



BİLGİSAYAR PROGRAMLAMA 1

Ders Notu 7 – Diziler ve Matrisler

Konya Teknik Üniversitesi
Elektrik – Elektronik Mühendisliği Bölümü

18.04.2024

Konya

Diziler (Arrays)

- Bugüne kadar anlatılan konularda verileri depolamak için çeşitli değişkenler kullanıldı.
- Bazı programlarda aynı tip ve çok fazla sayıda değişkeni depolamak gerekebilir.

Örnek:

Bir elektrik sinyalinin voltaj değerini 5'er saniye aralıklarla 500 saniye boyunca kaydetmek.

- Böyle bir program yazmak için 100 adet double tipinde değişken tanımlamamız gerekirdi.
- Diziler bu noktada işimizi kolaylaştırmaktadır.

$$a = \{a_0, a_1, a_2, \dots, a_{99}\}$$

Tanım ve Özellikler

Dizinin tanımı:

- Bir dizi belirli sayıda ve benzer tipteki değişkenlere tek bir isim ve çeşitli indisler yardımıyla erişim sağlayan özel bir veri yapısıdır.



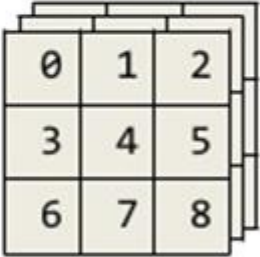
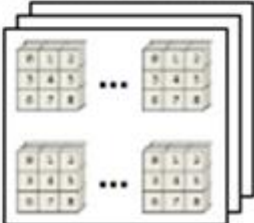
Özellikleri:

- Bir dizi aynı tip verilere sahiptir.
- Bellekte art arda saklanırlar.

```
int a[ ] = {1123, 1125, 1234, 1256, 1321};
```

Dizilerin Özellikleri

- Tek boyutlu ve çok boyutlu olabilirler.

| Boyutlar | Örnek | Adı |
|----------|---|----------------|
| 1 |  | Vektör |
| 2 |  | Matris |
| 3 |  | 3 Boyutlu Dizi |
| N |  | N Boyutlu Dizi |

Dizilerin Özellikleri

- Bir dizi çok sayıda değişken barındırdığından, bunları birbirinden ayırt etmek için indis adı verilen bilgiler kullanılır.
- Bilgisayar programları basit editörlerle yazılır. O editörler a_i simgesindeki gibi alt simge koyamaz. Üstelik, derleyiciler de alt simgeyi anlamaz. O nedenle, alt simge işlevini görmek üzere, indisler köşeli parantez içine yazılır.
- Dolayısıyla, bir dizinin öğeleri aşağıdaki gibi yazılır:

$$a = \{a[0], a[1], a[2], \dots, a[99]\}$$

Dizilerin Özellikleri

- [] operatörünün iki işlevi vardır:

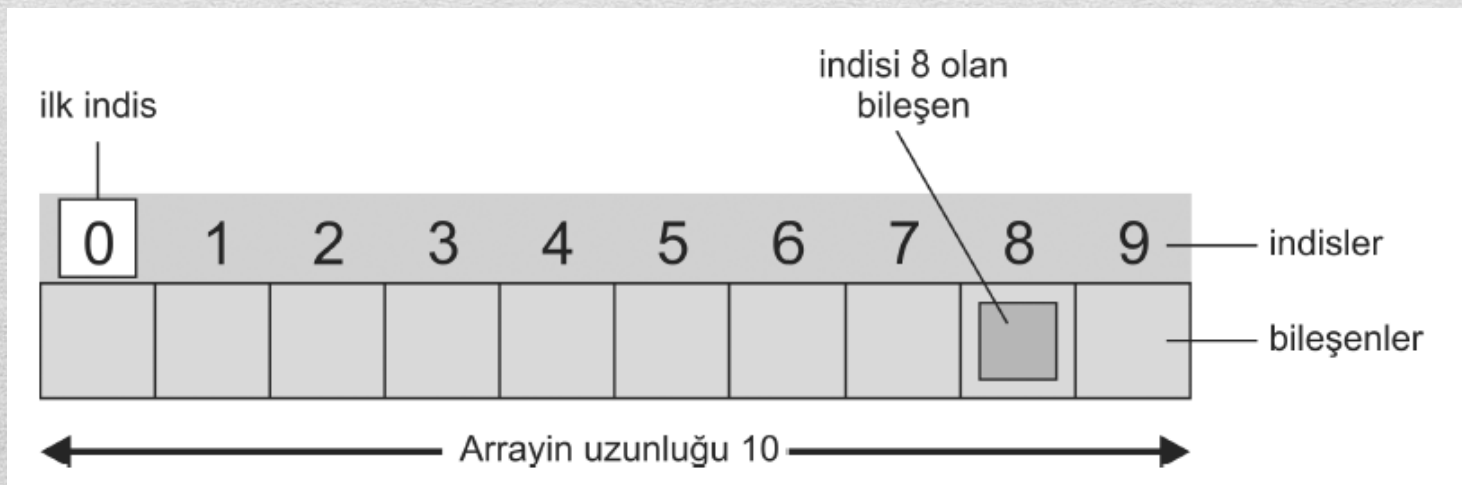
1) array bildirimi için kullanılır.

```
int a[n];
```

