

✓ DİZİLER

✓ Tek boyutlu diziler

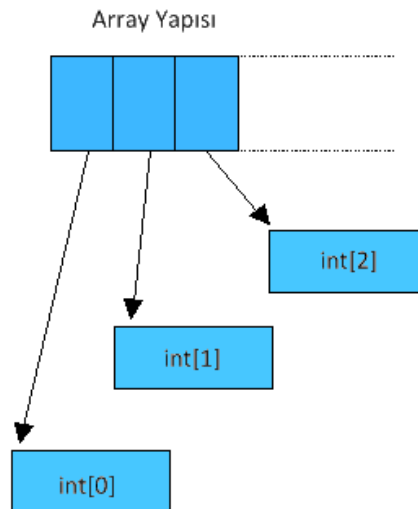
`tür[] diziIsmi=new tür[boyut];`

`tür`: veri türü

`diziIsmi`: diziye programcının verdiği isim

`new`: anahtar kelime

`boyut`: dizinin eleman sayısı veya değişken sayısı (değişkenler 0'dan başlanarak isimlendirilir!). Belirlediğimiz boyut kadar bellekte yer ayrılır.



Örnekler:

```
short[] ornek=new short[3];    //ornek[0], ornek[1] ve ornek[2] değişkenleri
```

```
int[] ort=new int[5];    ort[4]=100;    Console.WriteLine(ort[4]);
```

```
string[] ad=new string[100];    ad[0]="Ali";
```

```
char[] s=new char[10];    s[7]='E';
```

```
double[] D=new double[200];    D[45]= Convert.ToDouble(Console.ReadLine());
```

Tek boyutlu dizilerin farklı tanımlanma şekilleri ve değer atama yöntemleri aşağıdaki örnekte verilmektedir:

```
static void Main(string[] args)
{
    //Birinci tanımlama şekli
    int[] dizi;
    dizi = new int[4];

    //İkinci tanımlama şekli
    int[] dizi2 = new int[6];

    //Birinci ve ikinci tanımlama için değer ataması
    dizi[0] = 5;
    dizi[1] = 15;
    dizi[2] = 25;
    //dizi[3] = 35; Değer atanmayan elemanlar 0 kabul edilir.

    //Birinci ve ikinci tanımlama için değer ataması
    for (int i = 0; i < 6; i++)
    {
        dizi2[i]=Convert.ToInt16(Console.ReadLine());
    }

    //Üçüncü tanımlama şekli ve değer ataması
    int[] dizi3 = new int[5] { 10, 20, 30, 40, 50 };

    //Dördüncü tanımlama şekli ve değer ataması
    int[] dizi4 = { 10, 20, 30, 40, 50 };

    //Dizinin yazdırılması
    for (int i = 0; i < dizi4.Length; i++)
    //dizi4.Length komutu dizi4'ün boyutunu (=5) verir.
    {
        Console.Write(" {0}", dizi4[i]);
    }
    Console.ReadKey();
}
```

Karşılaştırma

`string.Compare(string A, string B)` //int türünde değerle döner

A ile B karşılaştırılır ve

B>A ise c=1

B=A ise c=0

B<A ise c=-1

sonucu üretilir.

Örnekler:

```
int c = string.Compare("Abdurrahman", "Abdullah"); // c=1
```

```
int c = string.Compare("Ayşe", "Fatma"); // c=-1
```

```
int c = string.Compare("Mehmet", "Mehmet"); // c=0
```

```
int c = string.Compare("Ali", "ali"); // c=1
```

```
int c = string.Compare("3a", "a3"); // c=-1
```

Örnek Problem:

Sınıf mevcudu ile öğrencilerin sırasıyla adı, soyadı ve öğrenci numarası klavyeden girildiğinde öğrencileri,

- Adına (alfabetik sıraya göre)
- Soyadına (alfabetik sıraya göre)
- Öğrenci numarasına (küçükten büyüğe)

göre sıralayarak sıralı listeyi temizlenen ekranda gösterecek programı C# dilinde yazınız.

