

# CENG 111 ALGORİTMALAR VE PROGRAMLAMA

Doç. Dr. Tufan TURACI

tturaci@pau.edu.tr

- Pamukkale Üniversitesi
- Mühendislik Fakültesi
- Bilgisayar Mühendisliği Bölümü
- Hafta 4
- 11 Ekim 2022

## 4. Hafta Konular

- **Karar Verme: Eşitlik ve Karşılaştırma Operatörleri**
- **Mantık Operatörleri**
- **Karşılaştırma Operatörleri**
  - if - else yapısı
  - switch-case yapısı
- **goto deyimi**
- **Çeşitli Örnekler**

# Karar Verme: Eşitlik ve Karşılaştırma Operatörleri

- Çalıştırılabilir C ifadeleri 2 iş yaparlar.
  - İşlem gerçekleştirir.( hesaplamalar, veri girdi/çıktısı)
  - Karar verir.
    - ❖ Örneğim; bir kişinin bir sınavdan aldığı not 60'tan büyükse yada 60'a eşitse 'geçtiniz' yazdırabiliriz.
- If kontrol yapısı
  - Bir koşulun doğruluğu yada yanlışlığına karar veren, if kontrol yapısı öncelikle basit bir biçimde tanıtılacaktır. Daha detaylı olarak sonra.
  - Eğer koşul doğru ise, if yapısının gövdesindeki ifade çalıştırılır.
    - ❖ Yanlışsa, if yapısının gövdesindeki ifade çalışmaz.
  - Çalıştırma süreci if yapısından hemen sonraki ifadeyle devam eder.

# Karar Verme: Eşitlik ve Karşılaştırma Operatörleri

Operatörler	C'deki karşılığı	C'de örneği	C'de anlamı
<i>Eşitlik operatörleri</i>			
=	==	x == y	x eşittir y
≠	!=	x != y	x eşit değildir y
<i>Karşılaştırma operatörleri</i>			
>	>	x > y	x büyüktür y
<	<	x < y	x küçüktür y
≥	>=	x >= y	x büyüktür ya da eşittir y
≤	<=	x <= y	x küçüktür yada eşittir y

# Genel Programlama Hataları

- ==, !=, >= ve <= operatörleri arasında boşluk kullanılması yazım hatasına sebep olur.
- !=, >= ve <= operatörlerinin ters çevrilerek =!, => ve =< şeklinde kullanılması yazım hatasına sebep olur.
- == operatörünün = ile karıştırılması.

```

#include <stdio.h>
#include <conio.h>
int main()
{
    int num1, num2;

    printf( "iki tam sayi girin.\n" );
    printf( "Bu iki sayinin karsilastirmasi yapılacaktır: " );
    scanf( "%d%d", &num1, &num2 );

    if ( num1 == num2 )
        printf( "%d esittir %d\n", num1, num2 );

    if ( num1 != num2 )
        printf( "%d esit degildir %d\n", num1, num2 );

    if ( num1 < num2 )
        printf( "%d kucuktur %d\n", num1, num2 );

    if ( num1 > num2 )
        printf( "%d buyuktur %d\n", num1, num2 );

    if ( num1 <= num2 )
        printf( "%d kucuktur ya da esittir %d\n",
            num1, num2 );

```

```

if ( num1 >= num2 )
    printf( "%d buyuktur ya da esittir %d\n",
        num1, num2 );
    getch();
    return 0;
}

```

```

iki tam sayi girin.
Bu iki sayinin karsilastirmasi yapılacaktır: 3 7
3 esit degildir 7
3 kucuktur 7
3 kucuktur ya da esittir to 7

```

```

iki tam sayi girin.
Bu iki sayinin karsilastirmasi yapılacaktır: 22 12
22 esit degildir 12
22 buyuktur 12
22 buyuktur ya da esittir to 12

```

```

iki tam sayi girin.
Bu iki sayinin karsilastirmasi yapılacaktır: 11 11
11 esittir 11
11 kucuktur ya da esittir to 11
11 buyuktur ya da esittir to 11

```

## Şu ana kadar anlatılan operatörlerin öncelikleri ve işleyişleri

Operatörler	İşleyişleri
( )	Soldan sağa
*   /   %	Soldan sağa
+   -	Soldan sağa
<   <=   >   >=	Soldan sağa
==   !=	Soldan sağa
=	Sağdan sola

# ÖRNEKLER

**Soru1:** 1'den 4'e kadar olan tamsayıları ekrana tek satırda görülecek şekilde yazdıran bir programı aşağıdaki metotları kullanarak yazınız.