2) arrayin bileşenlerinin sıra numarasını (indis) gösterir.

- Arrayde indisler daima tam sayıdır ve 0'dan başlar.

n bileşeni olan arrayin indisleri ($i = 0, 1, 2, \dots, n - 1$) olur.



Dizilerin Özellikleri

- Diziler tanımlanırken;
Dizinin adı, dizinin boyutu, dizi elemanlarının hangi tipte olacağı belirtilir.

veritipi diziAdi[elemanSayisi];

```
int a[n];
```

- Örnek:
double türündeki 8 adet öğrenci notunu bellekte tutmak için aşağıdaki gibi bir dizi tanımlayabiliriz:
double ogrenci_notu[8];

ogrenci_notu[0]

45

1. eleman

ogrenci_notu[1]

56

2. eleman

ogrenci_notu[2]

78

3. eleman

ogrenci_notu[3]

93

4. eleman

ogrenci_notu[4]

78

5. eleman

ogrenci_notu[5]

69

6. eleman

ogrenci_notu[6]

77

7. eleman

ogrenci_notu[7]

90

8. eleman

Dizilerin Özellikleri

- Örnek:

double türündeki 8 adet öğrenci notunu bellekte tutmak:
Dizileri kullanmasaydık;

```
double ogrenci_notu1;  
double ogrenci_notu2;  
double ogrenci_notu3;  
double ogrenci_notu4;  
double ogrenci_notu5;  
double ogrenci_notu6;  
double ogrenci_notu7;  
double ogrenci_notu8;
```


Array Bileşenlerinin Kullanımı:

- İki dizi elemanının toplanıp ekrana basılması:

```
printf( " Sonuc:%f ", ogrenci_notu[0] + ogrenci_notu[1] );
```

- Dizinin yedinci elemanının değerinin 2'ye bölünüp x değişkenine atanması: ?

```
x=ogrenci_notu[6]/2;
```



- Dizinin yedinci elemanı: `ogrenci_notu[6]`
- Dizinin yedi numaralı elemanı: `ogrenci_notu[7]`

Dizilerde Atama ve Döngü Kurma

- Bir dizinin uzunluğu belirtilmeden de başlangıç değeri atamak mümkündür.

Örnek:

- `int a[]={ 100, 200, 300, 400};`
- `float v[]={9.8, 11.0, 7.5, 0.0, 12.5};`
- Derleyici bu şekilde bir atama ile karşılaştığında, küme parantezi içindeki eleman sayısını hesaplar ve dizinin o uzunlukta açıldığını varsayar.
- Yukarıdaki örnekte, a dizisinin 4, v dizisinin 5 elemanlı olduğu varsayılır
- `int sayilar[20]={ };`
 - Tüm dizi elemanlarına 0 değeri atanır...

