

## 8. Bölüm

### DİZİLER

Dizi aynı ad altında toplanmış ve aynı tipte olan değişkenlerin sıralı bir listesidir. C'de bir dizi için bildirim yapılırken kullanılan genel kural şudur.

```
tip dizi_ismi[eleman_sayısı];
```

Bütün diziler 0 ile başlar.

### ÖRNEK

```
# include <stdio.h>
main( )
{
int sayi_listesi[10]; /* 10 tamsayı içeren bir dizi */
sayi_listesi[4]=823;
sayi_listesi[0]=38;
printf("eleman %d : %d\n",4,sayi_listesi[4]);
printf("eleman %d : %d\n",0,sayi_listesi[0]);
}
```

### ÖRNEK

```
# include <stdio.h>
main()
{
int i, tpl;
int a[10], b[10], c[10];
for (i=0;i<=9;i++)
    a[i]=b[i]=c[i]=0;
tpl=0;
for (i=0;i<=9;i++)
{
    a[i]=i;
    b[i]=i*i;
    c[i]=2*a[i]+b[i];
    tpl=tpl+c[i];
}
printf ("          sayi    karesi  2*sayi+karesi \n");
for (i=0;i<=9;i++)
printf("%8d %8d %8d \n", a[i],b[i],c[i]);
printf("          toplam =%d \n",tpl);
}
```

## ÖRNEK

```
# include <stdio.h>
int i;
int kare[21];
{
for (i=1;i<=20;i++)
    kare[i]=i*i
}
```

## ÖRNEK

```
# include <stdio.h>
main( )
{
int i;
float sonuc[10];
printf("10 tane reel sayı girin\n");
for (i=0;i<10;i++);
{
printf("No %d:",i+1);
scanf("%f, &sonuç[i]);
}
printf("sayılar girildi\n");
printf("sonuç için 1 ile 10 arasında bir sayı\n");
printf("çıkış için 0 yazın\n");
do
{
printf(":"); scanf("%d", &i);
if (i>0 && i<11) /* geçerli bir eleman (1-10) */
printf("Eleman %d, %f içeriyor\n", i, sonuç[i-1]); /* elemanlar 0 ile 9 arasında */
}
while (i!=0)
}
```

## KARAKTER DİZGİLERİ

Bir sözcük veya tümceyi bir değişkende saklamak istediğimizde bunu yapmanın bir yolu tümceyi bir karakter dizisine yerleştirmektir.

Bir dizgi, bir dizi olduğu için = işareti kullanılamaz.

Bir dizgiyi bir karakter dizisinden diğerine taşımak için dizgi kopyalama fonksiyonu olan `strcpy( )`'yi kullanmak gerekir. Bu fonksiyonun iki parametresi vardır. Kaynak dizgisi ve hedef dizgisi.

```
char hedef[80];
char kaynak[80];
strcpy (hedef,kaynak);
```

#### ÖRNEK

```
# include <stdio.h>
main( )
{
char dizgi1[80], dizgi2[80];
printf("lütfen bir dizgi girin\n:");
scanf("%s", &dizgi1);
printf("bu dizgi2'ye aktarılıyor\n");
strcpy(dizgi2, dizgi1);
printf("Dizgi %s\n", dizgi2);
}
```

NOT :Bir dizi aynı ad altında bir araya gelmiş olan bir grup değişkenden oluşur. Dizinin her bir değerine elman adı verilir. Bir elemanı belirtmek için kullanılacak sayı pozitif bir tam sayı olmalıdır ve bu elemanın içerdiği değer herhangi bir tipte olabilir.

#### MATRİS İLE İLGİLİ ÖRNEK:

```
# define sira 3
# define kolon 3
# include <stdio.h>
main()
{
int i, j, k, katsayi;
int matris [sira][kolon];
for (i=0;i<sira;i++)
for (j=0;j<kolon;j++)
matris[i][j]=0;

for (i=0;i<sira;i++)
{
j=0;
while (j<kolon)
{
printf("matris[%d][%d]=",i,j);
scanf("%d",&katsayi);
if (katsayi<0)
break;
matris[i][j]=katsayi;
j++;
}
```

```

}
}
for (i=0;i<sira;i++)
{
for (j=0;j<kolon;j++)
printf("matris[%d][%d]=%d",i,j,matris[i][j]);
printf("\n");
}
}

```

## BREAK DEYİMİ

C de döngüden çıkmak için kullanılır. Bilgisayar döngünün sonunu beklemeden, istenilen koşulun sağlandığını kabul ederek, bir sonraki satırdaki komutları işletir.

for, while veya do..while döngülerinde kullanılır. if..else ile kullanılamaz.

