

CENG 111 Algoritmalar ve Programlama 7: Diziler

Öğr.Gör. Şevket Umut ÇAKIR

Pamukkale Üniversitesi

Hafta 7

Anahat

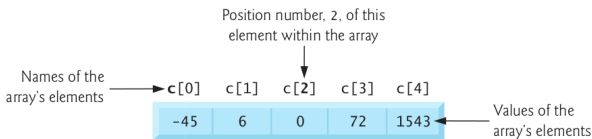
- 1 Giriş
- 2 Diziler
- 3 Dizileri Tanımlamak
- 4 Dizi Örnekleri
- 5 Metinleri Depolamak ve Değiştirmek İçin Karakter Dizilerini Kullanmak
- 6 Statik ve Otomatik Yerel Diziler
- 7 Foksiyonlara Dizileri Göndermek
- 8 Dizileri Sıralamak
- 9 Veri Bilimine Giriş: Anket Veri Analizi
- 10 Dizilerde Arama
- 11 Çok Boyutlu Diziler
- 12 Değişken Uzunluklu Diziler

Giriş

- Bu bölümde **dizi** veri yapısı incelenecektir
- **Diziler**, aynı türden ve birbirleri ile ilişkili birden fazla değer içerirler
- Diziler “statik” varlıklardır ve yaşam döngüsü boyunca yapıları sabit kalır

Diziler I

- Bir dizi, bellekte bitişik olarak depolanan aynı türdeki öğelerin bir grubudur
- Aşağıdaki örnek c adında, beş eleman içeren bir dizinin gösterimidir



- Dizideki belirli bir konuma veya öğeye erişmek için, dizinin adını ve ardından köşeli parantez (`[]`) içinde öğenin konum numarasını belirtiriz
- İlk eleman 0(sıfır) konumunda bulunur
- Elemanın konum değeri **indeks** olarak adlandırılır

Diziler II

- İndeks değeri 0 veya pozitif olmalıdır
- İndeksli bir ifade solda yer alarak atama işleminde kullanılabilir
 - `c[2] = 1000;`
- Dizinin ilk üç elemanın toplamını aşağıdaki gibi yazdırabiliriz
 - `printf("%d", c[0] + c[1] + c[2]);`
- Dizinin 3. konumundaki elemanın yarısını `x` değişkenine atama:
 - `x = c[3] / 2;`

Diziler

Öncelik

Operators	Grouping	Type
[] () ++ (<i>postfix</i>) -- (<i>postfix</i>)	left to right	highest
+ - ! ++ (<i>prefix</i>) -- (<i>prefix</i>) (<i>type</i>)	right to left	unary
* / %	left to right	multiplicative
+ -	left to right	additive
< <= > >=	left to right	relational
== !=	left to right	equality
&&	left to right	logical AND
	left to right	logical OR
?:	right to left	conditional
= += -= *= /= %=	right to left	assignment
,	left to right	comma

Dizileri Tanımlamak

- Dizileri tanımlayarak, dizinin türünü ve eleman sayısını belirtiriz
- Derleyici tanım sayesinde uygun miktarda yer ayıracaktır
- `int c[5];`
 - 5 elemanlı c tamsayı dizisi, indeksler 0-4 arasında
- `int b[100];`
- `int x[27];`

Dizi Örnekleri

Dizi Tanımlama ve Döngü İle Değerleri Atama

```
1 // fig06_01.c
2 // Bir dizinin elemanlarını sıfır olarak başlatma.
3 #include <stdio.h>
4
5 // main fonksiyonu programın çalışmasını başlatır
6 int main(void) {
7     int dizi[5]; // dizi, beş tamsayıdan oluşan bir dizidir
8
9     // dizi elemanlarını 0 yap
10    for (size_t indeks = 0; indeks < 5; ++indeks) {
11        dizi[indeks] = 0; // indeks konumundaki elemanı 0 yap
12    }
13
14    printf("%s%8s\n", "Eleman", "Değer");
15
16    // dizinin içeriğini tablo formatında yazdır
17    for (size_t indeks = 0; indeks < 5; ++indeks) {
18        printf("%6zu%7d\n", indeks, dizi[indeks]);
19    }
20 }
```


