

Bölüm 5
Döngü Komutları

- while Komutu
- Diğer Operatörler
- Bileşik Komut
- for Komutu
- do-while Komutu
- İçiçe Döngüler
- break ve continue Komutları

while Komutu

Döngü komutları komutların bir çok kez yeniden yürülmesini sağlayan, programlamada kullandığımız önemli yapılardan biridir.

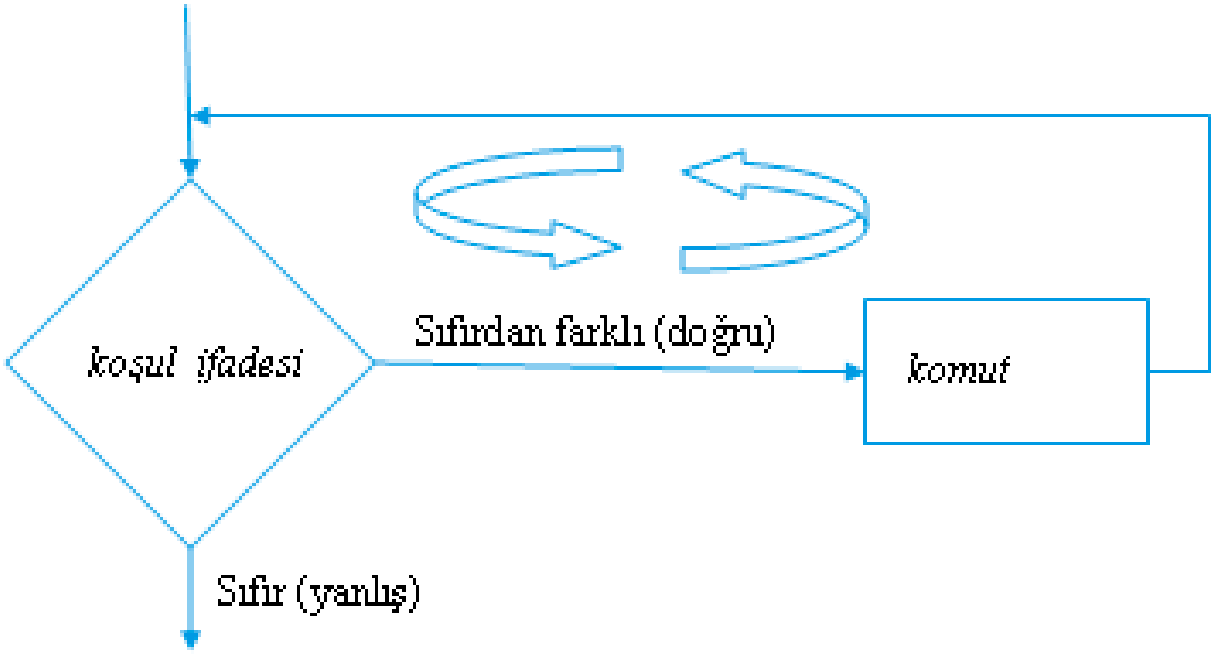
C dilinde bize bu özelliği sağlayan komutlar

- `while`
- `for`
- `do-while`

komutlarıdır.

while Komutu

```
while ( koşul_ifadesi )
    komut;
```



while döngüsünde *komut*'un yürütümü *koşul ifadesi* doğru olduğu sürece devam eder.

while Komutu

Örnek:

```
printf("Bir pozitif sayi giriniz:");
scanf("%d",&n);
while(n<0)
    scanf("%d",&n);
printf("En son n degeri: %d",n);
```

while Komutu

while Komutunun Aşamaları

```
sayac=1;                          → İlk değer atama aşaması
while(sayac<=10)                  → Kontrol aşaması
{
  printf("%3d",sayac);
  sayac=sayac+1;                  → Güncelleme aşaması
}
```

while Komutu

Örnek:

