

Bölüm 5

Döngü Komutları

while Komutu

Diğer Operatörler

Bileşik Komut

for Komutu

do-while Komutu

İççe Döngüler

break ve continue Komutları

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

C dilinde bize bu özelliği sağlayan komutlar

➤ **while**

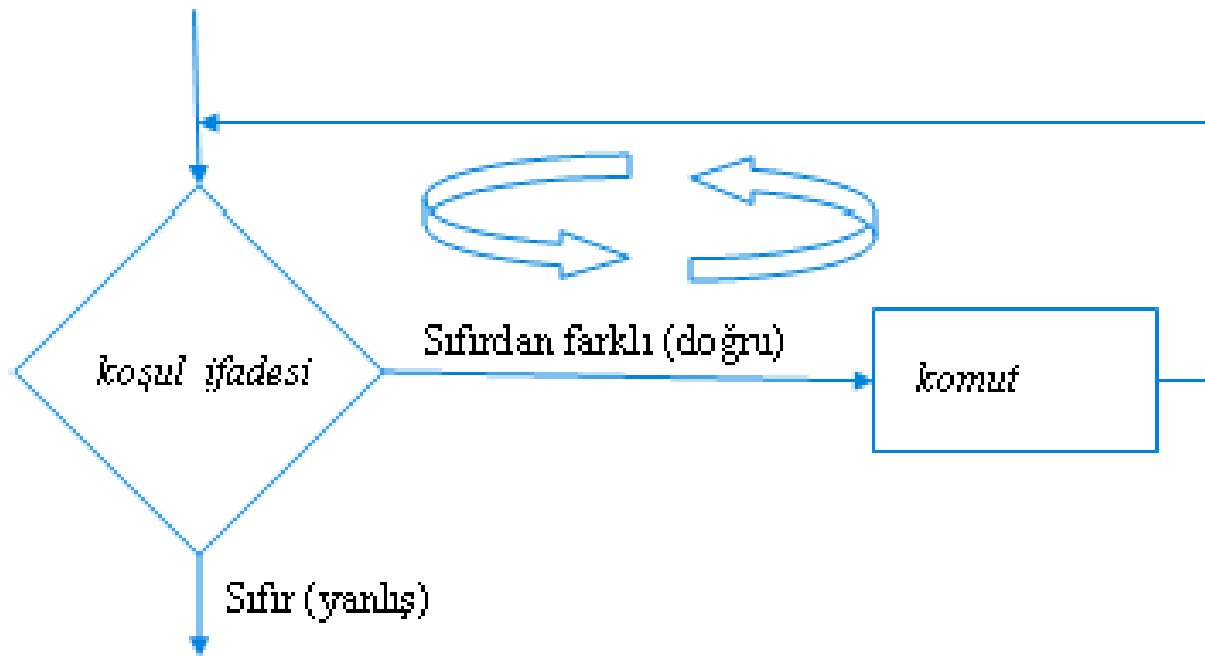
➤ **for**

➤ **do-while**

komutlarıdır.

while Komutu

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



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

Komutun yürütümü if deyimindeki gibi bir kez olmayıp, koşul ifadesi doğru olduğu sürece devam eder

Kullanıcının pozitif sayı girmesini sağlayan program

Örnek:

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

while Komutunun Aşamaları

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

→ İlkdeğer atama aşaması

→ Kontrol aşaması

→ Güncelleme aşaması

while doğru olduğu sürece gövdedeki komutlar yürütülür

İlk değer atama örnekleri

Örnek	Çıktı / Açıklama
<pre>sayac=6; while(sayac<=10) { printf("%3d", sayac); sayac=sayac+1; }</pre>	
<pre>sayac=10; while (sayac<=10) { printf("%3d",sayac); sayac=sayac+1; }</pre>	

İlk değer atama örnekleri

Örnek	Çıktı / Açıklama
<pre>sayac=6; while(sayac<=10) { printf("%3d", sayac); sayac=sayac+1; }</pre>	6 7 8 9 10
<pre>sayac=10; while (sayac<=10) { printf("%3d",sayac); sayac=sayac+1; }</pre>	10

İlk değer atama örnekleri

```
sayac=15;
while (sayac<=10)
{
    printf("%3d",sayac);
    sayac=sayac+1;
}
```

```
scanf("%d",&sayac);
while(sayac<=10)
{
    printf("%3d", sayac);
    sayac=sayac+1;
}
```

