

Bölüm 4
Seçme Komutları

Koşul İfadesi

if Komutu

Bileşik Komut

İçiçe if Komutu

if-else Eşleşmesi

Soru İşareti Operatörü

switch Komutu

Koşul İfadesi

Problem çözme sürecinde program akışını değiştirecek farklı komutlara da ihtiyaç duyulur. Bu komutlara program akışını kontrol etmeleri nedeniyle **kontrol komutları** adı verilir.

Kontrol komutları **seçme** ve **döngü** komutları olmak üzere iki grupta toplanır.

Koşul İfadesi

Koşul İfadeleri ilişkisel ve matıksal operatörler kullanılarak oluşturulur.

İlişkisel Operatörler değerlerin karşılaştırılmasında kullanılır.

İlişkisel Operatör	Anlamı
>	Büyük
>=	Büyük veya eşit
<	Küçük
<=	Küçük veya eşit
==	Eşit
!=	Eşit değil

Koşul İfadesi

Örnek:

Koşul ifadesi	Sonuç
(21+sayac) <= 50	1 (doğru)
alfa == 65.8	0 (yanlış)
sayac >= 12	1 (doğru)
ch == 'h'	0 (yanlış)
ch != 'h'	1 (doğru)
alfa < (sayac-1)	0 (yanlış)

Operatörlerin Öncelik Sıralaması

Öncelik Sırası	Operatör	Özellik
En yüksek	( )	İçten dışa
↓	- +	Tekli operatör (sağdan sola)
	* / %	İkili operatör (soldan sağa)
	+ -	İkili operatör (soldan sağa)
	< <= > >=	İkili operatör (soldan sağa)
	== !=	İkili operatör (soldan sağa)
En düşük	=	İkili operatör (sağdan sola)

Koşul İfadesi

Mantıksal Operatörler mantıksal işlemleri gerçekleştirmek amacıyla kullanılır.

&& Operatörü

ifade1	*ifade2*	*ifade1 && ifade2*
Sıfır hariç (doğru)	Sıfır hariç (doğru)	1 (doğru)
Sıfır hariç (doğru)	0 (yanlış)	0 (yanlış)
0 (yanlış)	Sıfır hariç (doğru)	0 (yanlış)
0 (yanlış)	0 (yanlış)	0 (yanlış)

! Operatörü

Ifade	*! ifade*
Sıfır hariç (doğru)	0 (yanlış)
0 (yanlış)	1 (doğru)

|| Operatörü

Ifade1	*ifade2*	*ifade1 || ifade2*
Sıfır hariç (doğru)	Sıfır hariç (doğru)	1 (doğru)
Sıfır hariç (doğru)	0 (yanlış)	1 (doğru)
0 (yanlış)	Sıfır hariç (doğru)	1 (doğru)
0 (yanlış)	0 (yanlış)	0 (yanlış)

Koşul İfadesi

Örnek:

Koşul ifadesi	Sonuç
(a<5.0)&&(d>b/2)	0 (yanlış)
!(b!=8.5)	0 (yanlış)
(c>=10)||(ch=='h')	0 (yanlış)
!(d-2.5<0)	1 (doğru)
(c==0)&&(d!=0)	1 (doğru)
(ch=='y')||(ch=='Y')	0 (yanlış)
a &&(a<10)	1 (doğru)

Operatörlerin Genişletilmiş Öncelik Sıralaması

Öncelik sırası	Operatör	Özellik
En yüksek	( )	İçten dışa
	- + !	Tekli operatör (sağdan sola)
	* / %	İkili operatör (soldan sağa)
	+ -	İkili operatör (soldan sağa)
	< <= > >=	İkili operatör (soldan sağa)
	== !=	İkili operatör (soldan sağa)
	&&	İkili operatör (soldan sağa)
	||	İkili operatör (soldan sağa)
En düşük	=	İkili operatör (sağdan sola)