Dizi Örnekleri

Dizi Tanımlama ve Döngü İle Değerleri Atama

Eleman	Değer
0	0
1	0
2	0
3	0
4	0

Dizi Örnekleri

Tanımda Bir Diziyi Başlatıcı Listesiyle Başlatma

```
1 // fig06_02.c
2 // Bir dizinin elemanlarını bir başlatıcı liste ile başlatma.
3 #include <stdio.h>
4
5 // main fonksiyonu programın çalışmasını başlatır
6 int main(void) {
7     int dizi[5] = {32, 27, 64, 18, 95}; // dizi, bir başlatıcı liste ile
8     ↪ başlatılıyor
9
10    printf("%s%8s\n", "Eleman", "Değer");
11
12    // dizinin içeriğini tablo formatında yazdır
13    for (size_t indeks = 0; indeks < 5; ++indeks) {
14        printf("%6zu%7d\n", indeks, dizi[indeks]);
15    }
```

Dizi Örnekleri

Tanımda Bir Diziyi Başlatıcı Listesiyle Başlatma

Eleman	Değer
0	32
1	27
2	64
3	18
4	95

- `int n[5] = {0};` // tüm diziyi 0 yapar
- `int n[3] = {32, 27, 64, 18};` // hata
- Dizi boyutu verilmeden de ilk değerler belirlenebilir
- `int n[] = {1, 2, 3, 4, 5};`

Dizi Örnekleri

Dizi Boyutunu Sabit ile Belirleme ve Değerleri Hesaplayarak Başlatma

```
1 // fig06_03.c
2 // s dizisinin elemanlarını 2'den 10'a kadar olan çift sayılarla başlatma.
3 #include <stdio.h>
4 #define BOYUT 5 // dizinin maksimum boyutu
5
6 // main fonksiyonu programın çalışmasını başlatır
7 int main(void) {
8     // sembolik sabit BOYUT dizi boyutunu belirtmek için kullanılabilir
9     int dizi[BOYUT] = {0}; // dizi, BOYUT kadar elemana sahiptir
10
11     for (size_t indeks = 0; indeks < BOYUT; ++indeks) { // değerleri ayarla
12         dizi[indeks] = 2 + 2 * indeks;
13     }
14
15     printf("%s%s\n", "Eleman", "Değer");
16
17     // dizinin içeriğini tablo formatında yazdır
18     for (size_t indeks = 0; indeks < BOYUT; ++indeks) {
19         printf("%6zu%7d\n", indeks, dizi[indeks]);
20     }
21 }
```

Dizi Örnekleri

Dizi Boyutunu Sabit ile Belirleme ve Değerleri Hesaplayarak Başlatma

Eleman	Değer
0	2
1	4
2	6
3	8
4	10

- *#define* ifadelerinde ; kullanılmamalıdır
- Sembolik sabitler büyük harfle yazılmalıdır
- Sembolik sabitlere atama yapmak derleme hatasıdır

Dizi Örnekleri

Dizinin Elemanlarını Toplama

```
1 // fig06_04.c
2 // Bir dizinin elemanlarının toplamını hesaplama.
3 #include <stdio.h>
4 #define BOYUT 5
5
6 // main fonksiyonu programın çalışmasını başlatır
7 int main(void) {
8     // bir başlatıcı liste kullanarak diziyi başlat
9     int dizi[BOYUT] = {1, 2, 3, 4, 5};
10    int toplam = 0; // dizinin toplamı
11
12    // dizi elemanlarının toplamını hesapla
13    for (size_t i = 0; i < BOYUT; ++i) {
14        toplam += dizi[i];
15    }
16
17    printf("Dizinin toplam değeri: %d\n", toplam);
18 }
```

Dizi Örnekleri

Örnek

Yirmi öğrenciden öğrenci kafeteryasındaki yiyeceklerin kalitesini 1'den 5'e kadar bir ölçekte derecelendirmeleri istendi (1 berbat, 5 mükemmel anlamına gelir). 20 yanıtı bir tam sayı dizisine yerleştirin ve anketin sonuçlarını özetleyin.