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


İlk değer atama örnekleri

<pre>sayac=15; while (sayac<=10) { printf("%3d",sayac); sayac=sayac+1; }</pre>	<p>sayac donguye girmeden once 15 degerini alır ve 'sayac<=10' ifadesi yanlış olduğundan while dongusune girilmez. Ciktida hicbirsey goruntulenmez.</p>
<pre>scanf("%d",&sayac); while(sayac<=10) { printf("%3d", sayac); sayac=sayac+1; }</pre>	<p>Bu program parcasinda ilk deger atama asamasi scanf'dir. Dolayisiyla, cikti klavyeden girilecek sayac degerine gore degisir</p>
<pre>while(sayac<=10) { printf("%3d", sayac); sayac=sayac+1; }</pre>	<p>sayac degiskenine atama yapilmadigindan bu degiskenin degeri belli degildir. Derleyiciye bagli olarak degisik ve beklenmeyen sonuclar elde edilebilir.</p>

Kontrol Aşaması örnekleri

Örnek	Çıktı / Açıklama
<pre>sayac=1; while (sayac<10) { printf("%3d",sayac); sayac=sayac+1; }</pre>	
<pre>sayac=1; while (sayac<=3) { printf("%3d",sayac); sayac=sayac+1; }</pre>	
<pre>sayac=1; while (sayac>=3) { printf("%3d",sayac); sayac=sayac+1; }</pre>	

Kontrol Aşaması örnekleri

Örnek	Çıktı / Açıklama
<pre>sayac=1; while (sayac<10) { printf("%3d",sayac); sayac=sayac+1; }</pre>	1 2 3 4 5 6 7 8 9
<pre>sayac=1; while (sayac<=3) { printf("%3d",sayac); sayac=sayac+1; }</pre>	1 2 3
<pre>sayac=1; while (sayac>=3) { printf("%3d",sayac); sayac=sayac+1; }</pre>	sayac degiskeni 1 degerini aldıktan sonra kontrol kisminde bulunan kosul ifadesi yanlis olacagindan while dongusune hic girilmeyecektir. Ekranda hicbir sey goruntulenmeyecektir.

Güncelleme Aşaması örnekleri

Örnek	Çıktı / Açıklama
<pre>sayac=1; while (sayac<=10) { printf("%3d",sayac); sayac=sayac+3; }</pre>	
<pre>sayac=1; while (sayac<=10) { printf("%3d",sayac); sayac=sayac-1; }</pre>	
<pre>sayac=1; while (sayac<=10) printf("%3d",sayac);</pre>	

Güncelleme Aşaması örnekleri

Örnek	Çıktı / Açıklama
<pre>sayac=1; while (sayac<=10) { printf("%3d",sayac); sayac=sayac+3; }</pre>	1 4 7 10
<pre>sayac=1; while (sayac<=10) { printf("%3d",sayac); sayac=sayac-1; }</pre>	sonsuz bir donguye girilir cunku sayac hep 10dan az olacaktır. 1 0 -1 -2 -3 -4 -5 -6 -7 -8 ...
<pre>sayac=1; while (sayac<=10) printf("%3d",sayac);</pre>	sonsuz bir donguye girilir cunku kosul ifadesi her zaman dogru olur

Döngü Değişkensiz while Komutları

Bazı while döngülerinde döngü değişkeni bulunmayabilir

```
while (350)

{
    scanf ("%d", &n) ;
    printf ("**%d**\n", n) ;
}
```

350 sayisi her zaman sifirdan farkli olduğu için sonsuz bir döngü oluşur

Döngü Değişkensiz while Komutları

Bazı while döngülerinde döngü değişkeni bulunmayabilir

```
while (-0.3)
{
    scanf ("%d", &n) ;
    if (n>0)
        printf ("**%d**\n", n) ;
    else
        return (0) ;
}
```

-0.3 sayısı her zaman sifirdan farklı olduğu için sonsuz bir döngü kurulmuştur ancak...

Döngü Değişkensiz while Komutları

Bazı while döngülerinde döngü değişkeni bulunmayabilir

```
while (0)

{
    scanf ("%d", &n) ;
    printf ("**%d**\n", n) ;
}
```

Döngüye hiç girilmeyecek !!!