Diziler (Arrays)

- Örnek:

```
1 #include<stdio.h>
2
3 int main()
4 {
5     int i=0;
6     int sayilar[20];
7     for (i=0; i<20; i++ )
8     {
9         printf("Dizinin %d. elemani: %d'dir. \n", i+1, sayilar [i]);
10    }
11
12
13 }
```

```
Dizinin 1. elemani: 4202608'dir.
Dizinin 2. elemani: 0'dir.
Dizinin 3. elemani: 164'dir.
Dizinin 4. elemani: 0'dir.
Dizinin 5. elemani: 4616768'dir.
Dizinin 6. elemani: 0'dir.
Dizinin 7. elemani: 1'dir.
Dizinin 8. elemani: 0'dir.
Dizinin 9. elemani: -1'dir.
Dizinin 10. elemani: -1'dir.
Dizinin 11. elemani: 164'dir.
Dizinin 12. elemani: 0'dir.
Dizinin 13. elemani: 1'dir.
Dizinin 14. elemani: 0'dir.
Dizinin 15. elemani: 4203641'dir.
Dizinin 16. elemani: 0'dir.
Dizinin 17. elemani: 3'dir.
Dizinin 18. elemani: 0'dir.
Dizinin 19. elemani: 164'dir.
Dizinin 20. elemani: 0'dir.
```

```
1 #include<stdio.h>
2
3 int main()
4 {
5     int i=0;
6     int sayilar[20]={};
7     for (i=0; i<20; i++ )
8     {
9         printf("Dizinin %d. elemani: %d'dir. \n", i+1, sayilar [i]);
10    }
11
12
13 }
```

```
Dizinin 1. elemani: 0'dir.
Dizinin 2. elemani: 0'dir.
Dizinin 3. elemani: 0'dir.
Dizinin 4. elemani: 0'dir.
Dizinin 5. elemani: 0'dir.
Dizinin 6. elemani: 0'dir.
Dizinin 7. elemani: 0'dir.
Dizinin 8. elemani: 0'dir.
Dizinin 9. elemani: 0'dir.
Dizinin 10. elemani: 0'dir.
Dizinin 11. elemani: 0'dir.
Dizinin 12. elemani: 0'dir.
Dizinin 13. elemani: 0'dir.
Dizinin 14. elemani: 0'dir.
Dizinin 15. elemani: 0'dir.
Dizinin 16. elemani: 0'dir.
Dizinin 17. elemani: 0'dir.
Dizinin 18. elemani: 0'dir.
Dizinin 19. elemani: 0'dir.
Dizinin 20. elemani: 0'dir.
```

Process exited after 0.01189 s

Dizilerde Atama ve Döngü Kurma

- Dizilere başlangıç değeri atarken, tüm elemanlara değer vermeden de atama yapmak mümkündür.

Örnek:

- `int sayilar[20]={ 1, 2, 3 };`
 - Dizinin ilk 3 elemanına 1, 2 ve 3 değerleri atanır. 4'ten itibaren olan dizi elemanlarına 0 değeri atanır...
- Sayısal tipteki dizi elemanlarına 0 değeri, metin tipindeki dizi elemanlarına NULL değeri atanır.

Dizilerde Atama ve Döngü Kurma

- `#include<stdio.h>`
- `#include<stdlib.h>` //Dizideki her bir elemanın değerinin 3 ile çarpılması
- `int main ()`
- `{`
- `int ornek_dizi [5]={ 1,2,3,4,5};` //Diziyi oluşturma
- `int i;`
- `for (i=0; i<5; i++)` //Dizi elemanlarına değer atama
- `{`
- `ornek_dizi [i]=ornek_dizi [i]*3;`
- `}`
- `for (i=0; i<5; i++)`
- `{` //Dizi elemanlarını yazdırma
- `printf("Dizinin %d. elemanı: %d'dir. \n", i+1, ornek_dizi [i]);`
- `}`
- `return(0);`
- `}`

```
Dizinin 1. elemanı: 3'dir.  
Dizinin 2. elemanı: 6'dir.  
Dizinin 3. elemanı: 9'dir.  
Dizinin 4. elemanı: 12'dir.  
Dizinin 5. elemanı: 15'dir.
```

Dizilerde Atama Örnekleri

`int x[5]={ };` //5 elemanlı sayısal x dizisi oluşturma

`int i=2;`

`x[0]=20;` //Geçerli atama

`x[2.3]=5;` //Geçersiz atama (Yanlış)

`x[2*i-3]=3;` //Geçerli atama, `x[1]` dizi elemanına 3 değerini atar

`x[i++];` //Önce `x[2]` dizi elemanına erişilir daha sonra `i`
//değişkeninin değeri 1 arttırılır.