Dizi Örnekleri

Öğrenci Anketini Analiz Etme

```
1 // fig06_05.c
2 // Bir öğrenci anketini analiz etme.
3 #include <stdio.h>
4 #define CEVAP_BOYUTU 20 // dizi boyutlarını tanımla
5 #define FREKANS_BOYUTU 6
6
7 // main fonksiyonu programın çalışmasını başlatır
8 int main(void) {
9     // anket cevaplarını cevaplar dizisine yerleştir
10    int cevaplar[CEVAP_BOYUTU] =
11        {1, 2, 5, 4, 3, 5, 2, 1, 3, 1, 4, 3, 3, 3, 2, 3, 3, 2, 2, 5};
12
13    // frekans sayacılarını 0 ile başlat
14    int frekans[FREKANS_BOYUTU] = {0};
15
```


Dizi Örnekleri

Öğrenci Anketini Analiz Etme

```
16 // her cevap için cevaplar dizisinin bir elemanının
17 // değerini seç ve bu değeri frekans dizisinde bir
18 // indis olarak kullanarak elemanı artır
19 for (size_t answer = 0; answer < CEVAP_BOYUTU; ++answer) {
20     ++frekans[cevaplar[answer]];
21 }
22
23 // sonuçları göster
24 printf("%s%12s\n", "Puan", "Frekans");
25
26 // frekansları tablo formatında yazdır
27 for (size_t rating = 1; rating < FREKANS_BOYUTU; ++rating) {
28     printf("%4zu%12d\n", rating, frekans[rating]);
29 }
30 }
```

Dizi Örnekleri

Öğrenci Anketini Analiz Etme

Puan	Frekans
1	3
2	5
3	7
4	2
5	3

- `frekans[0]` kullanılmamaktadır
- Dizide hatalı anket değerleri olsaydı hata alırdık(örn: 9)
- Dizi indeksi sıfırdan küçük ve dizi boyutunun 1 eksiğinden büyük olmamalıdır

Dizi Örnekleri

Çubuk Grafik Gösterme

```
1 // fig06_06.c
2 // Çubuk grafik gösterme.
3 #include <stdio.h>
4 #define BOYUT 5
5
6 // main fonksiyonu programın çalışmasını başlatır
7 int main(void) {
8     // bir başlatıcı liste kullanarak dizi n'yi başlat
9     int dizi[BOYUT] = {19, 3, 15, 7, 11};
10
11     printf("%s%13s%17s\n", "Eleman", "Değer", "Çubuk Grafik");
12 }
```

Dizi Örnekleri

Çubuk Grafik Gösterme

```
13 // dizi n'nin her elemanı için çubuk grafik bir çubuk yazdır
14 for (size_t i = 0; i < BOYUT; ++i) {
15     printf("%6zu%12d%4s", i, dizi[i], "");
16
17     for (int j = 1; j <= dizi[i]; ++j) { // bir çubuk yazdır
18         printf("%c", '*');
19     }
20
21     puts(""); // bir çubuğu yeni satır ile bitir
22 }
23 }
```

Dizi Örnekleri

Çubuk Grafik Gösterme

Eleman	Değer	Çubuk Grafik
0	19	*****
1	3	***
2	15	*****
3	7	*****
4	11	*****

Dizi Örnekleri

60M Zar Atma ve Özetleme

```
1 // fig06_07.c
2 // Altı yüzlü bir zar 60.000.000 kez atılıyor
3 #include <stdio.h>
4 #include <stdlib.h>
5 #include <time.h>
6 #define BOYUT 7
7
8 // main fonksiyonu programın çalışmasını başlatır
9 int main(void) {
10     srand(time(NULL)); // rastgele sayı üreticiyi başlat
11
12     int frekans[BOYUT] = {0}; // tüm frekans sayacılarını 0 ile başlat
13 }
```

Dizi Örnekleri

60M Zar Atma ve Özetleme

```
14 // zarı 60.000.000 kez at
15 for (int atis = 1; atis <= 60000000; ++atis) {
16     size_t yuz = 1 + rand() % 6;
17     ++frekans[yuz]; // Fig. 5.5'teki tüm switch yapısının yerine geçiyor
18 }
19
20 printf("%s%17s\n", "Yüz", "Frekans");
21
22 // frekans elemanlarını 1-6 arası tablo formatında yazdır
23 for (size_t yuz = 1; yuz < BOYUT; ++yuz) {
24     printf("%3zu%17d\n", yuz, frekans[yuz]);
25 }
26 }
```