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.

while Komutu

```
#include <stdio.h>
int main(void)
{
    int sayi,max=0;
    /*En büyük sayinin sifir oldugunu kabul edelim.*/
    printf("Sayi giriniz: ");
    scanf("%d",&sayi);
    while(sayi>0)
    {
        /*Eger girilen sayi max da bulunan büyük sayidan daha
        buyukse yeni girilen sayi max degiskenine atanir*/
        if(sayi>max)
            max=sayi;
        printf("Sayi giriniz: ");
        scanf("%d",&sayi);
    }
    printf("En büyük pozitif sayi: %d",max);
    return(0);
}
```

Bileşik Atama Operatörü

değişken = değişken operatör ifade;

değişken operatör = ifade;

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

Artırma ve Azaltma Operatörleri (++ --)

sayac=sayac+1;

sayac++; (son-ek)

++sayac; (ön-ek)

Kod	Eşdeğer Kod
<pre>i=5; j=++i; printf("%d %d",i,j);</pre> <p>6 6</p>	<pre>i=5; ++i; j=i; printf("%d %d",i,j);</pre>
<pre>i=5; j=i++; printf("%d %d",i,j);</pre> <p>6 5</p>	<pre>i=5; j=i; ++i; printf("%d %d",i,j);</pre>

Örnek

```
int i=10, k=2, bul;  
printf("%d", bul=+++i + 5 - k--);
```

bul=+++i + 5 - k--

bul=11 + 5 - k--

bul=16 - k--

bul=14

Program 1

Bir sınıfta 10 öğrencisi bulunan ve sınavı 100 puan üzerinden notlayan bir öğretmen, bu sınavın ortalamasını hesaplamak istemektedir. Bu sınavın ortalamasını hesaplayacak olan programı yazınız

Program 1-Çözüm

```
#include <stdio.h>
```

```
int main(void)
```

```
{
```

```
    int sayac=1;
```

```
    float toplam=0.0;
```

```
    while (sayac<=50)
```

```
    {
```

```
        printf("Ogrenci notu giriniz");
```

```
        scanf("%d",&not);
```

```
        toplam=toplam+1;
```

```
        sayac=sayac+1;
```

```
    }
```

```
    printf("Sinav ortalamasi: %0.2f", toplam/50);
```

```
    return(0);
```

```
}
```

Program 2

10 reel sayıyı girdi olarak alan ve pozitif sayıların toplamını bulan C programını yazınız.