Koşul İfadesi

Örnek:

```
int i=5, j=10, k=20;

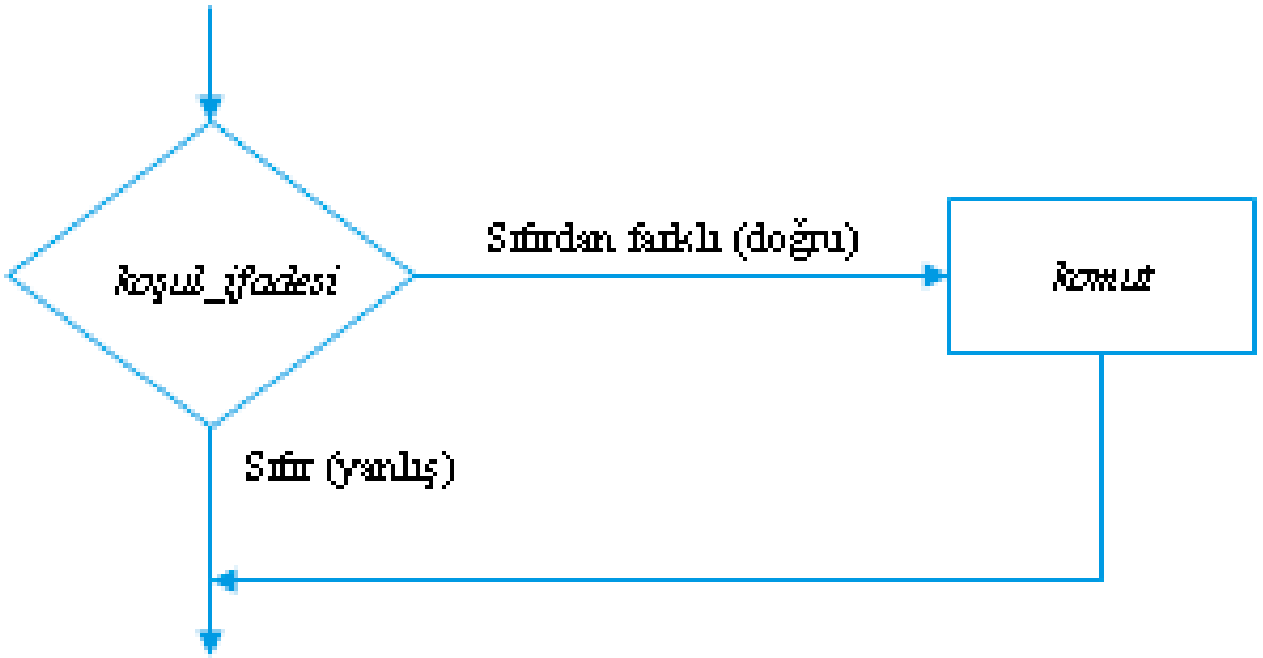
i > 5 * k % 3 && !(k-2!=18)
                    ↓
i > 5 * k % 3 && !(18!=18)
                      ↓
     i > 5 * k % 3 && !0
                      ↓
     i > 5 * k % 3 && 1
          ↓
      i > 100 % 3 && 1
            ↓
        i > 1 && 1
          ↓
         1 && 1
            ↓
            1
```

if Komutu

if komutu belirtilen koşul altında hangi komut veya komutların yürüleceğini belirleyen bir komuttur.

Tek Yönlü if

if (*koşul_ ifadesi*)
    komut;