Dizi Örnekleri

60M Zar Atma ve Özetleme

Yüz	Frekans
1	9998894
2	10000414
3	10003178
4	10000694
5	9998428
6	9998392

Metinleri Depolamak ve Değiştirmek İçin Karakter Dizilerini Kullanmak

- Karakter dizileri metinleri(string) depolamak için kullanılabilir
 - `char string1[] = "first";`
- C'de metinleri sonlandırmak için **null karakteri** kullanılır
- `string1` metni null ile birlikte toplam 6 karakterden oluşur
- Liste başlatıcı ile aşağıdaki gibi de tanımlanabilir
 - `char string1[] = {'f', 'i', 'r', 's', 't', '\0'};`
- Karakter dizisine aşağıdaki gibi girdi alınabilir

```
char string2[20];
scanf("%19s", string2);
```
- `printf("%s\n", string2);` ifadesi ile yazdırılabilir

Karakter Dizisi Örneği

```
1 // fig06_08.c
2 // Karakter dizilerini (string) ele alma.
3 #include <stdio.h>
4 #define BOYUT 20
5
6 // main fonksiyonu programın çalışmasını başlatır
7 int main(void) {
8     char dizi1[BOYUT] = ""; // 20 karakterlik yer ayırır
9     char dizi2[] = "string literal"; // 15 karakterlik yer ayırır
10
11     // kullanıcıdan bir string iste ve bunu dizi1'e oku
12     printf("%s", "Bir string girin (en fazla 19 karakter): ");
13     scanf("%19s", dizi1); // en fazla 19 karakter oku
14
15     // stringleri yazdır
16     printf("dizi1: %s\ndizi2: %s\n", dizi1, dizi2);
17     puts("Dizi1'in karakterleri arasında boşluk bırakarak yazdırılıyor.");
18
19     // null karakterine ulaşılan kadar karakterleri yazdır
20     for (size_t i = 0; i < BOYUT && dizi1[i] != '\0'; ++i) {
21         printf("%c ", dizi1[i]);
22     }
23
24     puts("");
25 }
```

Karakter Dizisi Örneği

```
Bir string girin (en fazla 19 karakter): Merhaba dünya  
dizi1: Merhaba  
dizi2: string literal  
Dizi1'in karakterleri arasında boşluk bırakarak yazdırılıyor:  
M e r h a b a
```

Statik ve Otomatik Yerel Diziler

```
1 // fig06_09.c
2 // Statik diziler açıkça başlatılmadığında sıfır ile başlatılır.
3 #include <stdio.h>
4
5 void statikDiziBaslat(void); // fonksiyon prototipi
6 void otomatikDiziBaslat(void); // fonksiyon prototipi
7
8 // main fonksiyonu programın çalışmasını başlatır
9 int main(void) {
10     puts("Her bir fonksiyona ilk çağrı:");
11     statikDiziBaslat();
12     otomatikDiziBaslat();
13
14     puts("\n\nHer bir fonksiyona ikinci çağrı:");
15     statikDiziBaslat();
16     otomatikDiziBaslat();
17     puts("");
18 }
19
```

Statik ve Otomatik Yerel Diziler

```
20 // statik yerel bir dizi göstermek için fonksiyon
21 void statikDiziBaslat(void) {
22     // fonksiyon çağrılmadan önce elemanlar sıfır ile başlatılır
23     static int dizi1[3];
24
25     puts("\nstatikDiziBaslat'a girerken değerler:");
26
27     // dizi1'in içeriğini yazdır
28     for (size_t i = 0; i <= 2; ++i) {
29         printf("dizi1[%zu] = %d ", i, dizi1[i]);
30     }
31
32     puts("\nstatikDiziBaslat'tan çıkarken değerler:");
33
34     // dizi1'in içeriğini değiştir ve yazdır
35     for (size_t i = 0; i <= 2; ++i) {
36         printf("dizi1[%zu] = %d ", i, dizi1[i] += 5);
37     }
38 }
39
```