Program 2-Çözüm

```
#include <stdio.h>

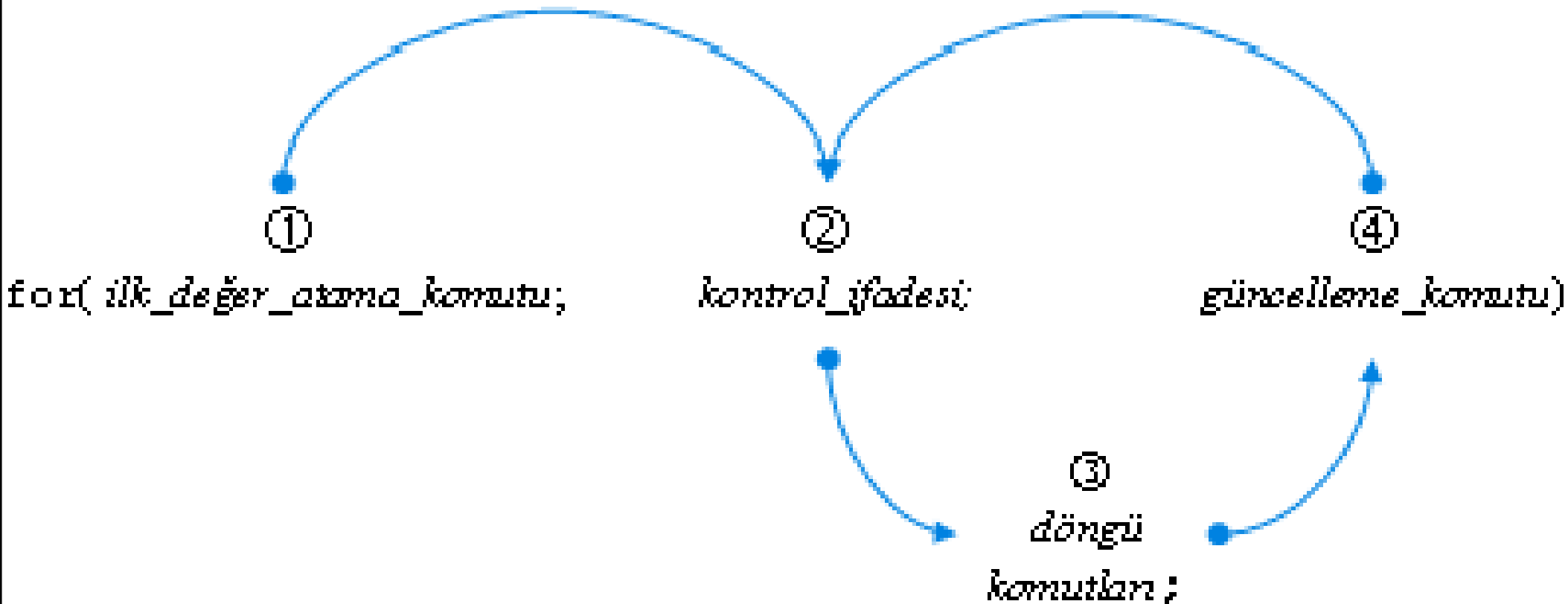
int main(void)
{
    int i=1;
    float sayi,toplam=0.0;

    while(i<=10)
    {
        printf("%d. sayiyi giriniz:",i);
        scanf("%f",&sayi);
        if(sayi>0)