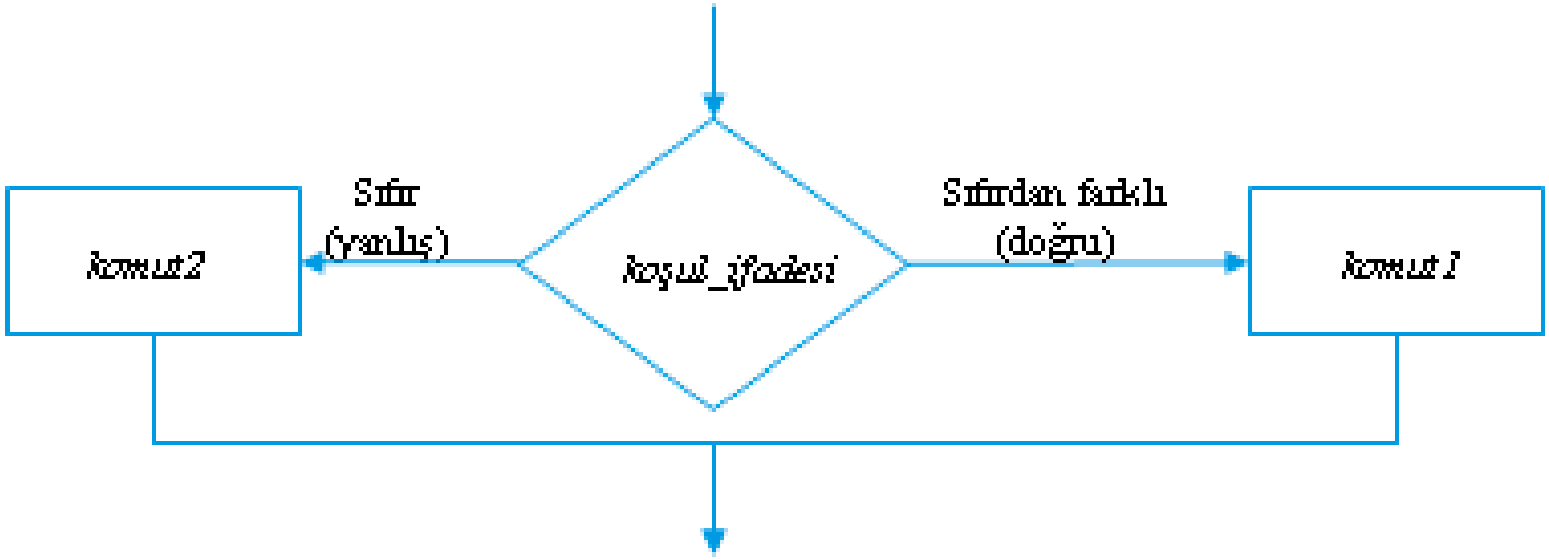
if komutunda ilk önce koşul ifadesinin değeri hesaplanır. Eğer koşul ifadesinin değeri sıfırdan farklı (doğru) ise *komut* yürütülür. Eğer koşul ifadesinin değeri 0 (yanlış) ise *komut* yürütülmez.

if Komutu

```
1  printf("Bir sayi giriniz:");
2  scanf("%d",&sayi);
3  if(sayi>0)
4     printf("%d pozitif sayidir.\n",sayi);
5  printf("-----");
```

Çift Yönlü if

```
if (koşul_ifadesi)
   komut1;
else
   komut2;
```



Bu yapıda *koşul_ifadesi* sıfırdan farklı (doğru) ise *komut1* yürütülür; *koşul_ifadesi* sıfır ise *komut2* yürütülür.

if Komutu

```
1  printf("Bir sayi giriniz: ");
2  scanf("%d",&sayi);
3  if(sayi>0)
4      sonuc=sayi*5;
5  else
6      sonuc=2-sayi;
7  printf("Sonuc: %d",sonuc);
```

Bileşik Komut

Bileşik Komut içinde bir çok komut bulunduran tek bir komuttur.

```
{
    komut1;
    komut2;
     .
     .
     .
    komutn;
}
```

