

10- DİZİLER (ARRAYS)

Örneğin 100 adet öğrencinin *adını* klavyeden girip, daha sonra bunları ekrandan yazdıralım. Bunun için, her öğrenci için ayrı bir değişken tanımlanmalıdır.

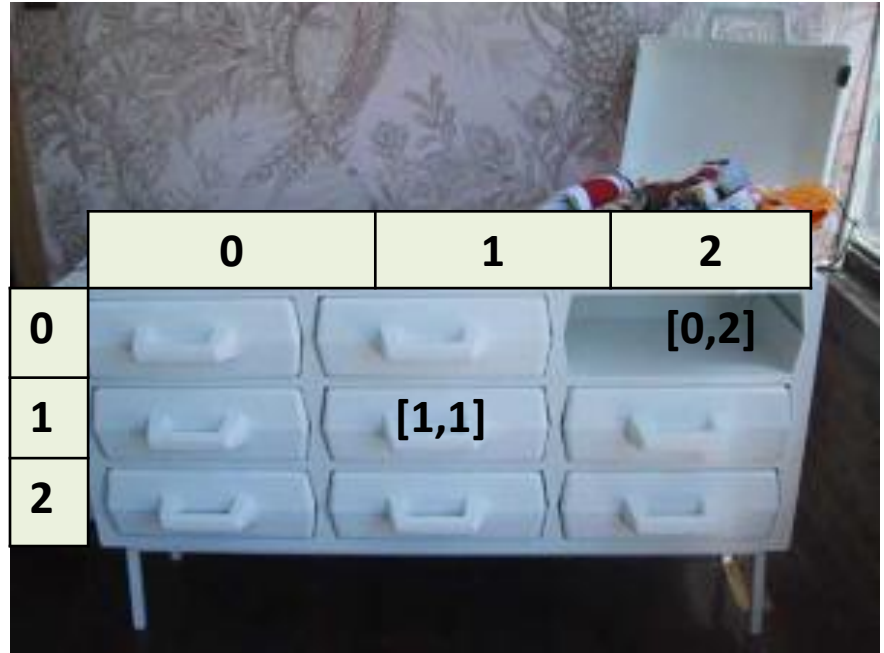
```
string ad1;  
string ad2;  
string ad3;  
...  
string ad100;
```

Ancak bu sağlıklı bir yöntem değildir. O halde aynı türdeki N adet değişken tek bir komutla tanımlanabilmeli ve daha sonra kolay bir şekilde kullanılabilmelidir.

Bu gibi işlemler için programlama dillerinde *dizi (array)* yapıları vardır.



Tek Boyutlu Dolap



İki Boyutlu Dolap

10- DİZİLER (ARRAYS)

Dizi : Aynı veri türünden birden çok değişken tanımlama;

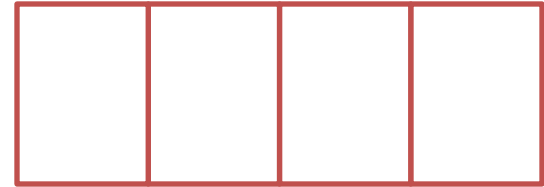
- ☐ Bir boyutlu,
- ☐ İki boyutlu ,
- ☐ Üç boyutlu ,
- ☐ Jagged

Dizi işlemleri genelde üç aşamada gerçekleşir.;

- I- Diziyi oluştur.
- II- Diziye eleman yerleştir.
- III- Dizi üzerinde işlem yap.

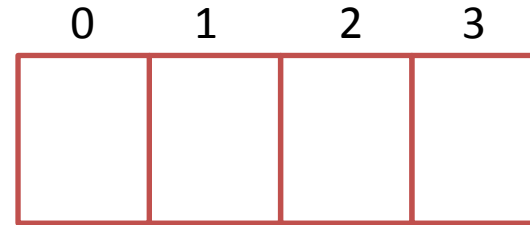
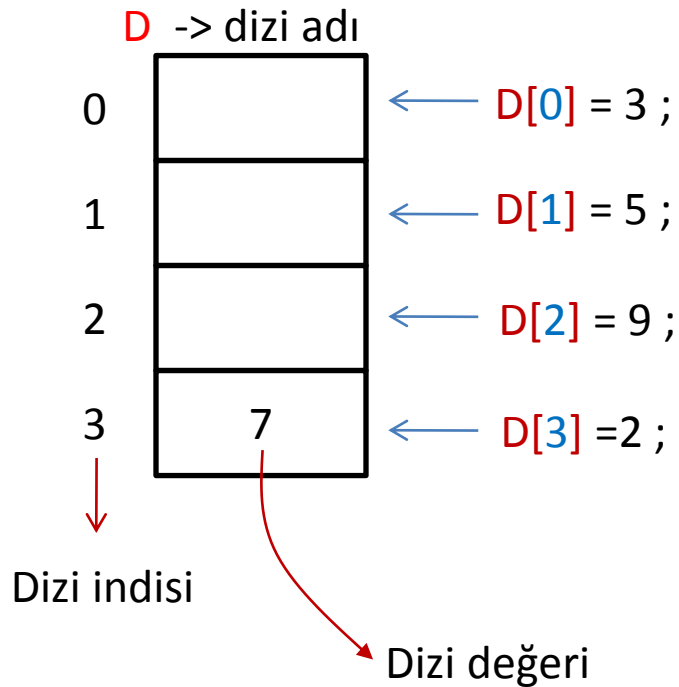
Diziler bellek modeline çok benzerler. Dolayısıyla bu gibi düşünmek kolaylık sağlayacaktır.