Pozitif sayıların girildiği ve girilen bu pozitif sayıların en büyüğünün bulunduğu programı yazınız. Programda sayı girme işlemi negatif bir sayı girilinceye kadar devam etmelidir.

```
#include <stdio.h>
int main(void)
{
  int sayi,max=0;
  /*En buyuk sayinin sifir oldugunu kabul edelim.*/
  printf("Sayi giriniz: ");
  scanf("%d",&sayi);
```

while Komutu

```
while(sayi>0)
  {
  /*Eger girilen sayi max da bulunan buyuk sayidan daha
    buyukse yeni girilen sayi max degiskenine atanir*/
     if(sayi>max)
        max=sayi;
     printf("Sayi giriniz: ");
     scanf("%d",&sayi);
  }
  printf("En buyuk pozitif sayi: %d",max);
  return(0);
}
```

Bileşik Atama Operatörü

değişken = değişken operatör ifade;
değişken operatör = ifade;

Atama Komutu	Bileşik Operatör ile Yazımı
`carp=carp*k;`	`carp *=k;`
`i=i+1;`	`i+=1`
`j=j-1;`	`j-=1`
`terim=terim/(n*(n-1));`	`terim/=n*(n-1);`
`h=h-4/st;`	`h-=4/st;`
`alfa=4*(k-5)/(k+1)*alfa`	`alfa*=4*(k-5)/(k+1)`

Diğer Operatörler

Artırma ve Azaltma Operatörleri (++ --)

```
sayac=sayac+1;
sayac++;  (son-ek)
++sayac;  (ön-ek)
```