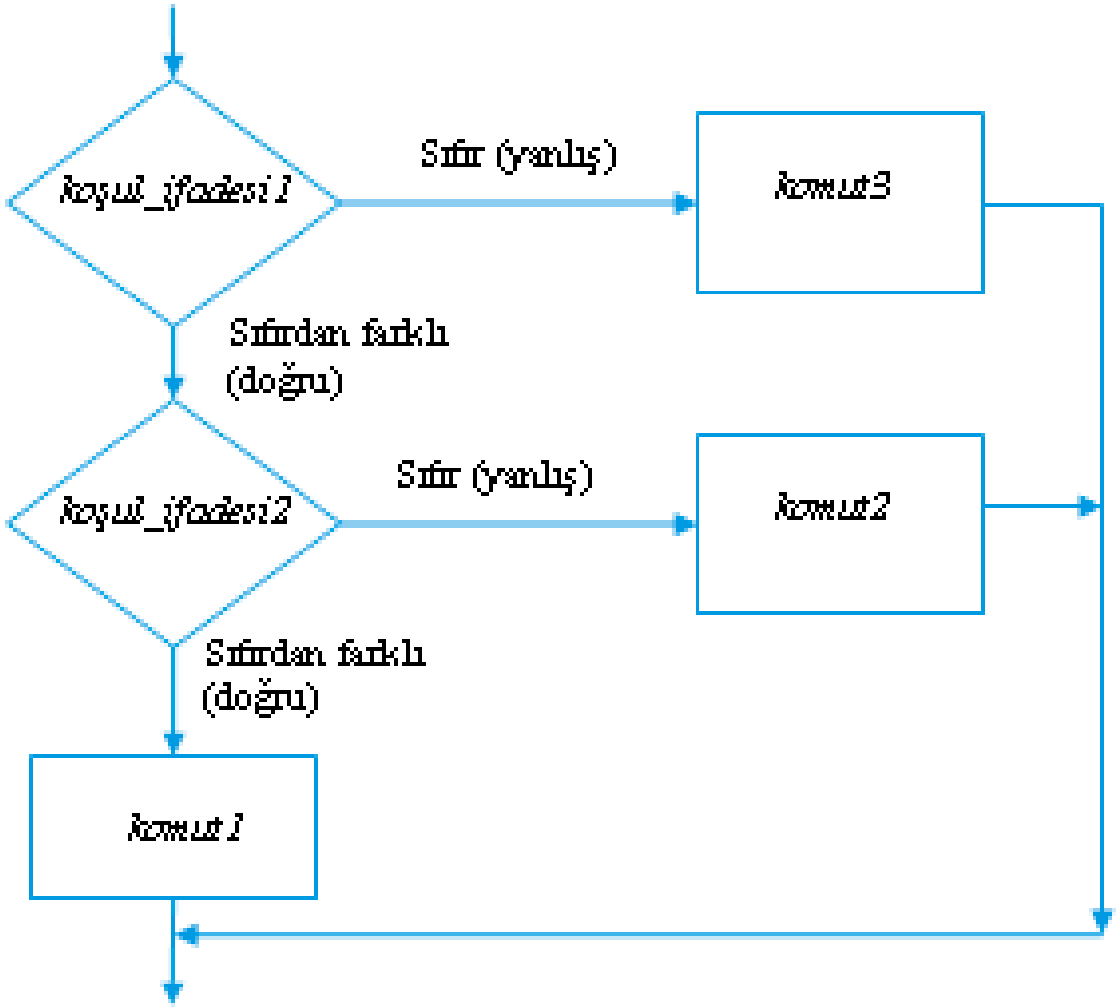
Örnek:

```
{
  printf("Sayi giriniz: ");
  scanf("%d",&a);
  printf("%d sayisinin karesi %d",a,a*a);
}
```