TEK BOYUTLU DİZİLER



10- DİZİLER (ARRAYS)

10.1- Tek Boyutlu diziler



Gösterim:

D [indis]

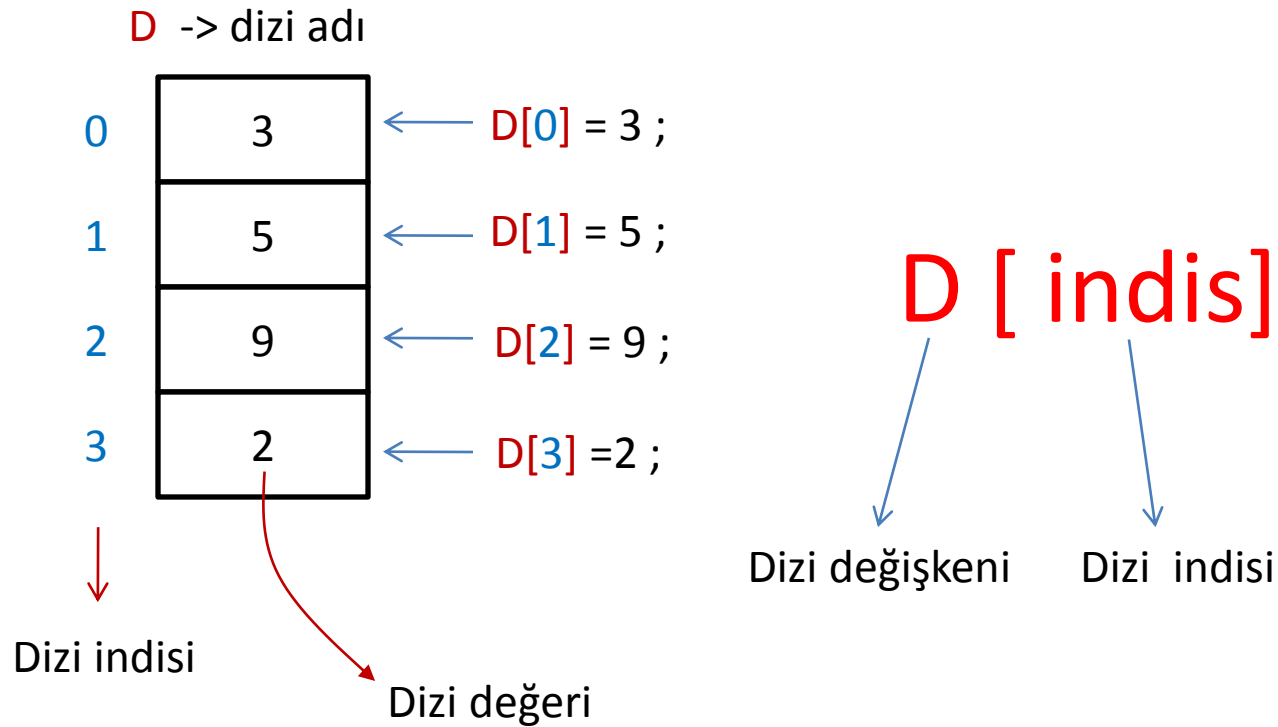
Gösterim:

D [3] = 2 ;

int x = **D** [0] ;

10- DİZİLER (ARRAYS)

10.1- Bir boyutlu diziler ;



10- DİZİLER (ARRAYS)

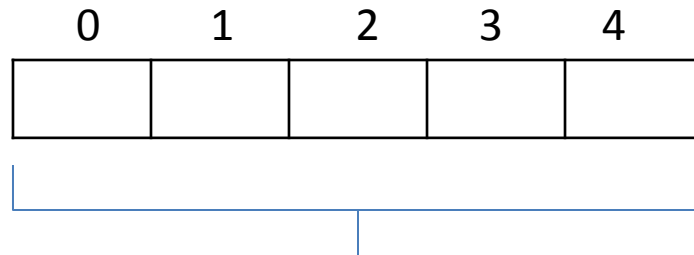
Bir boyutlu dizilerin tanımlanması :

```
int [ ] D = new int [ 5 ] ;
```

Dizideki elemanların tipi

Dizi değişkeni

Dizinin eleman sayısı



5 Hücre (eleman) her biri *int* türünden sayılar konabilecek

10- DİZİLER (ARRAYS)

Bir boyutlu dizilerin tanımlanması :

```
int [ ] d = new int [ 5];
```

```
int [ ] d = new int [ 3] { 2,5,6};
```

```
int [ ] d = { 2, 5, 6, 8 }; // 4 elemanlı dizi oluşur
```

<i>d</i>	0	1	2	3
	2	5	6	8

```
bool [ ] b = new bool [ 4] ;
```

<i>b</i>	0	1	2	3
	true	false	false	true

```
string [ ] d = new string [ 3] { "Kırmızı" , "Turuncu" , "Yeşil" } ;
```

10- DİZİLER (ARRAYS)

Bir boyutlu dizilerin tanımlanması :

```
char[ ] C = new char [ 4] { 'B' , 'A' , 'C' , 'A' } ;
```

C	0	1	2	3
	B	A	C	A

```
string D = "computer" ;
```

D	0	1	2	3	4	5	6	7
	c	o	m	p	u	t	e	r

Burada : D [0] ->c, D[1]->o gibidir

Not: *string* ifadeler *char* türünden tek boyutlu dizidirler .

10- DİZİLER (ARRAYS)

```
static void Main(string[] args)
{
    int [ ] D = new int [ 3 ];
    D[ 0 ] = 9 ;
    D[ 1 ] = 4 ;
    D[ 2 ] = 7 ;
    foreach( int x in D ) Console.Write( x );

    for( int i = 0 ; i < D.Length ; i++ ) Console.WriteLine( D [ i ] );
}
```

D[0]	D[1]	D[2]
0	1	2
9	4	7

D.Length : dizinin eleman sayısını bulur.
Örnekteki dizi için 3 değerini verecektir.