- a) Bir printf ifadesi kullanarak ve hiç format belirteci kullanmadan.
- b) Bir printf ifadesi ve dört format belirteci kullanarak.
- c) Dört printf ifadesi kullanarak.

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

```
#include <conio.h>
```

```
int main()
```

```
{
```

```
printf( "1 2 3 4\n\n" ); // a
```

```
printf( "%d %d %d %d\n\n", 1, 2, 3, 4 ); //b
```

```
printf( "1 " ); //c
```

```
printf( "2 " );
```

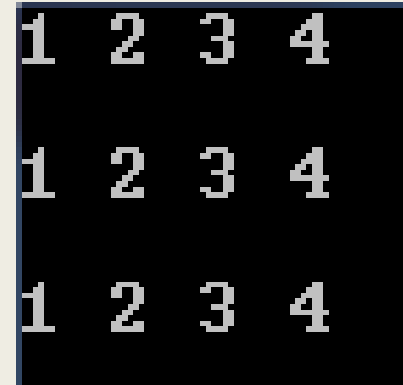
```
printf( "3 " );
```

```
printf( "4\n" );
```

```
getch();
```

```
return 0;
```

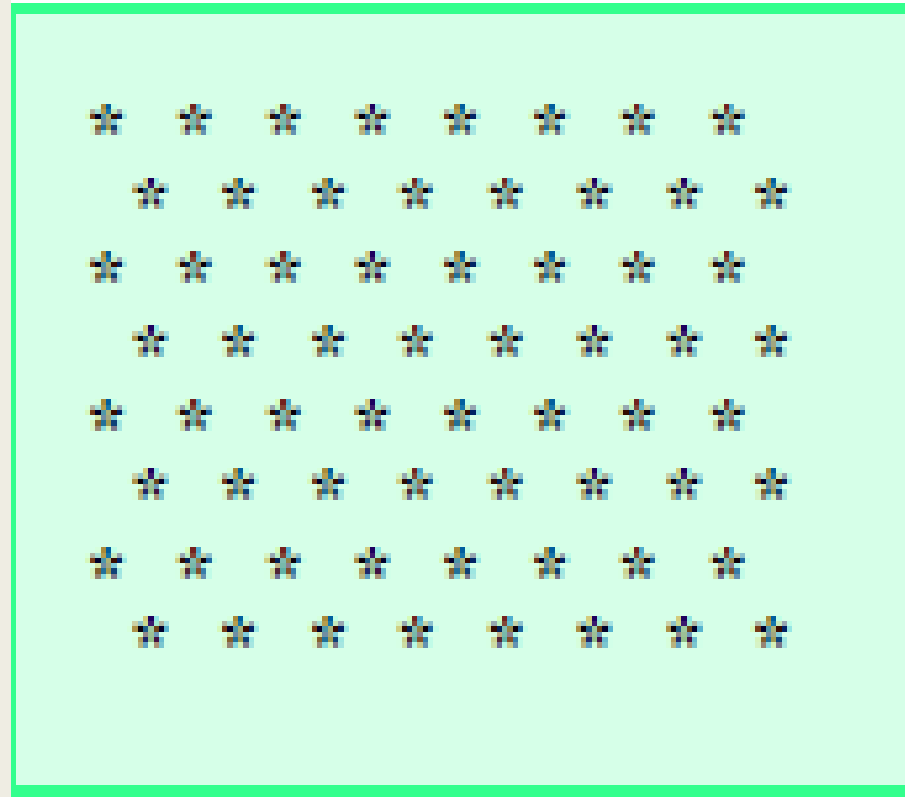
```
}
```



The image shows a 3x4 grid of numbers 1, 2, 3, 4 displayed on a black background. The numbers are white and arranged in three rows and four columns. The first row contains 1, 2, 3, 4. The second row contains 1, 2, 3, 4. The third row contains 1, 2, 3, 4.



**Soru2:** Aşağıdaki deseni sekiz printf ifadesiyle ekrana yazdıran bir program yazınız. Daha sonra aynısını kullanabileceğiniz en az printf ifadesiyle yazınız.