İçiçe if

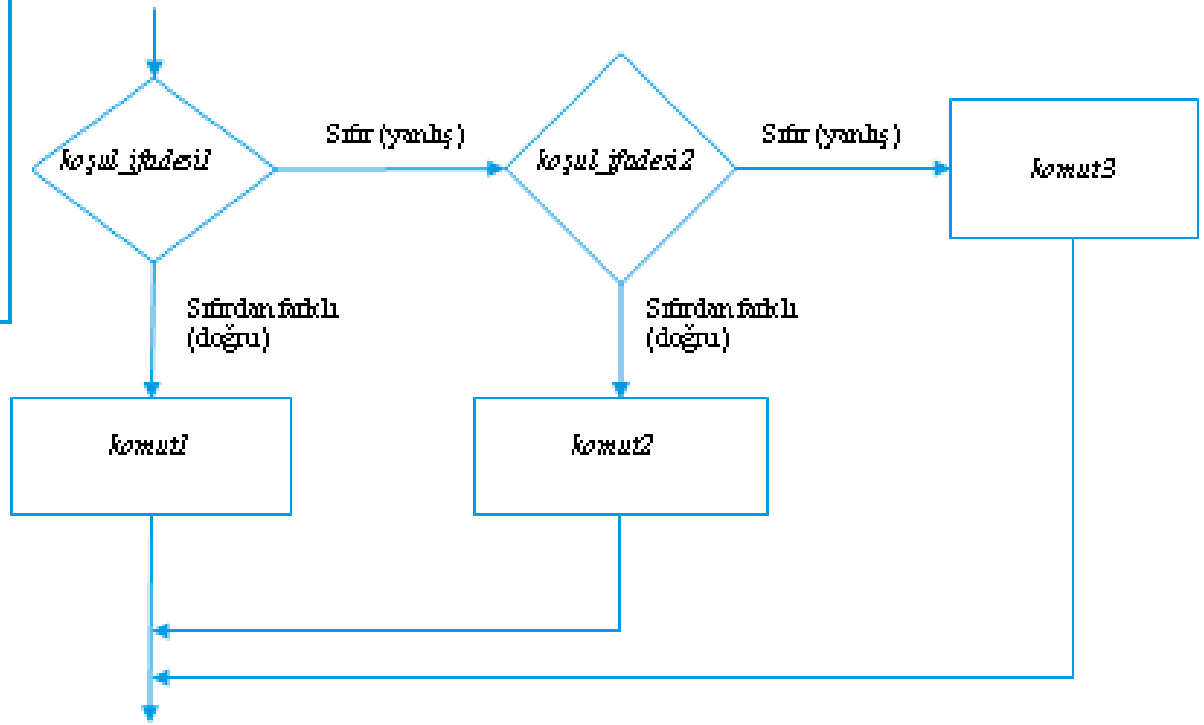
Durum I:

```
if (koşul_ifadesi1)
    if (koşul_ifadesi2)
        komut1;
    else
        komut2;
else
    komut3;
```