Statik ve Otomatik Yerel Diziler

```
40 // otomatik yerel bir dizi göstermek için fonksiyon
41 void otomatikDiziBaslat(void) {
42     // fonksiyon her çağrıldığında elemanlar başlatılır
43     int dizi2[3] = {1, 2, 3};
44
45     puts("\notomatikDiziBaslat'a girerken değerler:");
46
47     // dizi2'nin içeriğini yazdır
48     for (size_t i = 0; i <= 2; ++i) {
49         printf("dizi2[%zu] = %d ", i, dizi2[i]);
50     }
51
52     puts("\notomatikDiziBaslat'tan çıkarken değerler:");
53
54     // dizi2'nin içeriğini değiştir ve yazdır
55     for (size_t i = 0; i <= 2; ++i) {
56         printf("dizi2[%zu] = %d ", i, dizi2[i] += 5);
57     }
58 }
```

Statik ve Otomatik Yerel Diziler

```
otomatikDiziBaslat'a girerken değerler:  
dizi2[0] = 1  dizi2[1] = 2  dizi2[2] = 3  
otomatikDiziBaslat'tan çıkarken değerler:  
dizi2[0] = 6  dizi2[1] = 7  dizi2[2] = 8
```

Her bir fonksiyona ikinci çağrı:

```
statikDiziBaslat'a girerken değerler:  
dizi1[0] = 5  dizi1[1] = 5  dizi1[2] = 5  
statikDiziBaslat'tan çıkarken değerler:  
dizi1[0] = 10  dizi1[1] = 10  dizi1[2] = 10
```

```
otomatikDiziBaslat'a girerken değerler:  
dizi2[0] = 1  dizi2[1] = 2  dizi2[2] = 3  
otomatikDiziBaslat'tan çıkarken değerler:  
dizi2[0] = 6  dizi2[1] = 7  dizi2[2] = 8
```

Foksiyonlara Dizileri Göndermek

```
1 // fig06_10.c
2 // Dizi adı, dizinin ilk elemanının adresiyle aynıdır.
3 #include <stdio.h>
4
5 // main fonksiyonu programın çalışmasını başlatır
6 int main(void) {
7     char dizi[5] = ""; // boyutu 5 olan bir dizi tanımla
8
9     printf("    dizi = %p\n&dizi[0] = %p\n    &dizi = %p\n",
10           dizi, &dizi[0], &dizi);
11 }
```

```
dizi = 0x7fffffffed233
&dizi[0] = 0x7fffffffed233
&dizi = 0x7fffffffed233
```


Foksiyonlara Dizileri Göndermek

```
1 // fig06_11.c
2 // Dizileri ve dizinin bireysel elemanlarını fonksiyonlara geçirme.
3 #include <stdio.h>
4 #define BOYUT 5
5
6 // fonksiyon prototipleri
7 void diziDegistir(int b[], size_t boyut);
8 void elemanDegistir(int e);
9
10 // main fonksiyonu programın çalışmasını başlatır
11 int main(void) {
12     int a[BOYUT] = {0, 1, 2, 3, 4}; // dizi a'yı başlat
13
14     puts("Tüm diziyi referansla geçmenin etkileri:\n\n"
15         "Orijinal dizinin değerleri:");
16
17     // orijinal diziyi yazdır
18     for (size_t i = 0; i < BOYUT; ++i) {
19         printf("%3d", a[i]);
20     }
21 }
```

Foksiyonlara Dizileri Göndermek

```

22 puts(""); // yeni satır yazdır
23
24 diziDegistir(a, BOYUT); // dizi a'yı referansla diziDegistir fonksiyonuna
   ↪ geç
25 puts("Değiştirilen dizinin değerleri:");
26
27 // değiştirilmiş diziyi yazdır
28 for (size_t i = 0; i < BOYUT; ++i) {
29     printf("%3d", a[i]);
30 }
31
32 // a[3]'ün değerini yazdır
33 printf("\n\nDizi elemanını değer olarak geçmenin etkileri:\n\n"
34        "a[3]'ün değeri %d\n", a[3]);
35
36 elemanDegistir(a[3]); // dizi elemanı a[3]'ü değer olarak geç
37
38 // a[3]'ün değerini yazdır
39 printf("a[3]'ün değeri %d\n", a[3]);
40 }
41