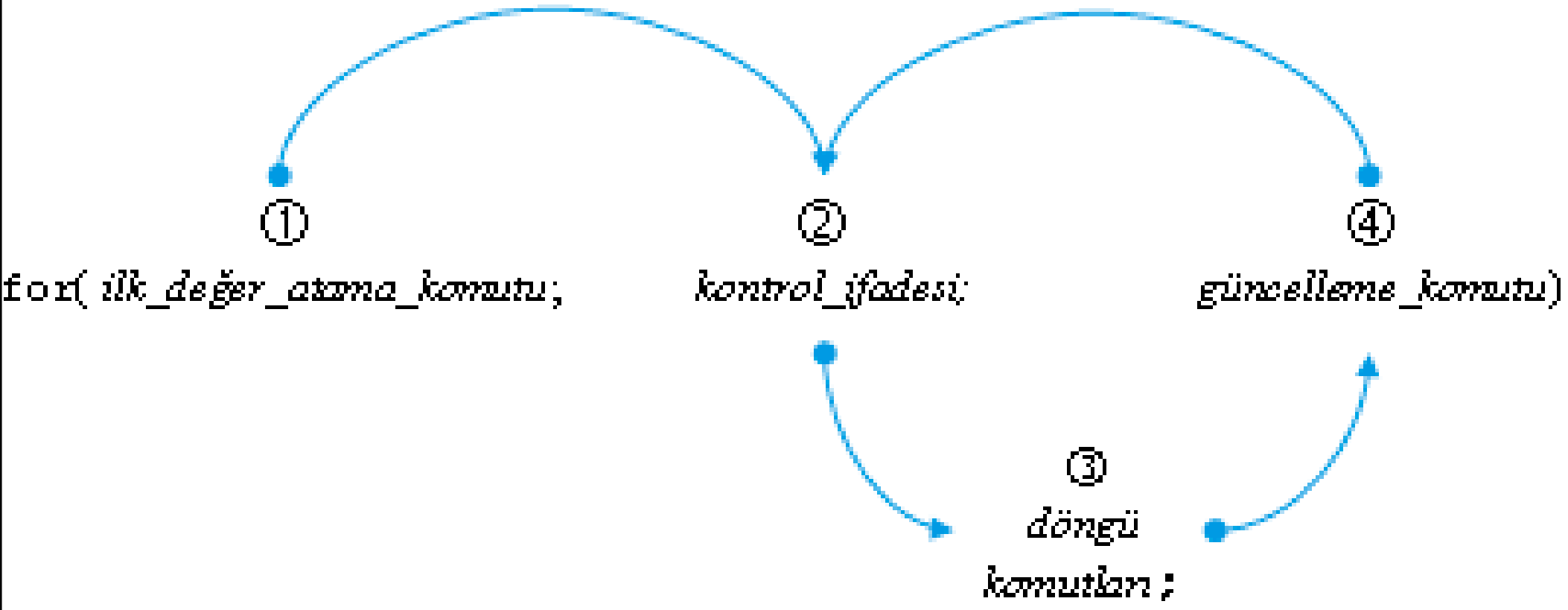
Kod	Eşdeğer Kod

`i=5;`
`j=++i;`
`printf("%d  %d",i,j);`
	`i=5;`
	`++i;`
	`j=i;`
	`printf("%d  %d",i,j);`

`i=5;`
`j=i++;`
`printf("%d  %d",i,j);`
	`i=5;`
	`j=i;`
	`++i;`
	`printf("%d  %d",i,j);`

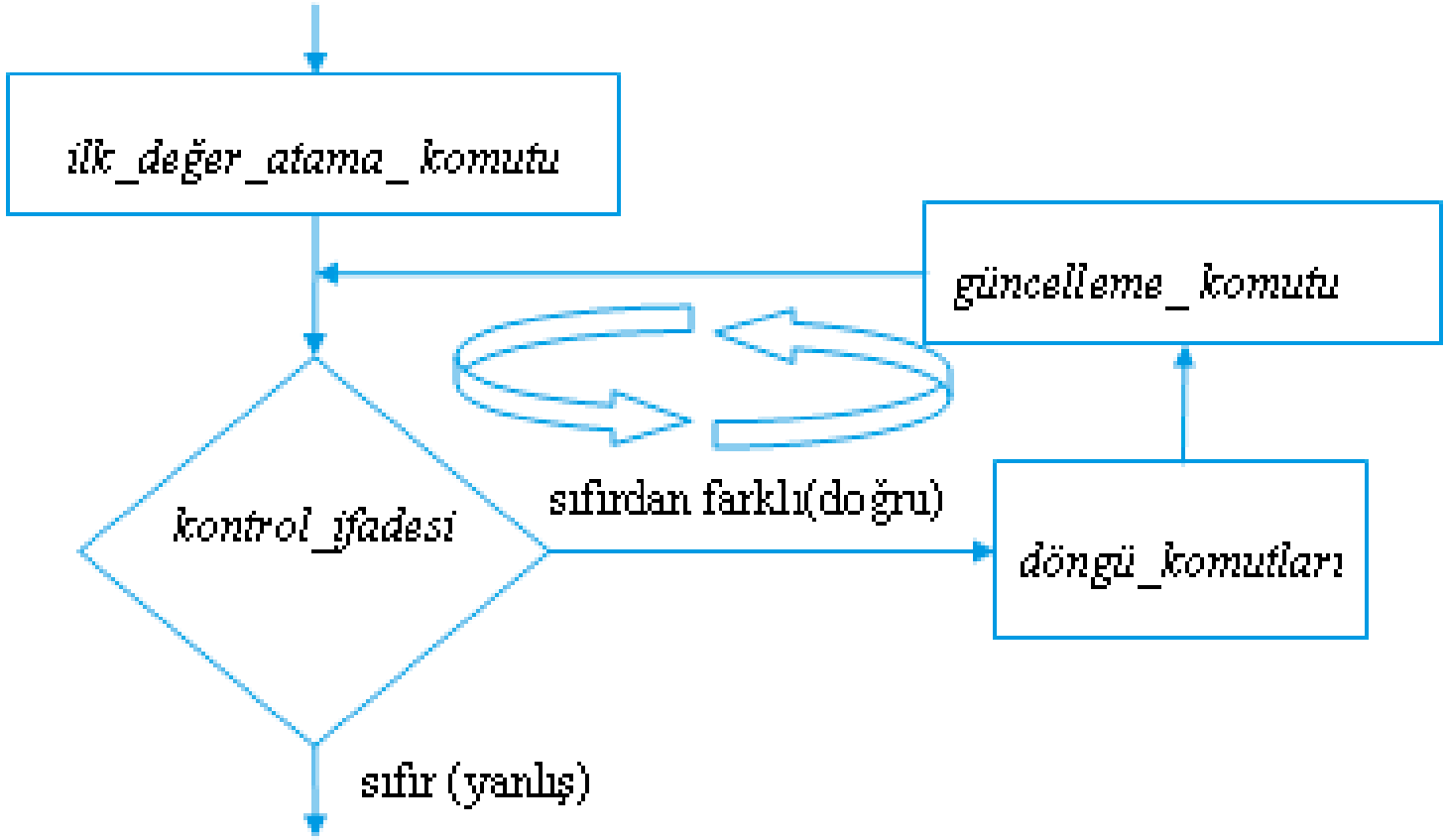
for Komutu

```
for( ilk_değer_atama_komutu;
     kontrol_ ifadesi;
     güncelleme_ komutu)
    döngü_komutları;
```