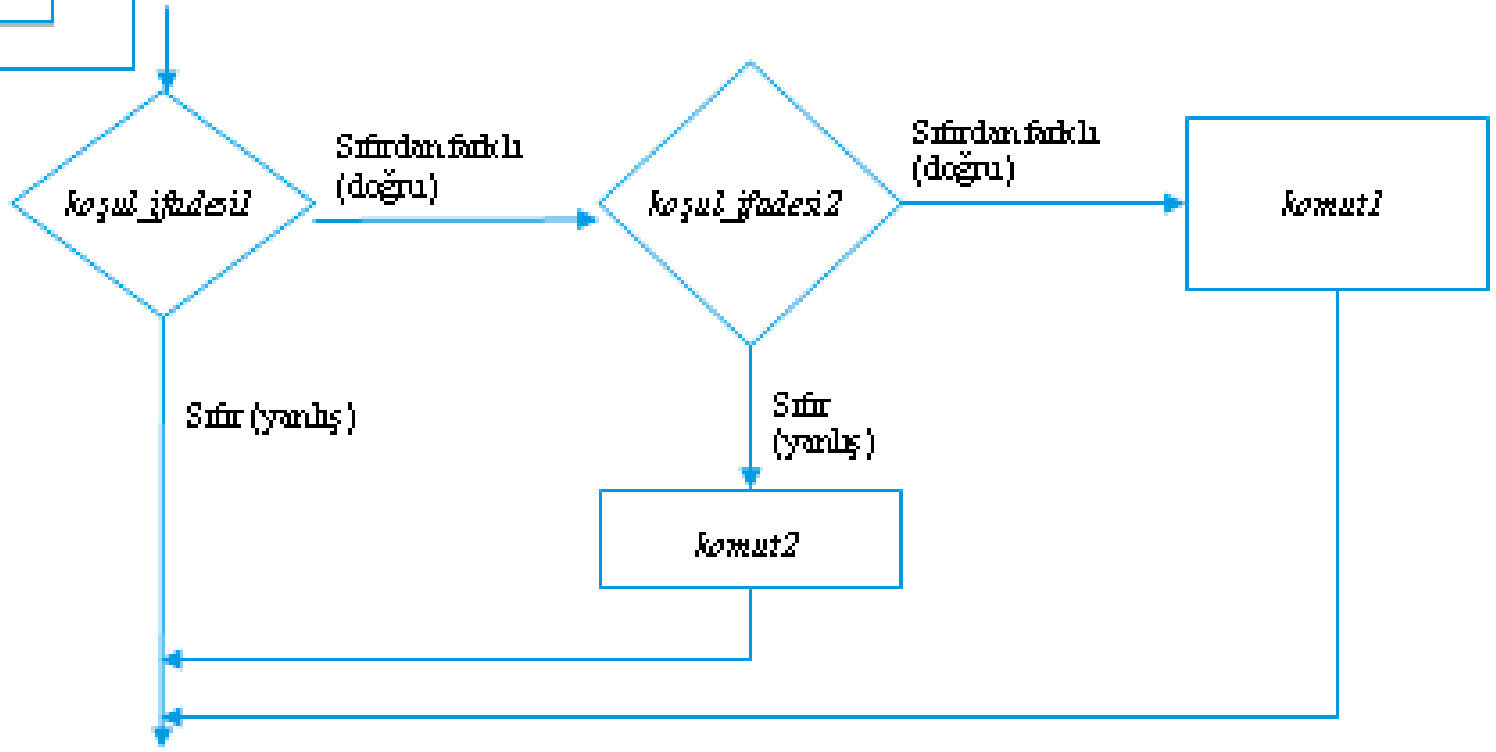
Durum II:

```
if (koşul_ifadesi1)
    komut1;
else
    if (koşul_ifadesi2)
        komut2;
    else
        komut3;
```



Durum III:

```
if (koşul_ifadesi1)
    if (koşul_ifadesi2)
        komut1;
    else
        komut2;
```



Örnek:

$$f(x,y)=\begin{cases} xy & x\geq 0,\ y\geq 0 \\ x+y & x\geq 0,\ y<0 \\ y-x & x<0,\ y\geq 0 \end{cases}$$

```
if (x>=0)
   if (y>=0)
        fonk=x*y;
   else
        fonk=x+y;
else
   if(y>=0)
        fonk=y-x;
```

if else Eşleşmesi

```
if(no_a>0)
    printf("%d", no_a);
else
    if(no_b<0)
        y=no_a*no_b;
    else
        y=no_b+no_a;
```

"Her else en yakın eşleşmemiş `if` ile eşleştirilir."