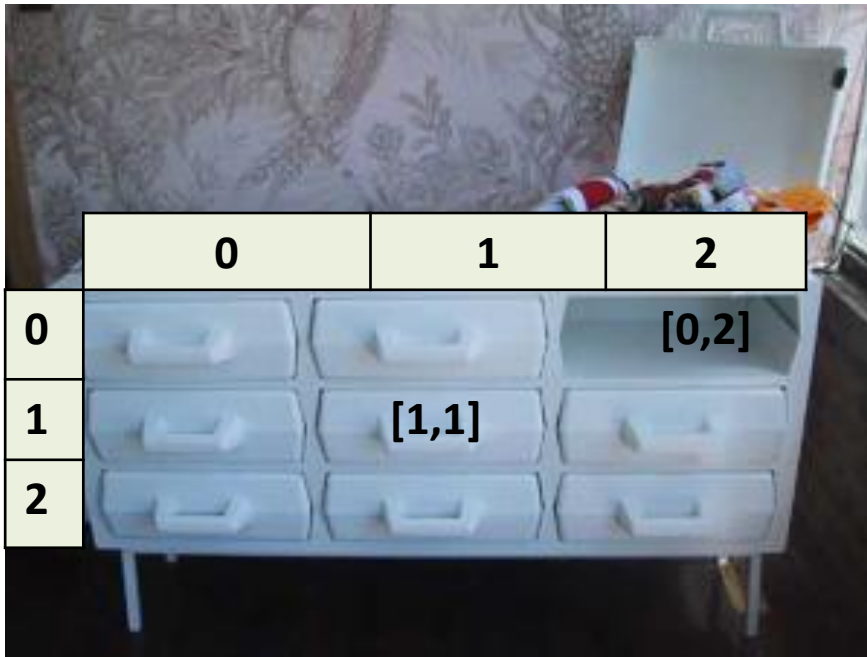
10- DİZİLER (ARRAYS)

```
using System;
using System.Collections;
namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            string [ ] D = new string [ 2 ];

            D[ 0 ] = "Ali Can";
            D[ 1 ] = "Elif Kara";

            foreach ( string s in D) Console.WriteLine ( s );
        }
    }
}
```

İKİ BOYUTLU DİZİLER



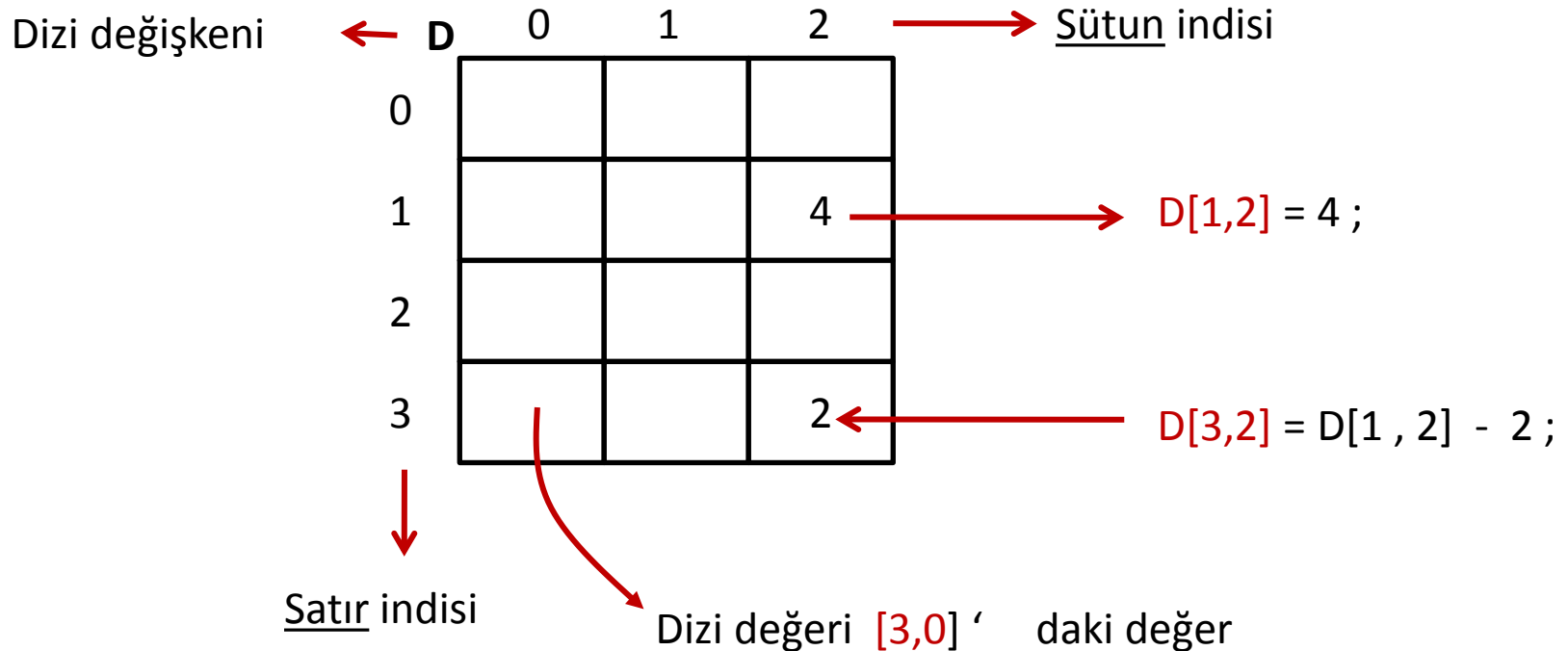
İki Boyutlu Dolap

	0	1	2
0			
1			
2			

10- DİZİLER (ARRAYS)

10.2 - İki boyutlu diziler . Matris de denir. Örn. *Satır x Sütun* (4x4) matris ;

D [satır, sütun]



10- DİZİLER (ARRAYS)

Bir boyutlu dizilerin tanımlanması :