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

```
#include <conio.h>
```

```
int main()
```

```
{
```

```
printf( "Sekiz printf() ifadesi ile: \n" );
```

```
printf( "* * * * * * * *\n" );
```

```
printf( " * * * * * * * *\n" );
```

```
printf( "* * * * * * * *\n" );
```

```
printf( " * * * * * * * *\n" );
```

```
printf( "* * * * * * * *\n" );
```

```
printf( " * * * * * * * *\n" );
```

```
printf( "* * * * * * * *\n" );
```

```
printf( " * * * * * * * *\n" );
```

```
printf( "\nSimdi ise bir printf() ifadesi ile: \n" );
```

```
printf( "* * * * * * * *\n * * * * * * * *\n"
```

```
"* * * * * * * *\n * * * * * * * *\n"
```

```
"* * * * * * * *\n * * * * * * * *\n"
```

```
"* * * * * * * *\n * * * * * * * *\n" );
```

```
getch();
```

```
return 0;
```

```
}
```

```
Sekiz printf() ifadesi ile:
```

```
* * * * * * * *  
 * * * * * * * *  
* * * * * * * *  
 * * * * * * * *  
* * * * * * * *  
 * * * * * * * *  
* * * * * * * *  
 * * * * * * * *
```

```
Simdi ise bir printf() ifadesi ile:
```

```
* * * * * * * *  
 * * * * * * * *  
* * * * * * * *  
 * * * * * * * *  
* * * * * * * *  
 * * * * * * * *  
* * * * * * * *  
 * * * * * * * *
```

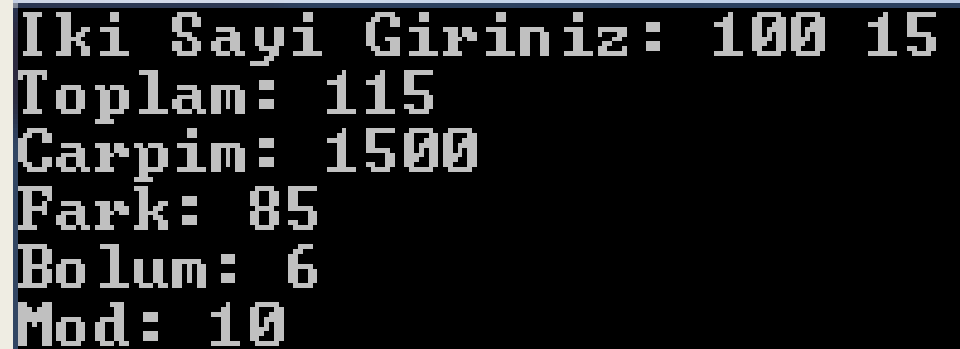
**Soru3:** Kullanıcının iki tamsayı girmesini ve sayıların kullanıcıdan aldıktan sonra bu sayıların toplamlarını, çarpımlarını, farklarını, bölümlerini ve modlarını bulan bir program yazınız.

```
#include <stdio.h>
#include <conio.h>
int main()
{
    int x;
    int y;

    printf( "İki Sayı Giriniz: ");
    scanf( "%d%d", &x, &y );

    printf( "Toplam: %d\n", x + y );
    printf( "Carpim: %d\n", x * y );
    printf( "Fark: %d\n", x - y );
    printf( "Bolum: %d\n", x / y );
    printf( "Mod: %d\n", x % y );

    getch();
    return 0;
}
```



```
İki Sayı Giriniz: 100 15
Toplam: 115
Carpim: 1500
Fark: 85
Bolum: 6
Mod: 10
```

## ATAMA OPERATÖRLERİ

Bu operatörler bir değişkene, bir sabit veya bir aritmetik ifade atamak (eşitlemek) için kullanılır. *Birleşik atama*: bazı ifadelerde işlem operatörü ile atama operatörü birlikte kullanılarak, ifadeler daha kısa yazılabilir. Eğer ifade

*değişken = değişken [operatör] aritmetik ifade;*

şeklinde ise, daha kısa bir biçimde

*değişken [operatör] = aritmetik ifade;* olarak yazılabilir.