Not: Öğrenci numaraları sadece rakamlardan oluşmaktadır.

Çözüm (a):

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace ConsoleApplication16
{
    class Program
    {
        static void Main(string[] args)
        {
            short boyut, i, j, N; int C;
            string A,S;

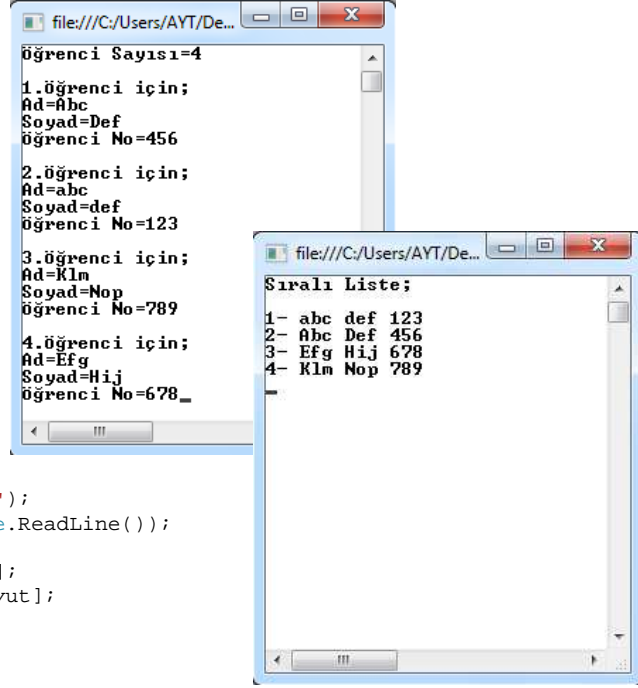
            Console.WriteLine("Öğrenci Sayısı=");
            boyut = Convert.ToInt16(Console.ReadLine());

            string[] ad = new string[boyut];
            string[] soyad = new string[boyut];
            short[] no = new short[boyut];

            for (i = 0; i < boyut; i++)
            {
                Console.WriteLine("\n{i}.Öğrenci için;", i + 1);
                Console.Write("Ad="); ad[i] = Console.ReadLine();
                Console.Write("Soyad="); soyad[i] = Console.ReadLine();
                Console.Write("Öğrenci No="); no[i] = Convert.ToInt16(Console.ReadLine());
            }

            for (i = 0; i < boyut; i++)
            {
                for (j = Convert.ToInt16(i + 1); j < boyut; j++)
                {
                    C = string.Compare(ad[i], ad[j]);
                    if (C == 1)
                    {
                        A = ad[i]; S = soyad[i]; N = no[i];
                        ad[i] = ad[j]; soyad[i] = soyad[j]; no[i] = no[j];
                        ad[j] = A; soyad[j] = S; no[j] = N;
                    }
                }
            }

            Console.Clear();
            Console.WriteLine("Sıralı Liste;\n");
            for (i = 0; i < boyut; i++)
            {
                Console.WriteLine("{0}- {1} {2} {3}\n", i+1,ad[i], soyad[i], no[i]);
            }
            Console.ReadKey();
        }
    }
}
```



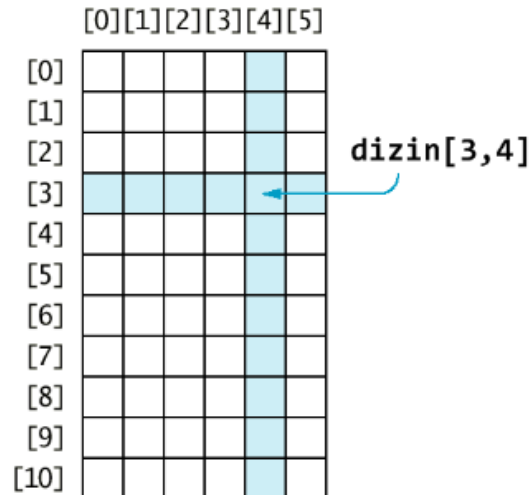
Çözüm (b): Ödev

Çözüm (c): Ödev

✓ *İki boyutlu diziler*