`int [,] D = new int [satır sayısı , sütun sayısı] ;`

Dizideki elamanların tipi

Dizi değişkeni

Dizinin Satır ve sütun sayısı

```
int [ , ] D = new int [ 3 , 4 ] ; //3 satır 4 sütun
```

D	0	1	2	3
0				
1				
2				

10- DİZİLER (ARRAYS)

İki boyutlu dizilerin tanımı ;

```
int [ , ] d = new int [2,3] {  
                                { 2, 5, 6 },  
                                { 8, 7, 4 }  
                                };
```

	0	1	2
0	2	5	6
1	8	7	4

```
string [ , ] d = {  
    { "A+", "Pekiyi" },  
    { "A", "İyi" },  
    { "A-", "Orta" },  
    { "F", "Zayıf" }  
};
```

	0	1
0	A+	Pekiyi
1	A	İyi
2	A-	Orta
3	F	Zayıf

10- DİZİLER (ARRAYS)

Örn : **2X3** lük matrise başlangıçta aşağıdaki gibi değerler yerleştirilmiştir. Daha sonra dizinin bazı elemanları toplanmıştır.

D	0	1	2
0	2	5	6
1	8	7	4

```
int [,] D = new int [ 2 , 3 ] {  
    { 2, 5, 6 },  
    { 8, 7, 4 }  
};  
  
int y = D [ 0,1 ] + D [ 1,2 ] + D [ 0,0 ] + D [ 1,1 ] ;  
//      5      +      4      +      2      +      7  
  
Console.WriteLine ( "Toplam : " + y );
```

10- DİZİLER (ARRAYS)

Örn : **4x4** lük matrise 1 ile 10 arasında rastgele sayılar yerleştirildikten sonra birinci ve ikinci köşegendeki sayılar toplamını bulalım.

D	0	1	2	3
0				
1				
2				
3				

2. köşegen

$D[0,3]$, $D[1,2]$, $D[2,1]$, $D[3,0]$

1. köşegen

$D[0,0]$, $D[1,1]$, $D[2,2]$, $D[3,3]$

10- DİZİLER (ARRAYS)

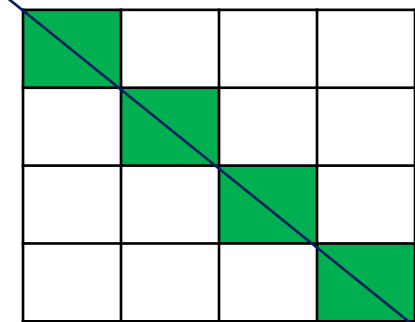
Örn : 4x4 lük matristeki (1. KÖŞEĞEN TOPLAMI)

```
int [,] D = new int [4, 4];
Random R= new Random();
for ( int sat=0 ; sat < D.GetLength(0); sat++ )
{
    for ( int sut=0 ; sut < D.GetLength(1); sut++ )
    {
        D[ sat, sut] = R.Next(1,10);
    }
}

int kosegen = 0;
for ( int sat=0 ; sat < D.GetLength(0); sat ++ )
    kosegen += D [sat, sat];

Console.Write ("1. Köşegen Toplamı:" + kosegen );
```

D	0	1	2	3
0				
1				
2				
3				



1. köşegen

D [0,0] , D [1,1] , D [2,2] , D [3,3]

10- DİZİLER (ARRAYS)

Örn : 4x4 lük matristeki (2. KÖŞEĞEN TOPLAMI)

```
int [,] D = new int [4, 4];
Random R = new Random();
for ( int sat=0 ; sat < D.GetLength(0); sat++ )
{
    for ( int sut=0 ; sut < D.GetLength(1) ; sut++ )
    {
        D[ sat, sut] = R.Next (1,10);
    }
}

int kosegen2 = 0;
int x= D.GetLength (1)-1 ; // sütun indexi için
for ( int sat=0 ; sat < D.GetLength(0); sat ++ )
    kosegen2 += D [sat, x-sat];

Console.Write ("2. Köşegen Toplamı:" + kosegen2 );
```

D	0	1	2	3
0				
1				
2				
3				

2. köşegen

D [0,3] , D [1,2] , D [2,1] , D [3,0]

10- DİZİLER (ARRAYS)

Örn : 4X3 lük bir matrise rastgele tamsayılar yerleştirildikten sonra , matrisi aşağıdaki gibi yazdırınız.

```
3 4 1
9 5 2
7 3 8
1 9 4
```

	0	1	2
0			
1			
2			
3			

10- DİZİLER (ARRAYS)

4X3 lük bir matrise rastgele tamsayılar yerleştirildikten sonra matris gibi yazdırınız.

Random r = new Random();

	0	1	2
0			
1			
2			
3			

10- DİZİLER (ARRAYS)

4X3 lük bir matrise rastgele tamsayılar yerleştirildikten sonra matris gibi yazdırınız.

```
Random r = new Random();  
int [ , ] d= new int [4,3];
```

	0	1	2
0			
1			
2			
3			

10- DİZİLER (ARRAYS)

4X3 lük bir matrise rastgele tamsayılar yerleştirildikten sonra matris gibi yazdırınız.

```
Random r = new Random();  
int [ , ] d= new int [4,3];  
for ( int sat=0 ; sat < d.GetLength(0); sat ++ )  
{  
  
}
```

	0	1	2
0			
1			
2			
3			

10- DİZİLER (ARRAYS)

4X3 lük bir matrise rastgele tamsayılar yerleştirildikten sonra matris gibi yazdırınız.

```
Random r = new Random();  
int [ , ] d= new int [4,3];  
for ( int sat=0 ; sat < d.GetLength(0); sat ++ )  
{  
    for ( int sut=0; sut < d.GetLength(1) ; sut ++)  
    {  
        d [sat,sut] = r.next (1,10) ;  
    }  
}
```