for Komutu

```
for( ilk_değer_atama_komutu;
     kontrol_ ifadesi;
     güncelleme_ komutu)
    döngü_komutları;
```



for Komutu

while döngüsü

```
sayac=1;
while(sayac<=10)
{
 printf(“%3d”,sayac);
   sayac++;
}
```

Eşdeğer for döngüsü

```
for(sayac=1; sayac<=10; sayac++)
     printf(“%3d”,sayac);
```

for Komutu

Örnek:

Pozitif bir n sayısını ve bu n sayısı kadar da tamsayıyı girdi olarak alan programı yazınız. Ayrıca, bu programda girilmiş olan n tane tamsayının kaçının negatif, pozitif ve sıfır olduğunu hesaplayıp ekranda gösteriniz.

```
#include <stdio.h>
int main(void)
{ int pos=0,neg=0,sifir=0,n,sayi,i;
  printf("Kac tane sayi gireceksiniz: ");
  scanf("%d",&n);
  /*Dongu icinde n tane sayinin girilmesi ve islenmesi*/
  for(i=1;i<=n; i=i+1)
  {
     printf("%d. sayi: ",i);
     scanf("%d",&sayi);
```

for Komutu

```
/*Girilen sayinin pozitif, negatif veya sifir olup
   olmadiginin belirlenmesi*/
     if(sayi>0)
        pos++;
     else if(sayi<0)
             neg++;
          else sifir++;
  }
/*Sonuclarin gosterilmesi*/
  printf("\n%d adet pozitif sayi.\n",pos);
  printf("%d adet negatif sayi.\n",neg);
  printf("%d adet sifir sayisi.\n",sifir);
  return(0);
}
```