```
tür[,] diziIsmi=new tür[boyut1,boyut2];
```

```
int[,] dizi=new int[11,6];
```



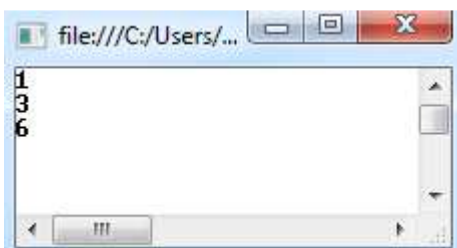
```
dizi.Length; // toplam eleman sayısı (=66)  
dizi.GetLength(0); // satır sayısı (=11)  
dizi.GetLength(1); // sütun sayısı (=6)
```

Örnek:

$$\text{dizi} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}$$

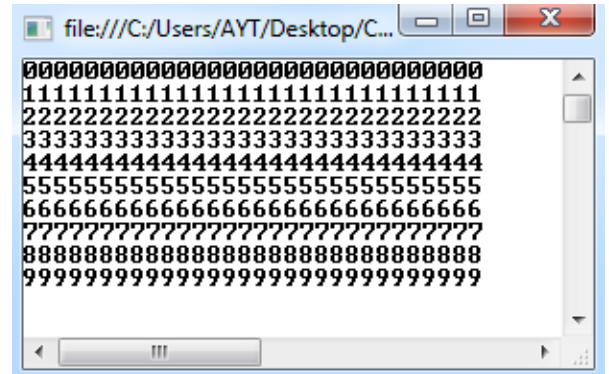
dizi[0,0] = 1	dizi[0,1] = 2
dizi[1,0] = 3	dizi[1,1] = 4
dizi[2,0] = 5	dizi[2,1] = 6

```
static void Main(string[] args)  
{  
    int[,] dizi = { { 1, 2 }, { 3, 4 }, { 5, 6 } };  
    Console.WriteLine(dizi[0, 0]); // 1  
    Console.WriteLine(dizi[1, 0]); // 3  
    Console.WriteLine(dizi[2, 1]); // 6  
    Console.ReadKey();  
}
```



Örnek:

```
static void Main(string[] args)
{
    double [,] ornek = new double[10,30];
    for (int i = 0; i < ornek.GetLength(0); i++)
    {
        for (int k = 0; k < ornek.GetLength(1); k++)
        {
            ornek[i, k] = i;
        }
    }
    for (int i = 0; i < ornek.GetLength(0); i++)
    {
        for (int k = 0; k < ornek.GetLength(1); k++)
        {
            Console.Write(ornek[i, k]);
        }
        Console.WriteLine("");
    }
    Console.ReadKey();
}
```



Örnek:

Aşağıdaki matrisi oluşturup ekrana yazdıracak program?

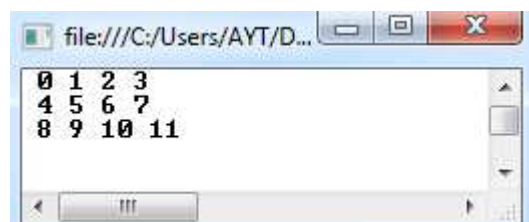
$$M = \begin{bmatrix} 0 & 1 & 2 & 3 \\ 4 & 5 & 6 & 7 \\ 8 & 9 & 10 & 11 \end{bmatrix}$$

```
static void Main(string[] args)
{
    int sayi = 0, i, j;
    int[,] dizi = new int[3, 4];

    for (i = 0; i < 3; i++)
        for (j = 0; j < 4; j++)
        {
            dizi[i, j] = sayi;
            sayi++;
        }

    for (i = 0; i < 3; i++)
    {
        for (j = 0; j < 4; j++)
        {
            Console.Write(" {0}", dizi[i, j]);
        }
        Console.WriteLine("");
    }