	0	1	2
0	3	4	1
1	9	5	2
2	7	3	8
3	1	9	4

10- DİZİLER (ARRAYS)

4X3 lük bir matrise rastgele tamsayılar yerleştirildikten sonra matris gibi yazdırınız.

```
Random r = new Random();
int [ , ] d= new int [4,3];
for ( int sat=0 ; sat < d.GetLength(0); sat ++ )
{
    for ( int sut=0; sut < d.GetLength(1) ; sut ++ )
    {
        d [sat,sut] = r.next (1,10) ;
    }
}
for ( int sat=0 ; sat < d.GetLength(0); sat ++ )
{
}
```

	0	1	2
0	3	4	1
1	9	5	2
2	7	3	8
3	1	9	4

10- DİZİLER (ARRAYS)

4X3 lük bir matrise rastgele tamsayılar yerleştirildikten sonra matris gibi yazdırınız.

```
Random r = new Random();
int [ , ] d= new int [4,3];
for ( int sat=0 ; sat < d.GetLength(0); sat ++ )
{
    for ( int sut=0; sut < d.GetLength(1) ; sut ++ )
    {
        d [sat,sut] = r.next (1,10) ;
    }
}
for ( int sat=0 ; sat < d.GetLength(0); sat ++ )
{
    for ( int sut=0; sut < d.GetLength(1) ; sut ++ )
    {
        Console.Write ( " { 0} " , d[sat, sut] );
    }
}
```

	0	1	2
0	3	4	1
1	9	5	2
2	7	3	8
3	1	9	4

10- DİZİLER (ARRAYS)

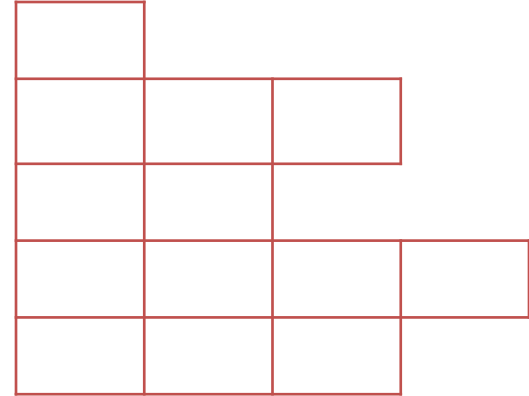
4X3 lük bir matrise rastgele tamsayılar yerleştirildikten sonra matris gibi yazdırınız.

```
Random r = new Random();
int [ , ] d= new int [4,3];
for ( int sat=0 ; sat < d.GetLength(0); sat ++ )
{
    for ( int sut=0; sut < d.GetLength(1) ; sut ++ )
    {
        d [sat,sut] = r.next (1,10) ;
    }
}
for ( int sat=0 ; sat < d.GetLength(0); sat ++ )
{
    for ( int sut=0; sut < d.GetLength(1) ; sut ++ )
    {
        Console.Write ( " { 0} " , d[sat, sut] );
    }
    Console.WriteLine("\n");
}
```

	0	1	2
0	3	4	1
1	9	5	2
2	7	3	8
3	1	9	4

```
3  4  1
9  5  2
7  3  8
1  9  4
```

JAGGED (girintili) DİZİLER



10- DİZİLER (ARRAYS)

10. 3- Jagged Diziler :

Bu tip diziler iki boyutlu dizilere benzerler ancak *her satırdaki sütun sayısı farklı* olabilmektedir.

X	0	1	2	3	4	5
0						
1						
2						
3						
4						

10- DİZİLER (ARRAYS)

10. 3- Jagged Diziler : Bu tip diziler iki boyutlu dizilere benzerler ancak her satırdaki sütun sayısı farklı olabilmektedir.

Yukarıdaki şekil şu şekilde tanımlanır :

```
int [][] x = new int [3] [ ] ;  
x [0] = new int [4] ;  
x [1] = new int [2] ;  
x [2] = new int [3] ;
```

X	0	1	2	3
0				
1				
2				

10- DİZİLER (ARRAYS)

10. 3- Jagged Diziler : Bu tip diziler iki boyutlu dizilere benzerler ancak her satırdaki sütun sayısı farklı olabilmektedir.

Yukarıdaki şekil şu şekilde tanımlanır :

```
int [][] x = new int [3] [ ] ;  
x [0] = new int [ ] {6 , 58} ;  
x [1] = new int [ ] {34 , 45 , 8 } ;  
x [2] = new int [ ] {4} ;
```

X	0	1	2
0	6	58	
1	34	45	8
2	4		

10- DİZİLER (ARRAYS)

Jagged Diziler

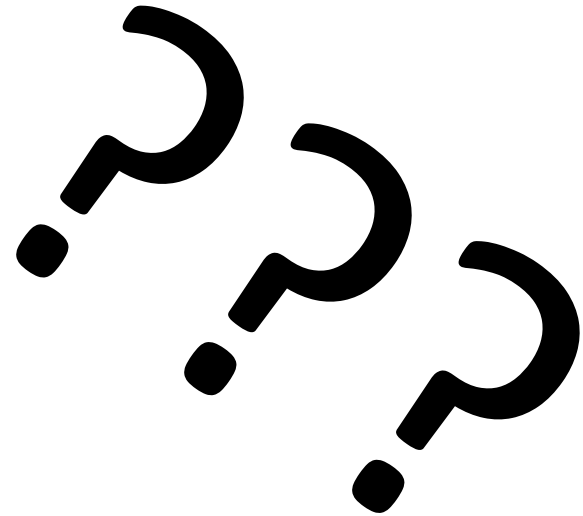
```
int i , k , t = 0 ;  
int [][] x = new int [3] [] ;  
x [0] = new int [] {6 , 8} ;  
x [1] = new int [] {3 , 5, 7 } ;  
x [2] = new int [] {4, 3} ;  
  
for (i = 0 ; i < x.Length ; i ++ )  
{  
    for (k = 0 ; k < x[i].Length ; k ++)  
    {  
        Console.WriteLine( x[i][k]);  
    }  
    Console.WriteLine();  
}
```