for Komutunun Farklı Kullanımları

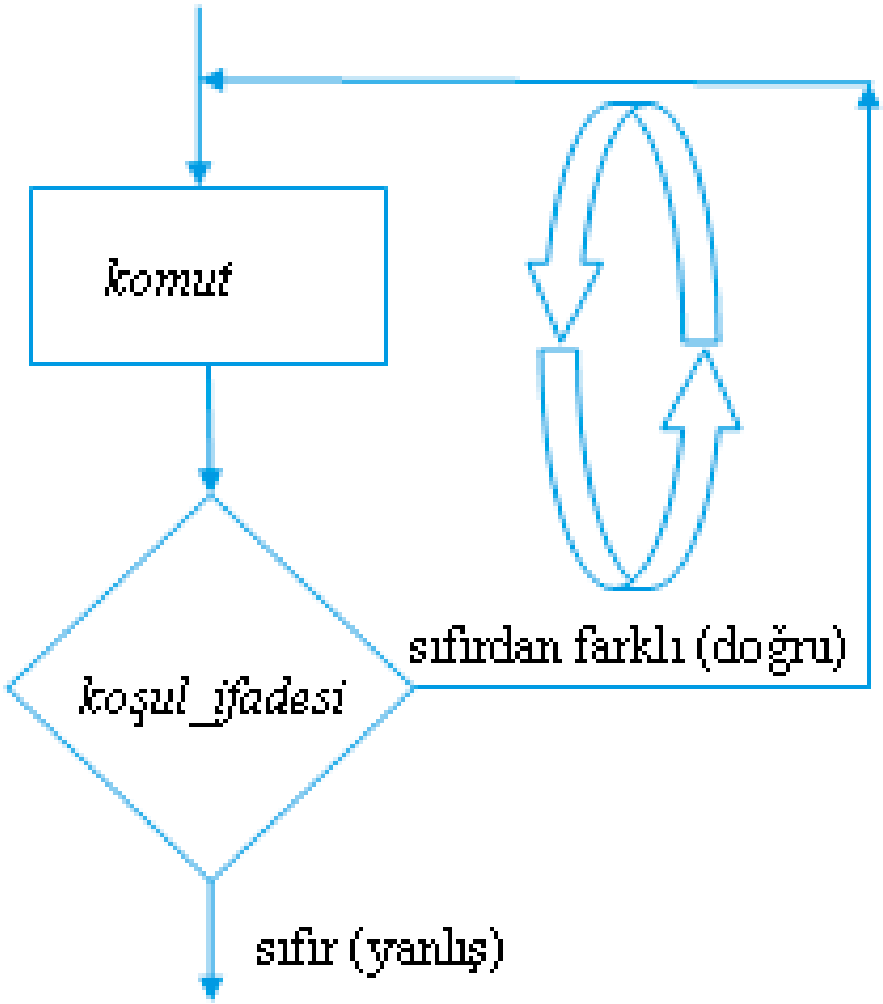
```
for(i=0,j=0; i<=4 && j>-8; i++, j--)
    printf("%3d %3d\n",i,j);
```

```
toplam=0;
i=1;
for(; i<=10; i++)
    toplam+=i;
```

```
toplam=0;
i=1;
for(; i<=10; )
{   toplam+=i;
    i++;
}
```

do while Komutu

```
do
   komut;
while (koşul_ifadesi );
```



do while Komutu

while ve **for** komutlarında koşul ifadesi, döngünün başında kontrol edilirken, **do-while** komutunda kontrol işlemi döngünün sonunda yapılır.

```
sayac=1;                          → İlk değer atama aşaması
do
{
   printf("%3d",sayac);
   ++sayac;                       → Güncelleme aşaması
}
while(sayac<=10);                 → Kontrol aşaması
```

do while Komutu

Örnek:

Aşağıdaki seçenekleri görüntüleyecek ve seçeneklere göre aşağıda tanımlanan seçeneğin alanını bulacak bir C programı yazınız.

1. Kare
2. Daire
3. Bitti

Seceneginizi belirtiniz (1, 2, 3):