```

Foksiyonlara Dizileri Göndermek

```
42 // diziDegistir fonksiyonunda, "b" orijinal dizi "a"yı bellekte işaret eder
43 void diziDegistir(int b[], size_t boyut) {
44     // her dizi elemanını 2 ile çarp
45     for (size_t j = 0; j < boyut; ++j) {
46         b[j] *= 2; // aslında orijinal diziyi değiştirir
47     }
48 }
49
50 // elemanDegistir fonksiyonunda, "e" main'den geçilen dizi elemanı a[3]'ün
   ↪ yerel bir kopyasıdır
51 void elemanDegistir(int e) {
52     e *= 2; // parametreyi 2 ile çarp
53     printf("elemanDegistir fonksiyonundaki değer %d\n", e);
54 }
```

Foksiyonlara Dizileri Göndermek

Tüm diziyi referansla geçmenin etkileri:

Orijinal dizinin değerleri:

0 1 2 3 4

Değiştirilen dizinin değerleri:

0 2 4 6 8

Dizi elemanını değer olarak geçmenin etkileri:

a[3]'ün değeri 6

elemanDegistir fonksiyonundaki değeri 12

a[3]'ün değeri 6

Foksiyonlara Dizileri Göndermek

- Aşağıdaki kod `const` ifadesinden kaynaklı hata verir

```
void diziyiDegistirmeyeCalis(const int b[]) {  
    b[0] /= 2; // hata  
    b[1] /= 2; // hata  
    b[2] /= 2; // hata  
}
```

Dizileri Sıralamak

```
1 // fig06_12.c
2 // Bir dizinin elemanlarını artan sıraya göre sıralama.
3 #include <stdio.h>
4 #define BOYUT 10
5
6 // main fonksiyonu programın çalışmasını başlatır
7 int main(void) {
8     int a[BOYUT] = {2, 6, 4, 8, 10, 12, 89, 68, 45, 37};
9
10    puts("Veri elemanları orijinal sırada");
11
12    // orijinal diziyi yazdır
13    for (size_t i = 0; i < BOYUT; ++i) {
14        printf("%4d", a[i]);
15    }
16
17    // kabarcık sıralama (bubble sort)
18    // geçiş sayısını kontrol eden döngü
19    for (int gecis = 1; gecis < BOYUT; ++gecis) {
20        // her geçişteki karşılaştırmaları kontrol eden döngü
21        for (size_t i = 0; i < BOYUT - 1; ++i) {
```

Dizileri Sıralamak

```
22 // ardışık elemanları karşılaştır ve ilk eleman
23 // ikinci elemandan büyükse yer değiştir
24 if (a[i] > a[i + 1]) {
25     int gecici = a[i];
26     a[i] = a[i + 1];
27     a[i + 1] = gecici;
28 }
29 }
30 }
31
32 puts("\nVeri elemanları artan sırada");
33
34 // sıralanmış diziyi yazdır
35 for (size_t i = 0; i < BOYUT; ++i) {
36     printf("%4d", a[i]);
37 }
38
39 puts("");
40 }
```

Dizileri Sıralamak

```
Veri elemanları orijinal sırada  
2 6 4 8 10 12 89 68 45 37  
Veri elemanları artan sırada  
2 4 6 8 10 12 37 45 68 89
```


Veri Bilimine Giriş: Anket Veri Analizi

- Kod incelnecek...

Dizilerde Arama

```
1 // fig06_14.c
2 // Bir dizinin doğrusal araması.
3 #include <stdio.h>
4 #define BOYUT 100
5
6 // fonksiyon prototipi
7 int dogrusalArama(const int dizi[], int anahtar, size_t boyut);
8
9 // main fonksiyonu programın çalışmasını başlatır
10 int main(void) {
11     int a[BOYUT] = {0}; // dizi a'yı oluştur
12
13     // bazı veriler oluştur
14     for (size_t x = 0; x < BOYUT; ++x) {
15         a[x] = 2 * x;
16     }
17
18     printf("Arama anahtarını girin: ");
19     int aramaAnahtari = 0; // dizide bulunacak değer
20     scanf("%d", &aramaAnahtari);
21 }
```