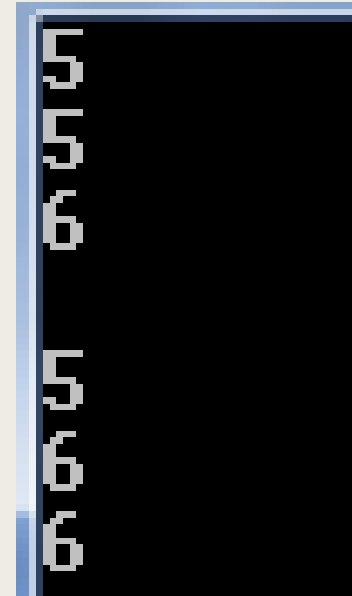
Açıklama	Örnek	Anlamı
Atama	$x = 7;$	$x = 7;$
ekleyerek atama	$x += 3$	$x = x + 3$
eksilterek atama	$x -= 5$	$x = x - 5$
çarparak atama	$x *= 4$	$x = x * 4$
bölerek atama	$x /= 2$	$x = x / 2$
bölüp, kalanını atama	$x \% = 9$	$x = x \% 9$
bir arttırma	$x++$ veya $++x$	$x = x + 1$
bir azaltma	$x--$ veya $--x$	$x = x - 1$

```
#include <stdio.h>
#include <conio.h>
int main()
{
    int c = 5;

    printf( "%d\n", c );
    printf( "%d\n", c++ );
    printf( "%d\n\n", c );

    c = 5;
    printf( "%d\n", c );
    printf( "%d\n", ++c );
    printf( "%d\n", c );

    getch();
    return 0;
}
```



# C Veri Türleri

Veri tipi program içinde kullanılacak değişken, sabit, fonksiyon isimleri gibi tanımlayıcıların tipini, yani bellekte ayrılacak bölgenin büyüklüğünü, belirlemek için kullanılır.

C programlama dilinde 5 tane temel veri tipi aşağıdaki şekildedir.

**char:** karakter veriler

**int:** tamsayı veriler

**float:** tek duyarlıklı kayan noktalı sayılar

**double:** Çift duyarlıklı kayan noktalı sayılar

**void:** Değer içermeyen verilerdir.

--Aşağıdaki özel niteleyici ifadeler ile farklı veri tipleri oluşturulabilir.

- short (kısa)
- long (uzun)
- unsigned (işaretsiz)

-- Özel niteleyiciler değişkenin bellekte kaplayacağı alanı değiştirilebilirler.

- Normal tamsayı: En fazla 32 bit (4 bayt) uzunlukta
- uzun tamsayı: En fazla 64 bit (8 bayt) uzunlukta
- kısa tamsayı 16 bit (2 bayt) geçmeyecek uzunluktadır.

-- Uzun ve kısa tamsayı için int kullanılmayabilir.

short x; (short int x;)

long x; (long int x;)

-- İşaretsiz ön eki ile sayının sıfır veya sıfırdan büyük olması sağlanır.

**Örnek:**

short int ---> -32768 ile 32767 arası verileri saklar

unsigned short int ---> 0 ile 65535 arası verileri saklar

C programında veri tiplerinin bellekte kapladığı alan sizeof operatörü ile öğrenilebilir.

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

```
#include <conio.h>
```

```
int main()
```

```
{
```

```
printf("char          : %d byte \n",sizeof(char));
```

```
printf("int           : %d byte \n",sizeof(int));
```

```
printf("short          : %d byte \n",sizeof(short));
```

```
printf("unsigned char   : %d byte \n",sizeof(unsigned char));
```

```
printf("unsigned short  : %d byte \n",sizeof(unsigned short));
```

```
printf("float           : %d byte \n",sizeof(float));
```

```
printf("double          : %d byte \n",sizeof(double));
```

```
getch();
```

```
return 0;
```

```
}
```

```
char          : 1 byte
int           : 4 byte
short         : 2 byte
unsigned char  : 1 byte
unsigned short : 2 byte
float         : 4 byte
double        : 8 byte
```

```
-----
Process exited after 2.188 seconds with return value 0
Press any key to continue . . .
```



Veri Tipi	Veri Tipi Açıklama	Bayt (Bellekte)	Alt Değer	Üst Değer
<del>char</del>	Tek bir karakter veya küçük tamsayı için	1	-128	127
İşaretsiz <del>char</del>			0	255
<del>short int</del>	Kısa tamsayı için	2	-32,768	32,767
İşaretsiz <del>short int</del>			0	65,535
<del>int</del>	Tamsayı için	4	-2,147,483,648	2,147,483,647
İşaretsiz <del>int</del>			0	4,294,967,295
<del>long int</del>	Uzun tamsayı için	8	-9,223,372,036,854,775,808	9,223,372,036,854,775,807
İşaretsiz <del>long int</del>			0	18,446,744,073,709,551,615
<del>float</del>	Tek duyarlı gerçel sayı için (7 basamak)	4	-3. <u>4e</u> +/- 38	+3. <u>4e</u> +/- 38
<del>double</del>	Çift duyarlı gerçel sayı için (15basamak)	8	-1. <u>7e</u> +/- 308	+1. <u>7e</u> +/- 308

# C dilinde sabit tanımlama

- **const** komutu sabit bir değişken tanımlayabiliriz.