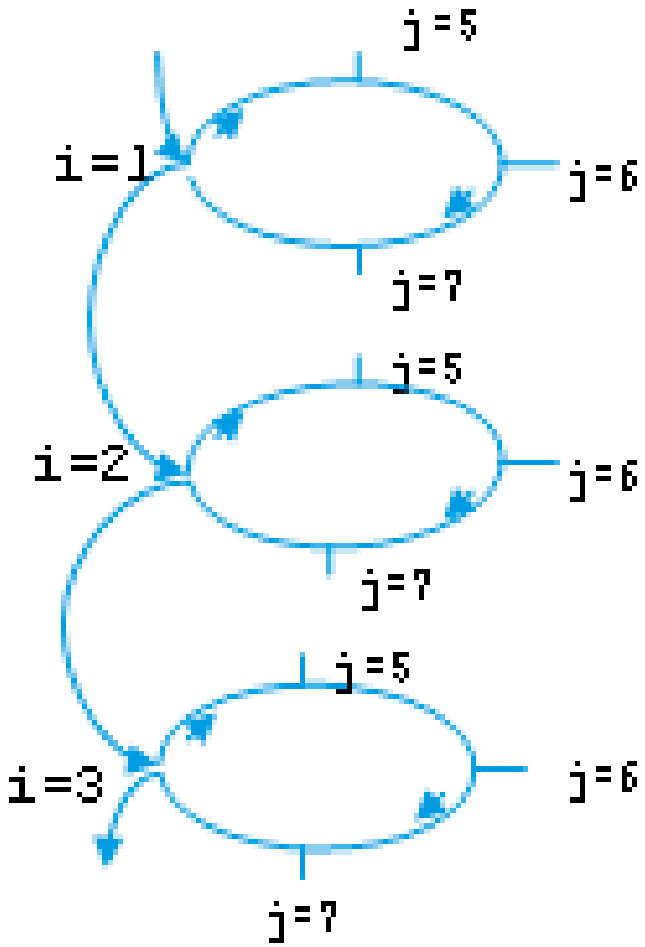
```
#include <stdio.h>
#define PI 3.141592654
int main(void)
{
        int cevap;
        double a,r,alan;
        do
        {   /*Menunun goruntulenmesi*/
              printf("\n     1. Kare");
        printf("\n     2. Daire");
        printf("\n     3. Bitir");
        printf("\n\nSeceneginizi belirtiniz: ");
        scanf("%d",&cevap);
```

do while Komutu

```
switch(cevap)
        {
           case 1: /*Karenin alaninin hesaplanmasi*/
                   printf("\nKarenin kenar uzunlugunu   giriniz:
");
                   scanf("%lf",&a);
                   alan=a*a;
                   printf("Karenin alani:%.2f\n",alan);
                   break;
           case 2: /* Dairenin alaninin hesaplanmasi*/
                   printf("\nDairenin yaricap uzunlugunu giriniz:
");
                   scanf("%lf",&r);
                   alan=PI*r*r;
                   printf("Dairenin alani:%.2f\n",alan);
        }
}
while(cevap!=3);
   return(0);
}
```

İçiçe Döngüler

```
for (i=1; i<=3; ++i)
      for (j=5;j<=7; ++j)
         printf("%3d%3d\n",i,j);
```



İçiçe Döngüler

Örnek:

Bir başka örnek olarak aşağıdaki çıktıyı veren program parçasını içiçe döngüler kullanarak yazalım.

```
1
2 1
3 2 1
4 3 2 1
5 4 3 2 1
6 5 4 3 2 1
7 6 5 4 3 2 1
8 7 6 5 4 3 2 1
```

```
for( i=1; i<=8; i++)
{   for(j=i; j>=1;j--)
       printf("%3d",j);
    printf("\n");
}
```

break ve continue Komutları

`break` komutu herhangi bir döngü komutu içinde kullanıldığında döngüden çıkılmasını sağlar.

```
for( i=1; i<=10;i++)
{
   printf("Sayi giriniz:");
   scanf("%d",&sayi);
   if(sayi<=0)
     break;
}
printf("%d pozitif sayi girildi.",i-1);
```

break ve continue Komutları

Döngünün herhangi bir aşamasında `continue` komutuna gelindiğinde, o turun yürütümü o noktada kesilir ve yürütüm bir sonraki tur ile devam eder.

```
toplam=0;
for( i=1; i<=5;i++)
{
   printf("Sayi giriniz:");
   scanf("%d",&sayi);
   if(sayi<=0)
     { i--;
       continue;}
   toplam+=sayi;
}
printf(“Toplam: %d”,toplam);
```