    Console.ReadKey();
}
```



Örnek:

Aşağıdaki matrisi oluşturup ekrana yazdıracak program?

$$M = \begin{bmatrix} 2 & 3 & 4 & 5 & 6 & 7 \\ 3 & 4 & 5 & 6 & 7 & 8 \\ 4 & 5 & 6 & 7 & 8 & 9 \\ 5 & 6 & 7 & 8 & 9 & 10 \\ 6 & 7 & 8 & 9 & 10 & 11 \\ 7 & 8 & 9 & 10 & 11 & 12 \end{bmatrix}$$

```
static void Main(string[] args)
{
    int [,] M = new int[6,6];
    int i, j;

    for (i = 0; i < 6; i++)
        for (j = 0; j < 6; j++)
            M[i,j] = i+j+2;

    for (i = 0; i < M.GetLength(0); i++)
    {
        for (j = 0; j < M.GetLength(1); j++)
            Console.Write(" {0}", M[i,j]);
        Console.WriteLine("");
    }

    Console.ReadKey();
}
```

Örnek:

$n \times m$ boytundaki iki matrisin toplamını ve farkını hesaplayıp ekrana yazdıracak program?

$$C_{n \times m} = A_{n \times m} \pm B_{n \times m}$$
$$c_{i,j} = a_{i,j} \pm b_{i,j}$$

```

static void Main(string[] args)
{
    short n, m, i, j;
    n = Convert.ToInt16(Console.ReadLine());
    m = Convert.ToInt16(Console.ReadLine());
    int[,] A = new int[n, m];
    int[,] B = new int[n, m];
    int[,] C = new int[n, m];

    for (i = 0; i < n; i++)
        for (j = 0; j < m; j++)
        {
            Console.Write("A[{0},{1}]= ", i, j);
            A[i, j] = Convert.ToInt32(Console.ReadLine());
        }

    for (i = 0; i < n; i++)
        for (j = 0; j < m; j++)
        {
            Console.Write("B[{0},{1}]= ", i, j);
            B[i, j] = Convert.ToInt32(Console.ReadLine());
        }

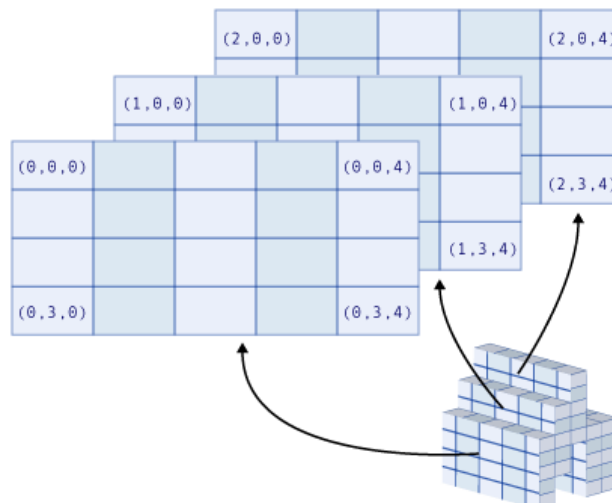
    for (i = 0; i < n; i++)
        for (j = 0; j < m; j++)
            C[i, j] = A[i, j] + B[i, j];

    Console.WriteLine("A+B=");
    for (i = 0; i < n; i++)
    {
        for (j = 0; j < m; j++)
            Console.Write(" {0}", C[i, j]);
        Console.WriteLine("");
    }
    Console.ReadKey();
}

```

✓ Üç boyutlu diziler

`tür[, ,] diziIsmi=new tür[boyut1,boyut2,boyut3];`



✓ Fonksiyonlar/Yöntemler

Belirli bir işi yapan program parçaları