10- DİZİLER (ARRAYS)

Örnek : LOTO 6-49 tahmini yapacak programı yazınız .

```
static void Main(string[] args)
{
    int[] x = new int[6];
    Random r = new Random();
    for (int i = 0; i < x.Length; i++)
    {
        int sayi=r.Next(1,50);
        if (Array.IndexOf (x, sayi) == -1) x[i] = sayi; // alttaki if ile aynı işi yapar
        //if (!x.Contains(sayi)) x[i] = sayi; // yukardaki if ile aynı işi yapar
    }
    for (int i = 0; i <x.Length; i++) Console.Write("{0,3}", x[i]);
    Console.ReadKey();
}
```

10- DİZİLER (ARRAYS)



ÇEŞİTLİ ÖRNEKLER

10- DİZİLER (ARRAYS)

Örnek : loto 6-49 tahmini yapacak programı yazınız .

```
int[ ] m = new int[6];
Random r = new Random();
bool varmi = false;
for (int k = 0; k < m.Length ;    )
{
    varmi = false;
    int sayi = r.Next(1, 50); // [1-49] arası rastgele sayı üretir
    for (int i = 0; i < m.Length ; i++)
    {
        if ( sayi == m[ i ] ) { varmi = true; break; }
    }
    if ( !varmi )
    {
        m[ k ] = sayi;
        k++;
    }
}
foreach (int x in m) Console.Write(" {0} ", x );
```

10- DİZİLER (ARRAYS)

```
using System;
using System.Collections; // ekleyiniz

namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            int[ ] d = { 4, 3, 8, 0, 5 };
            foreach(int x in d ) Console.Write (« » + x );

            Array.Sort(d); // sırala
            Console.WriteLine(«Sıralandıktan sonraki hali»);

            foreach(int x in d ) Console.Write(« »+x );

            Console.ReadLine();
        }
    }
}
```

10- DİZİLER (ARRAYS)

Örnek : loto 6-49 tahmini yapacak programı yazınız .

```
static void Main(string[] args)
{
    int [] D = new int [6];           // 6 adet tamsayı tutulacak
    Random r = new Random();
    for (int k = 0 ; k < D.Length ;    )
    {
        int sayi = r.Next(1, 50);      // 1 ile 49 arası rastgele tamsayı
        if ( Array.IndexOf ( D , sayi ) == -1 ) // D dizisinde sayi değerin index' inin ver yoksa -1 dir
        {
            D[ k ] = sayi;
            k++; ;
        }
    }
    foreach (int x in D) Console.Write(" {0} ", x ); // Dizideki her bir elemanı ekrana yazdır
}
```

10- DİZİLER (ARRAYS)

8.1. Bir boyutlu diziler ile ilgi örnek;

```
Random r = new Random();  
int [ ] m = new int [10];  
  
for ( int k = 0 ; k<10 ; k++ ) m [ k] = r.Next(10,100) ;  
for ( int k = 0 ; k<10 ; k++ ) Console.Write(" " + m [ k] ) ;  
Console.WriteLine( );  
foreach ( int x in m) Console.Write(" " + x ) ;
```


10- DİZİLER (ARRAYS)

Örnek : 10 elemanlı bir tamsayı dizisine 1 ile 100 arasında rastgele sayılar yerleştirildikten sonra bu sayılardan en büyüğünü bulup yazdıran programı yazınız .

```
int [ ] m = new int [10];
Random r = new Random();
for ( int k = 0 ; k<10 ; k++ )    m [ k] = r.Next(1,100) ;

int  eb= m[0];
for ( int k = 1 ; k<10 ; k++ )
{
    if ( d[k] > eb )  eb = d [k];
}
Console.Write(“ En büyük değer : {0} ” , eb ) ;
```

10- DİZİLER (ARRAYS)

Örnek: 1 ile 7 arası rastgele üretilmiş 20 adet sayı , her satırda 5 eleman olacak şekilde ekrandan yazdıran program ı yazınız.

```
Random r= new Random();
string s = "";
for ( int c = 1; c <= 20; c++ )
{
    int n = r.Next( 1, 7 ); // 1-6 arasında
    s += n + "    ";
    if ( c % 5 == 0 )
        s+= "\n";
}
Console.Write(s);
```

Örnek ekran çıktısı :

3	2	3	5	2
1	5	6	1	3
2	4	2	4	4
4	6	2	6	6

10- DİZİLER (ARRAYS)

Örnek : Klavyeden girilen string bilginin harf frekansını bulup ekrandan yazdıran programı yazınız ?

```
static void Main(string[] args)
{
    int[] d = new int[26];
    string s = Console.ReadLine();
    for (int i = 0; i < s.Length; i++)
    {
        int pos = (Char) s[i]; // ASCII sırası
        if (pos > 96) d[pos - 97]++; // a - z ise
        else d[pos - 65]++; // A-Z ise
    }

    for (int i = 0; i < d.GetLength(0); i++)
    {
        Console.Write("{0} -> {1}\n", (Char) (i+65), d[i] );
    }
}
```

ÖRNEĞİN :

aBCAcZSDFz

A->2

B->1

C->1

....

Z->2

10- DİZİLER (ARRAYS)

Soru -1 : Permütasyon dizi; dizi içerisine rastgele dizi indisini yerleştirme. Dizi içinde dizi indisleri rastgele yerleştirilecek. Aynı değer tekrarlanmayacak .

0	1	2	3	4
3	4	0	1	2

Soru -2 : Rastgele üretilmiş 10 sayıdan asal olanlarını bulup yazdıracak programı yazınız.