            toplam=toplam+sayi;
        i=i+1;
    }

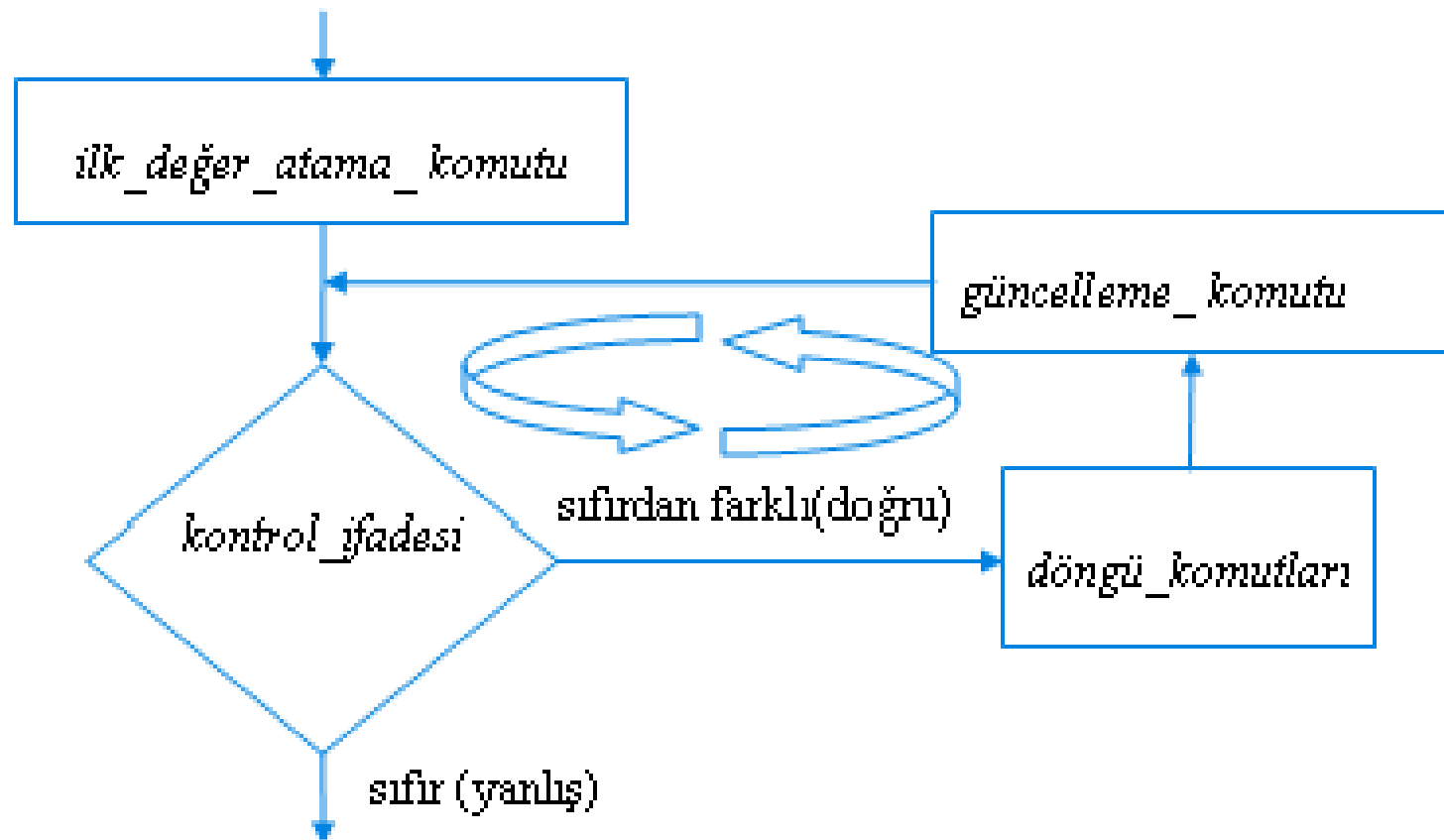
    printf("Pozitif sayilarin toplami: %0.2f", toplam);

    return(0);
}
```

```
for( ilk_değer_atama_komutu; kontrol_ifadesi; güncelleme_komutu)  
    döngü_komutları;
```