Örnekte, katsayı sıfırdan küçük olursa, o sıradaki diğer elemanların hep sıfır değeri alması istendi. Bu durumda break ile bilgisayar j döngüsünden çıkıp bir sonraki i değerini alarak işleme devam eder.

ÖRNEK (CONTINUE deyimi) : C 'de continue "kaldığın yerden devam et", döngüyü tamamlayıncaya kadar devam et anlamındadır.

```

# include <stdio.h>
main()
{
int i, j;
int a[5];
for (i=0;i<5;i++)
a[i]=0;
for (i=0;i<5;i++)
scanf("%d",&a[i]);
for (i=0;i<5;i++)
{
if ( a[i]<0)
continue;
printf("%d TAMAM \n", a[j]);
} }

```

Bu örnekte, eğer a[j] değeri sıfırdan küçükse döngünün sonuna, yani { işaretine kadar gitmeden başa dön, j nin bir sonraki değerini alarak işleme devam et denmektedir.

program çalıştırıldığında; 11 -22 33 -44 55 değerleri girilirse  
11 TAMAM

33 TAMAM

55 TAMAM

çıktısı elde edilir.

ÖRNEK : Enter tuşuna basıncaya kadar verilen tüm karakterleri ASCII koduna çeviren program.

```
# include <stdio.h>
main()
{
    int i;
    char c;
    for (i=0;(c=getchar())!='\n';++i)
        printf("%d \n",c);
}
```

ÖRNEK ( Özel operatörler ve ifadeler)

```
# include <stdio.h>
main()
{
    int a, b, i;
    for (i=0;i<=10;i++)
        a=a+i;
    printf("toplam a nın değeri = %d \n",a);

    for (i=0;i<=10;i++)
        b+=i;
    printf("toplam b nin değeri = %d \n", b);
}
```

Bu örnekte a=55 ve b=55 değeri elde edilir.

Genel olarak  $x=x+y$  yerine  $x+=y$  yazılabilir. Yani,

$x=x \text{ op } y$  yerine  $x \text{ op } =y$  komutu yazılabilir.

Örneğin;

$x=x*(y+1)$  yerine  $x *= (y+1)$  yazılabilir.

ÖRNEK toplamını a, b, n 'in çeşitli değerleri için hesaplanacağını düşünüp bunun için bir fonksiyon yazalım.

```
# include <stdio.h>
main()
{
```

```

double seri();
int i;
for (i=1;i<=5;++i)
    printf("%d %1.6f %1.6f %1.6f \n", i, seri(1,i,0), seri(2,i,1));
}
double seri(a,n,b)
int a,n,b;
{
    double s, r;
    int j;
    r=0;
    s=0;
    for (j=1;j<=n;++i)
        {
            r=a*j-b;
            s=s+(1/r);
        }
    return(s);
}

```

ÖRNEK (Kısaltılmış if)

```

# include <stdio.h>
main()
{
    int i; float a, b, z;
    a=4; b=1.5; z=0;
    for (i=0; b<=a+100.0; i++)
    {
        a=(a+7.3)*i;
        b=(b+1.0)*(i+1.0);
        if (a>b)
            z=a;
        else
            z=b;
        printf("%12.7f \n", z); } }

```

NOT : C ' de mümkün olan diğer bir kısaltma da if ... else deyimlerinde ? işaretinin operatör olarak kullanılmasıdır. Yukarıdaki programın aynısı bu operatör kullanılarak aşağıda verilmiştir.

```

# include <stdio.h>
main()
{

```

```
int i; float a, b, z;  
a=4; b=1.5; z=0;  
for (i=0; b<=a+100.0; i++)  
{  
    a=(a+7.3)*i;  
    b=(b+1.0)*(i+1.0);  
    z=(a>b) ? a: b;  
    printf("%12.7f \n", z); } }
```

Not:  $z=(a>b) ? a: b;$  ifadesinde önce  $(a>b)$  işlem konur, eğer true (yani pozitif, doğru) ise 2.parametre olan a işlem görür, yani  $z=a$ ; daha sonra ise yani  $(a>b)$  değilse son parametre olan b işlem görüyor, yani  $z=b$ ; olur.