10- DİZİLER (ARRAYS)

Cevap-1 – Permütasyon dizi (indisi içerisinde rastgele yerleşmiş dizi) .

```
static void Main(string[] args)
{
    int[] x = new int[9];
    Random r = new Random();
    for (int i = 0; i < x.Length; )
    {
        bool yok = true;
        int sayi = r.Next(1,10) ;
        for (int k = 0; k < x.Length; k++)
        {
            if (sayi == x[k]) { yok = false; break; }
        }
        if (yok)
        {
            x[i] = sayi;
            i++;
        }
    }
    foreach (int i in x)
        Console.Write(i);
}
```

```
using System.Collections; // ekleyiniz en başa

static void Main(string[] args)
{
    int[] x = new int[9];
    Random r = new Random();
    for (int i = 0; i < x.Length; )
    {
        int sayi= r.Next(1,10);
        if (Array.IndexOf(x, sayi) == -1)
        {
            x[i] = sayi;
            i++;
        }
    }

    foreach (int i in x)
        Console.Write(i);
}
```

10- DİZİLER (ARRAYS)

Cevap-2 Asal Sayı .

```
static void Main ( string [] args )
{
    int x, y, say =0;
    bool b;
    string s=" 1";
    int n = Convert.ToInt32( Console.ReadLine() );
    for (x=2 ; x<=n ; x++)
    {
        b = true;
        for (y=2; y<x; y++)
            if ( (x%y)==0 ) b=false;
        if( b )
        {
            say++;
            if ((say%6)==0) s += " \n ";
            s+=" \t" + x.ToString( ) + " \t" ;
        }
    }
    Console.Write ( s );
}
```

6- SÖZDE KOD (PSEUDE CODE)

Örn: Klavyeden girilecek N sayısının **fibonacci** serisini hesaplayıp yazdıran sözde kodu yazalım.



```
static void Main(string[] args)
{
    const int ADIMSAYISI = 9;
    int[] dizi = new int[ADIMSAYISI];
    dizi[0] = 0;
    dizi[1] = 1;
    for(int i = 2; i < ADIMSAYISI; i++)
    {
        dizi[i] = dizi[i - 1] + dizi[i - 2];
        Console.WriteLine( (i-1) + ". adım: " + dizi[i]);
    }
}
```

```
static void Main(string[] args)
{
    const int N = 9 ;
    int A=0, B=1 , C = 0;
    //Console.Write ( B);
    for ( int x = 1 ; x<= N ; x++ )
    {
        C = A + B;
        Console.Write ( C );
        A= B;
        B=C;
    }
}
```

N=9 için ekrana ;

1 2 5 3 8 13 21 yazar

0 + 1 = 1
1 + 1 = 2
1 + 2 = 3
2 + 3 = 5
3 + 5 = 8
5 + 8 = 13
8 + 13 = 21
....

10- DİZİLER (ARRAYS)

Örnek: Yandakine benzer bir çıktı verecek bir fonksiyon yazınız ?

Elaman	Değeri	Şekli
0	19	*****
1	3	***
2	15	*****
3	7	*****
4	11	*****
5	9	*****
6	13	*****
7	5	*****
8	17	*****
9	1	*_

```
static void yazdir()
{
    int[] n = { 19, 3, 15, 7, 11, 9, 13, 5, 17, 1 };
    string s = "";
    s += "Elaman\tDeğeri\tŞekli\n";
    for ( int i = 0; i < n.Length; i++ )
    {
        s += "\n" + i + "\t" + n[ i ] + "\t";
        for ( int j = 1; j <= n[ i ]; j++ ) s += "*";
    }
    Console.Write(s);
}
```


10- DİZİLER (ARRAYS)

Jagged Diziler

```
int i,k;
Random r = new Random();
int [][] x = new int [4] [] ;
string s ="";
x [0] = new int [6] ; // 0. satır 6 sütun
x [1] = new int [3] ; // 1. satır 3 sütun
x [2] = new int [2] ; // 2. satır 2 sütun
x [3] = new int [5] ; // 3. satır 5 sütun
for (i = 0 ; i < x.Length ; i ++ )
{
    for (k = 0 ; k < x[i].Length ; k ++ )
    {
        x[i][k] =r.Next()%10 ;
        s += x[i][k] + " \t ";
    }
    s += "\n";
}
Console.Write( s );
```

10- DİZİLER (ARRAYS)

Örnek : iki boyutlu dizilerde belirlenen sütuna göre sıralama.

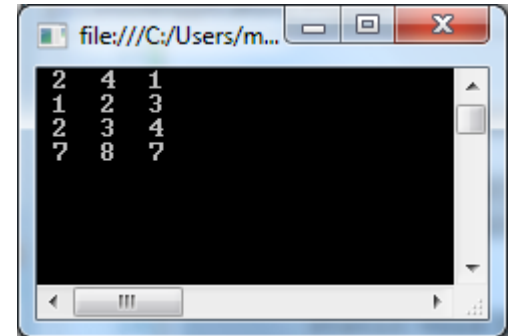
```
private static void Sort<T>(T[][] data, int col)
{
    Comparer<T> comparer = Comparer<T>.Default;
    Array.Sort<T[]>(data, (x, y) => comparer.Compare(x[col], y[col])); // ascending
    // Array.Sort<T[]>(data, (y, x) => comparer.Compare(x[col], y[col])); // descending
}

static void Main(string[] args)
{
    int[][] data = new int[][]
    {
        new int[] { 1, 2, 3 },
        new int[] { 2, 3, 4 },
        new int[] { 2, 4, 1 },
        new int[] { 7, 8, 7 }
    };

    Sort<int>(data, 2); // SIRALA

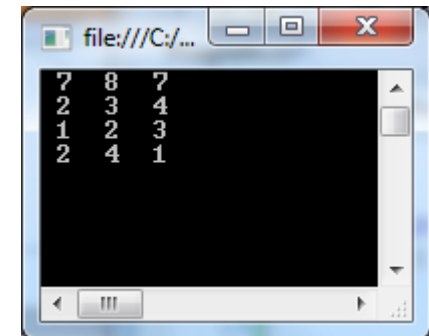
    for (int i = 0; i < data.Length; i++)
    {
        for (int j = 0; j < data[i].Length; j++) Console.Write(" {0} ", data[i][j] );
        Console.WriteLine();
    }
}
```