Soru işareti operatörü

ifade1? *ifade2* : *ifade3*

Soru işareti operatörü kullanarak

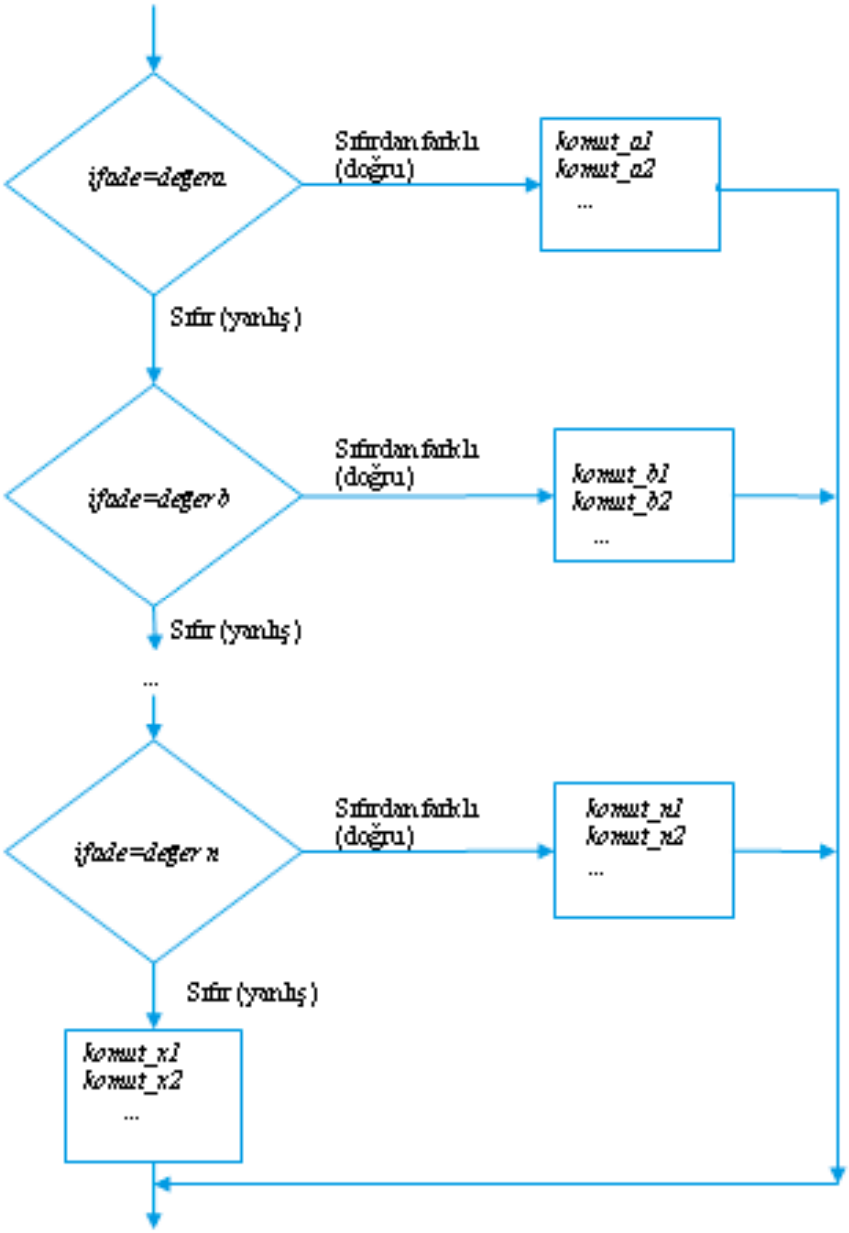
```
int x=10, y;
y=x>9 ? 100 : 200;
```

if komutu kullanarak

```
int x=10, y;
if(x>9)
    y=100;
else
    y=200;
```

switch Komutu

```
switch( ifade )
{ case     değer a: komut_a1;
                    komut_a2;
                       .
                       .
                       .
                    break;
  case     değer b: komut_b1;
                    komut_b2;
                       .
                       .
                       .
                    break;
     .
     .
     .
  case     değer n: komut_n1;
                    komut_n2;
                       .
                       .
                       .
                    break;
  default: komut_x1;
           komut_x2;
                       .
                       .
                       .
}
```



switch Komutu

Örnek:

```
scanf("%d",&sinif);
switch (sinif)
{     case 1:  printf("Birinci sinif ogrencisi\n");
               break;
      case 2:  printf("Ikinci sinif ogrencisi\n");
               break;
      case 3:  printf("Ucuncu sinif ogrencisi\n");
               break;
      case 4:  printf("Dorduncu sinif ogrencisi\n");
               break;
      case 5:  printf("Yuksek lisans ogrencisi\n");
               break;
      default: printf("Sinif bilgisi yanlis\n");
}
```