=temp;
        }
    }
    for(k=0;k<10;k++)
        printf("%d ",a[k]);
}
```

Doğrusal Arama Yapmak

19

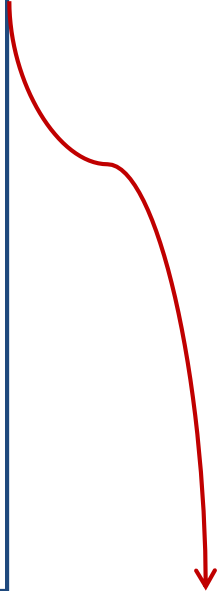
- Bir dizide belirli bir arama değerine eşit bir değer olup olmadığının kararı
- **Lineer arama**
 - ▣ Dizinin her elemanını arama değeriyle karşılaştırır.
 - ▣ Küçük ya da sıralanmamış dizilerde iyi çalışır.

Lineer Arama

20

```
#include <stdio.h>
#include <conio.h>
main()
{
    int a[10], ara, i, n=0;

    for (i = 0; i < 10; i++)
    {
        scanf("%d", &a[i]);
    }
    printf("Dizide aranacak olan sayiyi giriniz:\n");
    scanf("%d", &ara);
```



```
for (i = 0; i < 10; i++)  
{  
    if (a[i] == ara)    /* aranan eleman bulunduysa */  
    {  
        printf("%d sayisi %d. sirada bulunmaktadir\n", ara, i+1);  
        n=1;  
        break;  
    }  
}
```

```
if (n == 0)  
    printf("%d sayisi dizide bulunmamaktadir.\n", ara);  
else printf("Sayi bulundu");
```

```
    getch();  
}
```

Çok Boyutlu Diziler

22

- Satırlar ve sütunlar biçiminde düzenlenmiş tablolar.
- İki boyutlu diziler *matris* olarak adlandırılır.
 - ▣ ilk boyuta *satır*, ikinci boyuta *sütun* denir.
- Matrislerin bildirimleri için genel yazım biçimi,

tip **ad**[**satır_sayisi**][**sutun_sayisi**]

- Örneğin, elemanları tamsayı olan 3 satır, 5 sütunlu bir matris aşağıdaki gibi bildirilir.

int **b**[**3**][**5**]

Matrisler

23

	0 th column	1 st column	2 nd column	3 rd column	4 th column
0 th row	b [0] [0]	b [0] [1]	b [0] [2]	b [0] [3]	b [0] [4]
1 st row	b [1] [0]	b [1] [1]	b [1] [2]	b [1] [3]	b [1] [4]
2 nd row	b [2] [0]	b [2] [1]	b [2] [2]	b [2] [3]	b [2] [4]

Örnek: Çok Boyutlu Dizi Bildirimi ve Dizi Elemanlarını Yazdırma

24

```
#include<stdio.h>
#include<conio.h>
main()
{
    int i,j;
    int x[3][4]={ 16,35,48,63,23,12,26,34,56,52,34,33 };

    for(i=0;i<3;i++)
    {
        for(j=0;j<4;j++)
        {
            printf("%4d",x[i][j]);
        }
        printf("\n");
    }
    getch();
}
```


İki Matrisin Toplamı

25

Doğrusal cebirde veya daha genel ifade ile matematikte **matris toplamı**, iki matrisin ilgili girişlerinin eklenmesi işlemidir.

İki matrisin toplanabilmesi için boyutlarının aynı olması gerekir. Farklı boyutlu iki matris toplanamaz.

$$\left[a_{ij} + b_{ij} \right] = \left[c_{ij} \right]$$

```
#include<stdio.h>
```

```
#include<conio.h>
```

```
main()
```

```
{
```

```
    int i,j;
```

```
    int a[2][3]={3,5,8,0,1,2};
```

```
    int b[2][3]={6,2,9,1,1,4};
```

```
        printf("\n A MATRISI\n");
```

```
        for(i=0;i<2;i++)
```

```
        {
```

```
for(j=0;j<3;j++)
```

```
        {
```

```
        printf("%4d",a[i][j]);
```

```
        }
```

```
        printf("\n");
```

```
    }
```

A MATRISI

3 5 8

0 1 2

B MATRISI

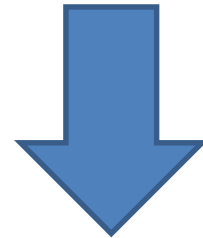
6 2 9

1 1 4

C MATRISI

9 7 17

1 2 6



```
printf("\n B MATRISI\n");
    for(i=0;i<2;i++)
    {
for(j=0;j<3;j++)
    printf("%4d",b[i][j]);
    printf("\n");
    }
printf("\n C MATRISI\n");
    for(i=0;i<2;i++)
    {
        for(j=0;j<3;j++)
        printf("%4d",a[i][j]+b[i][j]);
        printf("\n");
    }
getch();
}
```

Matris Çarpımı

28

A' nın 1. Satır elemanları B' nin 1. sütun elemanları ile çarpılıp toplanır. Bu AB çarpım matrisinin birinci elemanıdır (C[0][1]).

A' nın 1. Satır elemanları B' nin 2. sütun elemanları ile karşılıklı çarpılıp toplanarak çarpım matrisinden (C[0][2]) elemanı elde edilir.

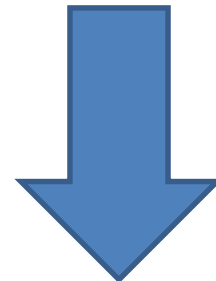
Bu çarpım A matrisinin bütün satırları B matrisinin bütün sütunları ile çarpılıp yeni matris elde edilinceye kadar devam eder.

$$\sum_{k=1}^n a_{ik} b_{kj} = a_{i1} b_{1j} + a_{i2} b_{2j} + \dots + a_{in} b_{nj} = [c_{ij}]$$

```
#include<stdio.h>
#include<conio.h>
```

```
main()
{
    int i,j,k,toplam;
    int a[3][3]={3,5,8,0,1,2,4,4,4};
    int b[3][3]={6,2,9,1,1,4,2,5,4};
    int c[3][3];

    printf("\n C MATRISI\n");
```



```
for(i=0;i<3;i++)
```

```
{
```

```
    for(j=0;j<3;j++)
```

```
    {
```

```
        toplam=0;
```

```
        for(k=0;k<3;k++)
```

```
        {
```

```
            toplam+=a[i][k]*b[k][j];
```

```
        }
```

```
        c[i][j]=toplam;
```

```
        printf("%4d",c[i][j]);
```

```
    }
```

```
    printf("\n");
```

```
}
```

```
getch();
```

```
}
```

C MATRISI

39 51 79

5 11 12

36 32 68

Sorular???