2. sütun indeksine göre **ascending**



2	4	1
1	2	3
2	3	4
7	8	7

2. sütun indeksine göre **descending**



7	8	7
2	3	4
1	2	3
2	4	1

10- DİZİLER (ARRAYS)

Örnek : iki boyutlu dizilerde belirlenen sütuna göre sıralama.

```
private static void ArrayToJagged( int [,] d, ref int [][] JG )
{
    int x = d.GetLength(0); //sıra
    int y = d.GetLength(1); //sütun
    for (int i = 0; i < d.GetLength(0); i++)
    {
        JG[i] = new int[y];
        for (int j = 0; j < d.GetLength(1); j++)
        {
            JG[i][j] = d[i, j];
        }
    }
}

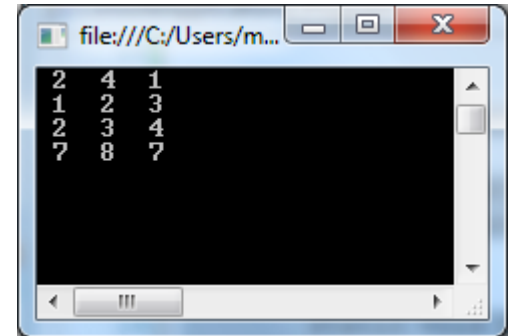
static void Main(string[] args)
{
    int[,] A = { { 1, 2, 3 }, { 2, 3, 4 }, { 2, 4, 1 }, { 7, 8, 7 } };

    int [][] B = new int[A.GetLength(0)] [];
    ArrayToJagged(A, ref B);

    Sort<int>(B, 2); // SIRALA

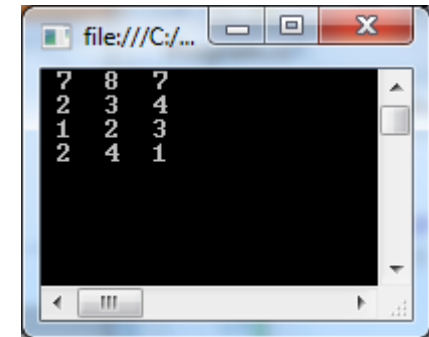
    for (int i = 0; i < B.Length; i++)
    {
        for (int j = 0; j < B[i].Length; j++) Console.Write(" {0} ", B[i][j] );
        Console.WriteLine();
    }
}
```

2. sütun indeksine göre **ascending**



2	4	1
1	2	3
2	3	4
7	8	7

2. sütun indeksine göre **descending**



7	8	7
2	3	4
1	2	3
2	4	1

10- DİZİLER (ARRAYS)

Örnek : iki boyutlu dizilerde belirlenen sütuna göre sıralama.

```
class Program
{
    // jagged dizi sıralama *****
    private static void Sort<T>( T [][] data, int col )
    {
        Comparer<T> comparer = Comparer<T>.Default;

        //Array.Sort<T[]>(data, (x, y) => comparer.Compare(x[col], y[col])); // Küçükten büyüğe
        Array.Sort<T[]>(data, (y, x) => comparer.Compare(x[col], y[col])); // Büyükten küçüğe
    }

    //iki boyutlu diziden jagged dönüşüm *****
    private static void ArrayToJagged( int [,] d, ref int [][] JG )
    {
        int x = d.GetLength(0); //sıra
        int y = d.GetLength(1); //sütun
        //int [][] JG= new int[x][];

        for (int i = 0; i < d.GetLength(0); i++)
        {
            JG[i] = new int[y];
            for (int j = 0; j < d.GetLength(1); j++)
            {
                JG[i][j] = d[i, j];
            }
        }
    }
}
```

```
//Main metodu *****
static void Main(string[] args)
{
    int[][] data = new int[][]
    {
        new int[] { 1, 2, 3 },
        new int[] { 2, 3, 4 },
        new int[] { 2, 4, 1 },
        new int[] { 7, 8, 7 }
    };

    Sort<int>(data, 1);

    for (int i = 0; i < data.Length; i++)
    {
        for (int j = 0; j < data[i].Length; j++)
        {
            Console.Write(" {0} ", data[i][j]);
        }
        Console.WriteLine();
    }
    Console.WriteLine();

    //*****
    int[,] A = { { 1, 2, 3 }, { 2, 3, 4 }, { 2, 4, 1 }, { 7, 8, 7 } };

    int[][] B = new int[A.GetLength(0)][];
    ArrayToJagged(A, ref B);
    Sort<int>(B, 2);

    for (int i = 0; i < B.Length; i++)
    {
        for (int j = 0; j < B[i].Length; j++)
        {
            Console.Write(" {0} ", B[i][j]);
        }
        Console.WriteLine();
    }

    Console.ReadKey();
}
```

10- DİZİLER (ARRAYS)

Soru: Matriste A,B,C,D, illerinin birbirine göre uzaklıkları (Km) verilmiştir. Buna göre A noktasından çıkan bir yolcu bütün illere uğramak şartıyla tekrar A noktasına ulaşması için en kısa yol hangisidir ?

	A	B	C	D	E	F
A		30	69	87	34	67
B	30		75	98	59	45
C	69	75		45	77	34
D	87	98	45		60	56
E	34	59	77	60		67
F	67	45	34	56	67	

Örnek sonuç:

A-1, B-2, C-3, D-4, E-5, F-6
(problemin çözümü değildir)