```
<Access Specifier> <Return Type> <Method Name>(Parameter List)
{
    Method Body
}
```

Following are the various elements of a method:

- **Access Specifier:** This determines the visibility of a variable or a method from another class.
- **Return type:** A method may return a value. The return type is the data type of the value the method returns. If the method is not returning any values, then the return type is **void**.
- **Method name:** Method name is a unique identifier and it is case sensitive. It cannot be same as any other identifier declared in the class.
- **Parameter list:** Enclosed between parentheses, the parameters are used to pass and receive data from a method. The parameter list refers to the type, order, and number of the parameters of a method. Parameters are optional; that is, a method may contain no parameters.
- **Method body:** This contains the set of instructions needed to complete the required activity.

Örnek: 3'ün karesi

```
namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            KareBul(3);
            Console.ReadLine();
        }

        static void KareBul(int sayi)
        {
            Console.WriteLine(sayi * sayi);
        }
    }
}
```

Örnek: 0-100 arasındaki sayıların kareleri

```
namespace ConsoleApplication1
{
    class Program
    {
        static void KareYaz()
        {
            for (int i = 0; i <= 100; i++)
            {
                int sonuc = i * i;
                Console.WriteLine(sonuc);
            }
        }

        static void Main(string[] args)
        {
            KareYaz();
            Console.ReadLine();
        }
    }
}
```

Örnek: Girilen sayının karekökü

```
namespace ConsoleApplication1
{
    class Program
    {
        static double karekok(double deger)
        {
            return Math.Sqrt(deger);
        }

        static void Main(string[] args)
        {
            Console.Write("Karekök alınmasını istediğiniz sayıyı girin: ");
            double sayi1 = Convert.ToDouble(Console.ReadLine());
            Console.WriteLine(karekok(sayi1));
            Console.ReadLine();
        }
    }
}
```

Örnek: Girilen iki sayı arasındaki sayıların toplamı

```
namespace ConsoleApplication22
```

```
{
```

```
    class Program
```

```
    {
```

```
        static int topla(int sayi1, int sayi2)
```

```
        {
```

```
            int i, sonuc = 0;
```

```
            for (i = sayi1; i <= sayi2; i++)
```

```
            {
```

```
                sonuc += i;
```

```
            }
```

```
            return sonuc;
```

```
        }
```

```
        static void Main(string[] args)
```

```
        {
```

```
            Console.Write("İlk Sayıyı Giriniz = ");
```

```
            int sayi1 = Convert.ToInt32(Console.ReadLine());
```

```
            Console.Write("İkinci Sayıyı Giriniz = ");
```

```
            int sayi2 = Convert.ToInt32(Console.ReadLine());
```

```
            Console.WriteLine("İşlemin Sonucu = {0}", topla(sayi1, sayi2));
```

```
            Console.ReadKey();
```

```
        }
```

```
    }
```

```
}
```

Örnek: Permutasyon hesabı

```
namespace ConsoleApplication1
```

```
{
```

```
    class Program
```

```
    {
```

```
        static void Main(string[] args)
```

```
        {
```

```
            int n, r;
```

```
            double permutation;
```

```
            //Permutasyon hesaplama P(n,r)
```

```
            Console.WriteLine("n değerini girin: ");
```

```
            n = Convert.ToInt32(Console.ReadLine());
```

```
            Console.WriteLine("r değerini girin: ");
```

```
            r = Convert.ToInt32(Console.ReadLine());
```

```
            //P(n,r)=n!/(n-r)!
```

```
            permutation = CalculateFactorial(n) / CalculateFactorial(n - r);
```

```
            Console.WriteLine("\nSonuç: {0}", permutation);
```

```
            Console.ReadKey();
```

```
        }
```

```
        static double CalculateFactorial(int value)
```

```
        {
```

```
            int i;
```

```
            double result = 1;
```

```
            for (i = value; i > 0; i--)
```

```
            {
```

```
                result *= i;
```

```
            }
```

```
            return result;
```

```
        }
```

```
    }
```

```
}
```