`x[(int)x[1]];` //x[3] dizi elemanına erişilir

Dizi Elemanlarını Kullanıcıdan Okumak

- Klavyeden maksimum 10 tane sayısal değer girilecektir.
- Girilen sayılar bir dizide saklanacaktır.
- Sayı girme işlemi 0 girilene kadar devam edecektir.
- 0 değeri girildiğinde 0 sayısını hariç girilen diğer tüm değerler diziden okunarak ekrana yazdırılacaktır.

```
1  #include<stdio.h>
2  #include<stdlib.h>
3  int main ()           //Kullanıcıdan sayısal değer okuma
4  {
5      int ornek_dizi [10]; //Diziyi oluşturma
6      int i, j;
7      for (i=0; i<10; i++ ) //Dizi elemanlarına kullanıcıdan değer alma
8      {
9          printf ("%d. sayiyi giriniz", i+1);
10         scanf("%d", &ornek_dizi[i]);
11         if(ornek_dizi[i]==0)
12             break;
13         j=i;
14     }
15     for (i=0; i<=j; i++ ) //Dizi elemanlarını yazdırma
16     {
17         printf("%d \n", ornek_dizi[i]);
18     }
19     return(0);
```

C:\Users\Kema\\Desktop\G

```
1. sayiyi giriniz4
2. sayiyi giriniz6
3. sayiyi giriniz8
4. sayiyi giriniz2
5. sayiyi giriniz3
6. sayiyi giriniz4
7. sayiyi giriniz0
```

```
4
6
8
2
3
4
```

Process exited after
Press any key to con

Dizilerle ilgili Önemli Hatırlatmalar

- Dizi boyunca döngü kullanırken dizi indisi asla 0'ın altına inmemeli ve her zaman dizideki toplam eleman sayısından az olmalıdır (büyüklük-1).
- Döngü devam şartının bu aralığın dışındaki elemanlara ulaşılmasını engellediğinden emin olmamız gereklidir.
- Dizi sınırlarının dışındaki elemanları kullanmanın yaratacağı hatalar (genelde ciddi hatalardır) sistemden sisteme farklılık gösterir.

Dizilerin Bazı Dezavantajları

- **Dizinin boyu değiştirilemez:**

Dizinin boyunu (boyutunu, bileşenlerinin sayısını) ya bildiriminde belirtiriz ya da bileşenlerine ilk değerlerini atayarak belirlemiş oluruz. Her durumda, dizinin boyu başlangıçta kesin belirlenmiş olur.

- **Dizinin bileşen değerleri aynı veri tipindendir:**

Bazı durumlarda farklı veri tiplerinden oluşan veri koleksiyonlarını işlemek gerekebilir. O zaman, bileşenleri aynı veri tipinden olan dizi yapısı işimize yaramaz. Java, C#, Python, Ruby gibi yeni nesil diller farklı veri tiplerinden oluşan koleksiyonlarla iş yapmayı sağlayan yapılar getirmişlerdir.

Çok Boyutlu Diziler

- Bir dizi birden çok boyuta sahip olabilir.

Örneğin iki boyutlu y dizisi şöyle tanımlanabilir:

```
int y[5][10];
```

- İki boyutlu diziler **matris** olarak adlandırılır.
- İlk boyuta **satır**, ikinci boyuta **sütun** denir. y matrisinin eleman sayısı $5 \times 10 = 50$ 'dir.

Çok Boyutlu Diziler

| Dizi Çeşidi | Genel Bildirimi | Örnek |
|---------------------------------|---|-------------------------------------|
| Tek boyutlu diziler (Vektörler) | tip dizi_adı[eleman_sayısı] | int veri[10]; |
| İki boyutlu diziler (Matrisler) | tip dizi_adı[satır_sayısı][sutun_sayısı] | float mat [5] [4] ; |
| Çok boyutlu diziler | tip dizi_adı[boyut_1][boyut_2]...[boyut_n] | double x [2] [4] [2] ; |

Çok Boyutlu Diziler

- Çok boyutlu diziler tek boyuta indirgenerek bellekte tutulurlar.
- Tek indisli dizilerde olduğu gibi, çok indisli dizilere de başlangıç değeri vermek mümkündür.