Dizilerde Arama

```

22 // aramaAnahtarını dizi a'da bulmaya çalış
23 int altIndeks = dogrusalArama(a, aramaAnahtari, BOYUT);
24
25 // sonuçları göster
26 if (altIndeks != -1) {
27     printf("Değer %d. indeksde bulundu\n", altIndeks);
28 }
29 else {
30     puts("Değer bulunamadı");
31 }
32 }
33
34 // anahtarı her dizi elemanı ile karşılaştır, anahtar bulunduğunda
35 // veya dizinin sonuna gelindiğinde, anahtar bulunursa dizinin indeksini
36 // geri döndür, bulunmazsa -1 döndür
37 int dogrusalArama(const int dizi[], int anahtar, size_t boyut) {
38     // diziyi döngü ile tara
39     for (size_t n = 0; n < boyut; ++n) {
40         if (dizi[n] == anahtar) {
41             return n; // anahtarın bulunduğu konumu döndür
42         }
43     }
44
45     return -1; // anahtar bulunamadı
46 }

```

Dizilerde Arama

```
Arama anahtarını girin: 36  
Değer 18. indeksde bulundu  
Arama anahtarını girin: 37  
Değer bulunamadı
```

İkili Arama

- Kod incelenecek...

Çok Boyutlu Diziler

	Column 0	Column 1	Column 2	Column 3
Row 0	a[0][0]	a[0][1]	a[0][2]	a[0][3]
Row 1	a[1][0]	a[1][1]	a[1][2]	a[1][3]
Row 2	a[2][0]	a[2][1]	a[2][2]	a[2][3]

Diagram illustrating a 2D array structure with 3 rows and 4 columns. The array is represented as a grid of cells. The first column is labeled 'Column 0', the second 'Column 1', the third 'Column 2', and the fourth 'Column 3'. The first row is labeled 'Row 0', the second 'Row 1', and the third 'Row 2'. The cells are labeled with their respective indices: a[0][0], a[0][1], a[0][2], a[0][3] for Row 0; a[1][0], a[1][1], a[1][2], a[1][3] for Row 1; and a[2][0], a[2][1], a[2][2], a[2][3] for Row 2. Arrows point from the labels 'Array name', 'Row index', and 'Column index' to the corresponding parts of the cell a[2][1]: 'Array name' points to 'a', 'Row index' points to '[2]', and 'Column index' points to '[1]'.

Çok Boyutlu Diziler

```
1 // fig06_16.c
2 // Çok boyutlu dizilerin başlatılması.
3 #include <stdio.h>
4
5 void diziyiYazdir(int a[][3]); // fonksiyon prototipi
6
7 // main fonksiyonu programın çalışmasını başlatır
8 int main(void) {
9     int dizi1[2][3] = {{1, 2, 3}, {4, 5, 6}};
10    puts("dizi1'deki değerler satırlara göre:");
11    diziyiYazdir(dizi1);
12
13    int dizi2[2][3] = {{1, 2, 3}, {4, 5}};
14    puts("dizi2'deki değerler satırlara göre:");
15    diziyiYazdir(dizi2);
16
17    int dizi3[2][3] = {{1, 2}, {4}};
18    puts("dizi3'teki değerler satırlara göre:");
19    diziyiYazdir(dizi3);
20 }
21
```

Çok Boyutlu Diziler

```
22 // iki satır ve üç sütun içeren diziyi yazdıran fonksiyon
23 void diziYazdir(int a[][3]) {
24     // satırlar üzerinden döngü
25     for (size_t i = 0; i <= 1; ++i) {
26         // sütun değerlerini yazdır
27         for (size_t j = 0; j <= 2; ++j) {
28             printf("%d ", a[i][j]);
29         }
30         printf("\n"); // yeni bir satır başlat
31     }
32 }
```


Çok Boyutlu Diziler

```
dizi1'deki değerler satırlara göre:  
1 2 3  
4 5 6  
dizi2'deki değerler satırlara göre:  
1 2 3  
4 5 0  
dizi3'teki değerler satırlara göre:  
1 2 0  
4 0 0
```

Çok Boyutlu Diziler

- fig06_17 incelenecek

Değişken Uzunluklu Diziler

- fig06_18 incelenecek