**const float** PI = 3.14;

**const int** UZAK\_AB= 225;

**const int** EOF= -1;

- Sabit içerikleri program boyunca değiştirilemez. Yalnızca kullanılabilir.

## const komutu ile

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

int main()
{
    const int sabit_a=15;
    const int sabit_b=20;
    printf("Sabitlerin toplami= %d", sabit_a + sabit_b);

    getch();
    return 0;
}
```

```
Sabitlerin toplami= 35
-----
Process exited after 21.64 seconds with return value 0
Press any key to continue . . .
```

## define komutu ile

```
#include <stdio.h>
#include <conio.h>
#define sabit_a 15
#define sabit_b 20

int main()
{
    printf("Sabitlerin toplami= %d", sabit_a + sabit_b);
    getch();
    return 0;
}
```

```
Sabitlerin toplami= 35
-----
Process exited after 2.146 seconds with return value 0
Press any key to continue . . .
```

# Koşul

- Sonucu doğru veya yanlış olan ifadeler koşul denir.

 DOĞRU: 1  YANLIŞ: 0

- Koşul ifadelerinin ve akış kontrollerinin programlamada her zaman önemli bir yeri vardır.
- Yazdığımız programda koşulların nasıl davranacağını belirterek bir akış kontrolü sağlanabilir.
- En temel karşılatırma IF-ELSE VE SWITCH-CASE bloklarıdır.

# Mantık Operatörleri

- && ( mantıksal VE)
  - *İki koşulda true ise true döndürür.*
- || ( mantıksal VEYA)
  - *İki koşuldan biri true ise true döndürür.*
- ! ( mantıksal DEĞİL)
  - *Koşul yanlış olduğunda izlenecek yollu belirler.*
- Döngülerde ki koşul durumları;

İfade	Sonuç
true && false	false
true    false	true
!false	true

<b>Operatör</b>	<b>Açıklama</b>	<b>Örnek</b>	<b>Anlamı</b>
<b>&amp;&amp;</b>	<b>VE</b>	$a > b \ \&\& \ b < c$	a, b'den büyük <b>VE</b> b, c'den küçük
<b>  </b>	<b>VEYA</b>	$a > b \    \ b < c$	a, b'den büyük <b>VEYA</b> b, c'den küçük
<b>!</b>	<b>DEĞİL</b>	$!(a > b)$	a, b'den büyük değilse (a, b'den küçük <b>VEYA</b> b'ye eşitse)

**1&&1=1**

**1&&0=0**

**0&&1=0**

**0&&0=0**

**1||1=1**

**1||0=1**

**0||1=1**

**0||0=0**

**!0=1**

**!1=0**

## Örnek:

```
#include<stdio.h>
#include<conio.h>
int main()
{ int a=3,b=5,c=-1,x1,x2,x3;

  x1= ((a<b) && !(1)) ;
  x2= ((a!=c) || (0==1)) ;
  x3= ((b==c) || (a==3)) ;

  printf ("x1 degeri= %d\n",x1);
  printf ("x2 degeri= %d\n",x2);
  printf ("x3 degeri= %d\n",x3);

  getch ();
  return 0;
}
```

```
x1 degeri= 0
x2 degeri= 1
x3 degeri= 1
```

```
-----
Process exited after 1.644 seconds with return value 0
Press any key to continue . . .
```

## Karşılaştırma Yapıları

Verilen şart veya şartlara göre program akışını değiştirmek için kullanılan yapılardır. Bu komutlar ile verilen şartın sonucuna göre istenen işlemler yapılır ve\veya istenen işlemlerin atlanması sağlanır. C\C++ programlama dilinde bu işlemler için *if-else* ve *switch-case* yapıları kullanılır.

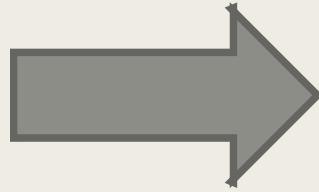
- **if-else yapısı**

Program akışını verilen şart(koşul) veya şartlara göre değiştirmek için kullanılır.

$$\text{İf (şart)} \left\{ \begin{array}{l} \text{komut veya} \\ \text{komutlar} \end{array} \right\} ;$$
$$\text{if (şart)} \left\{ \begin{array}{l} \text{komut veya} \\ \text{komutlar} \end{array} \right\} ; \text{ else } \left\{ \begin{array}{l} \text{komut veya} \\ \text{komutlar} \end{array} \right\}$$



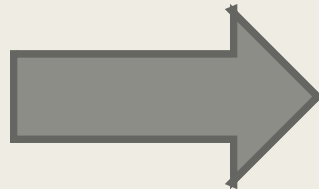
- If (a>20) s++;



Tek komut varsa { } kullanılmayabilir.

- If (a>20) s++;  
    else  
        s--;

- If (a>20) {s++;  
            x--;}



Birden fazla komut varsa { } kullanılmalıdır!!!

- If (a>20) {s++;  
            x--;}  
    else  
        {s--;  
        x++;}

## Örnek: { } kullanılmadı...