Örneğin 3 satır ve 4 sütunlu ($3 \times 4 = 12$ elemanlı) bir x matrisinin elemanları şöyle tanımlanabilir:

- `int x[3][4]={ 11, 34, 42, 60, 72, 99, 10, 50, 80, 66, 21, 38};`
- | | |
|--|------------------------------------|
| <code>int x[3][4]={ 11, 34, 42, 60,</code> | <code>/*1.satirelemanlari*/</code> |
| <code>72,99,10,50,</code> | <code>/*2.satirelemanlari*/</code> |
| <code>80,66,21,38};</code> | <code>/*3.satirelemanlari*/</code> |

Çok Boyutlu Diziler

ÖRNEK 1

```
1  #include<stdio.h>
2  #include<stdlib.h>
3  int main ()                                //İki Boyutlu Dizi Oluşturma ve Yazdırma
4  {
5      int i, j;
6      int x[3][4]={11,34,42,60,              //Diziyi oluşturma
7                  72,99,10,50,
8                  80,66,21,38};
9      for (i=0; i<3; i++ )                  //Diziyi yazdırma
10     {
11         for (j=0; j<4; j++ )
12         {
13             printf ("%d ", x[i][j]);
14         }
15         printf ("\n");
16     }
17     return(0);
18 }
19
```

```
11 34 42 60
72 99 10 50
80 66 21 38
```

```
-----
Process exited
Press any key to
```

- `int x[3][4]={11, 34, 42, 60, 72, 99, 10, 50, 80, 66, 21, 38};`
 - `int y[3][4]={};` \Rightarrow `0000`
`0000`
`0000`
 - `y[0][0]=y[0][0]+x[0][0];`
 - `y[0][1]=x[0][2]*2;`
 - `y[0][3]=5;`
 - `for (i=0; i<3; i++) //y dizisini yazdırma`
`{`
`for (j=0; j<4; j++)`
`{`
`printf ("%d ", y[i][j]);`
`}`
`printf ("\n");`
`}`
-

```
Process exited after
Press any key to cont
```


Matris İşlemleri

Matris Toplama

$$\begin{bmatrix} 1 & 3 \\ 1 & 2 \end{bmatrix} + \begin{bmatrix} 7 & 5 \\ 2 & 1 \end{bmatrix} = \begin{bmatrix} 8 & 8 \\ 3 & 3 \end{bmatrix}$$

```
for(i=0; i<2; i++){  
    for(j=0; j<2; j++){  
        C[i][j] = A[i][j] + B[i][j];  
        printf("%d ", C[i][j]);  
    }  
    printf("\n"); }
```

Matris Çarpma

$$\begin{bmatrix} 1 & 2 \\ 3 & 1 \end{bmatrix} \cdot \begin{bmatrix} 4 & 3 \\ 1 & 2 \end{bmatrix} = \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix}$$

$$C_{11} = 1.4 + 2.1 = 6$$

$$C_{12} = 1.3 + 2.2 = 7$$

```
for(i=0; i<2; i++){  
    for(j=0; j<2; j++){  
        for(k=0; k<2; k++){  
            C[i][j] += A[i][k] * B[k][j];  
        }  
        printf("%d ", C[i][j]);  
        printf("\n");  
    }  
}
```


ÖDEV:

Bir dersi alan 10 öğrencinin vize ve final notlarını dizilerde saklayabilecek bir program tasarlayın. Notların girişini kullanıcıdan isteyerek dizilere kaydedin.

Bu notlara göre vize notu %40, final notu %60 olacak şekilde her bir öğrencinin yıl sonu puanını, harf karşılığını ve sınıf ortalamasını ekrana yazdırın.

KAYNAKLAR:

- Programlama Sanatı, Algoritmalar, C Dili Uyarlaması, Dr. Rifat ÇÖLKESEN, Papatya Yayıncılık
- Her Yönüyle C, Tevfik KIZILÖREN, Kodlab
- C Programlama Dili, Dr. Rifat ÇÖLKESEN, Papatya Yayıncılık
- Celal Bayar Üniversitesi, Hasan Ferdi Turgutlu Teknoloji Fakültesi, YZM1105 Ders Notu