for(*ilk_değer_atama_komutu*; *kontrol_ifadesi*; *güncelleme_komutu*)
 döngü_komutları;



while döngüsü

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

Eşdeğer for döngüsü

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

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);
```

```
/*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++;  
}
```

Soru

Aşağıdaki for döğülerini inceleyerek Merhaba sözcüğünün kaç kere çıktıda görüntülendiğini belirtiniz.

- a) `for (s=10; s==10; s++)
 printf("Merhaba");`
- b) `for (s=10; s>=10; s++)
 printf("Merhaba");`
- c) `for (s=10; s<=10; s++)
 printf("Merhaba");`
- d) `for (s=10; s<10; s++)
 printf("Merhaba");`
- e) `for (s=13; s>9; s--)
 printf("Merhaba");`

Program 3

Aşağıda verilen matematiksel ifadeleri **for ve while** döngüsü kullanarak hesaplayacak olan program parçacıklarını yazınız

a) $1+2+3+4+5+\dots+1000$

b) $2+4+6+\dots+1000$

c) $1+1/2+1/3+1/4+\dots+1/100$

Program 4

Bir sınıfta 10 öğrencisi bulunan ve sınavı 100 puan üzerinden notlayan bir öğretmen, bu sınavın ortalamasını hesaplamak istemektedir. Bu sınavın ortalamasını hesaplayacak olan programı **for döngüsünü** kullanarak yazınız

Ödev

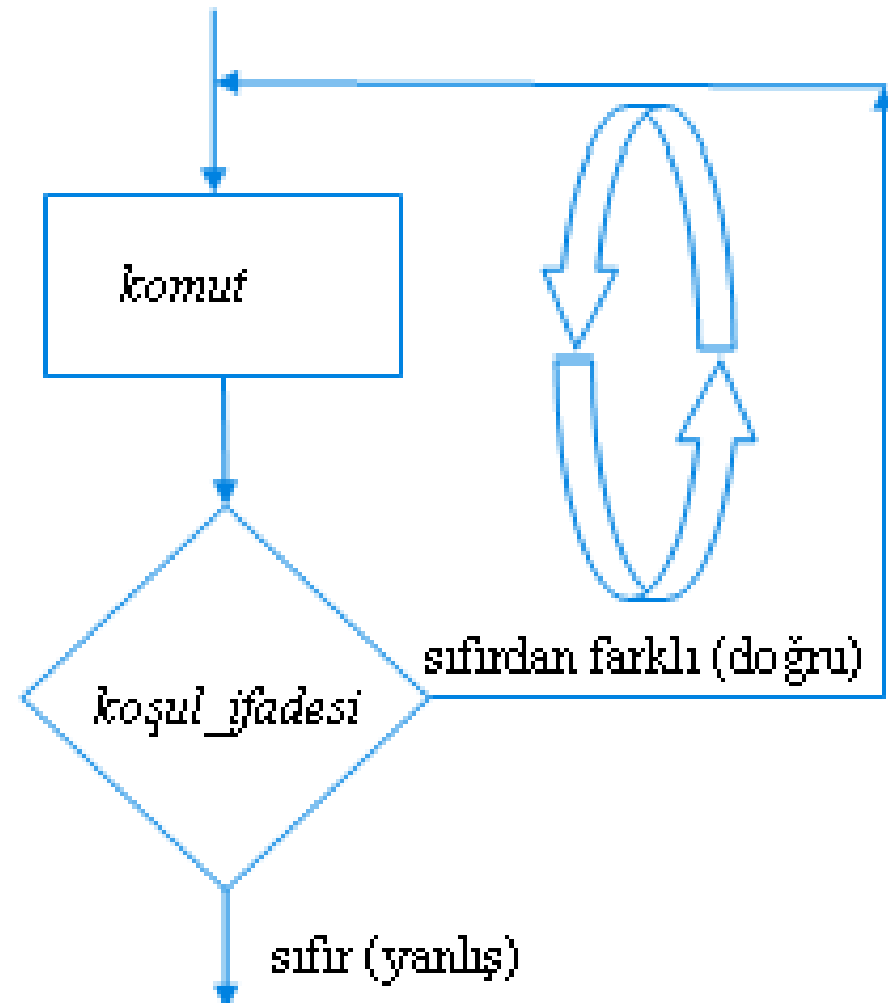
- Bir çiftlikte tavşanların sayısı 1042 iken, kuşların sayısı 2272'dir.
- Bir yılda tavşanlar %3.8 oranında çoğalırken, kuşlar %1.2 oranında çoğalmaktadır.
- Tavşanların, kuşların sayısını kaç yıl sonra geçeceğini bulan bir C programı yazınız.

Not: while döngüsünü kullanınız



do while Komutu

do
komut;
(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

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):

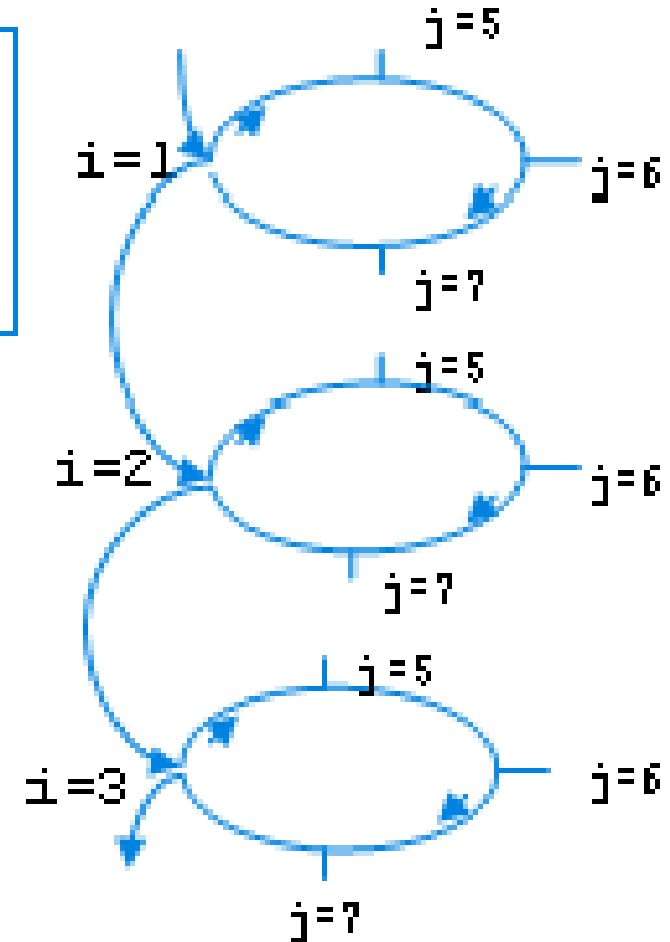
```
#include <stdio.h>
#define PI 3.141592654
int main(void)
{
    int cevap;
    double a,r,alan;
    do
    {
        /*Menunun goruntulenmesi*/
        printf("\n      1. Kare");
        printf("\n      2. Daire");
        printf("\n      3. Bitir");
        printf("\n\nSeceneginizi belirtiniz: ");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

```
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);
}
```


İççe Döngüler

```
for (i=1; i<=3; ++i)
```

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



Örnek:

Bir başka örnek olarak aşağıdaki çıktıyı veren program parçasını içİçe 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);
```