```
#include<stdio.h>
#include<conio.h>
int main()
{ int a;
  printf ("a degerini giriniz\n");
  scanf ("%d",&a);

  printf ("Girilen a degeri= %d\n", a);

  if (a>20) a++;
    else
      a--;

  printf ("Son a degeri= %d", a);
  getch ();
  return 0;
}
```

```
a degerini giriniz
23
Girilen a degeri= 23
Son a degeri= 24
-----
Process exited after 4.253 seconds with return value 0
Press any key to continue . . . _
```

```
a degerini giriniz
5
Girilen a degeri= 5
Son a degeri= 4
-----
Process exited after 4.228 seconds with return value 0
Press any key to continue . . . _
```

## Örnek: { } kullanıldı...

```
#include<stdio.h>
#include<conio.h>
int main()
{ int a;
  printf ("a degerini giriniz\n");
  scanf ("%d",&a);

  printf ("Girilen a degeri= %d\n", a);

  if (a>20) {a++;}
    else
      {a--;}

  printf ("Son a degeri= %d", a);
  getch ();
  return 0;
}
```

```
a degerini giriniz
23
Girilen a degeri= 23
Son a degeri= 24
-----
Process exited after 4.253 seconds with return value 0
Press any key to continue . . . _
```

```
a degerini giriniz
5
Girilen a degeri= 5
Son a degeri= 4
-----
Process exited after 4.228 seconds with return value 0
Press any key to continue . . . _
```

# İç-içe if...else Yapısı

**Kullanım şekli :**

```
if(koşul_1) {  
    komutlar; //(küme_1)  
}  
else if(koşul_2) {  
    komutlar; // (küme_2)  
}  
....  
....  
else if(koşul_n-1) {  
    komutlar; //(küme_n-1)  
}  
else {  
    komutlar; //(küme_n)  
}
```

**Örnek:** 0 ile 100 arasında girilen bir pozitif tam sayının hangi çeyrekte olduğunu bulan C programını yazınız.

### Çeyrekler

0-25 : 1. çeyrek

26-50 : 2. çeyrek

51-75 : 3. çeyrek

76-100 : 4. çeyrek

```
#include<stdio.h>
#include<conio.h>
int main()
{ int x;
  printf ("x sayisini giriniz\n");
  scanf ("%d",&x);

  if ((x>75) && (x<=100)) printf ("Sayi 4. ceyrektedir\n");
  else if ((x>50) && (x<=75)) printf ("Sayi 3. ceyrektedir\n");
  else if ((x>25) && (x<=50)) printf ("Sayi 2. ceyrektedir\n");
  else if ((x>=0) && (x<=25)) printf ("Sayi 1. ceyrektedir\n");
  else printf ("uygun sayi girilmemistir");

  getch ();
  return 0;
}
```

```
x sayisini giriniz
56
Sayi 3. ceyrektedir

-----
Process exited after 5.075 seconds with return value 0
Press any key to continue . . .
```

```
x sayisini giriniz
98
Sayi 4. ceyrektedir

-----
Process exited after 4.986 seconds with return value 0
Press any key to continue . . .
```

```
x sayisini giriniz
190
uygun sayi girilmemistir

-----
Process exited after 4.526 seconds with return value 0
Press any key to continue . . .
```

# ÇEŞİTLİ ÖRNEKLER

## ÖRNEK 1:

Girilen 4 sayının en büyüğünü bulan programı yazınız.

```
#include <stdio.h>
#include <conio.h>
int main ()
{int x,y,z,t,enb;
 printf("1.sayiyi giriniz\n"); scanf("%d",&x);
 printf("2.sayiyi giriniz\n"); scanf("%d",&y);
 printf("3.sayiyi giriniz\n"); scanf("%d",&z);
 printf("4.sayiyi giriniz\n"); scanf("%d",&t);
 if(x>y) enb=x; else enb=y;
 if(z>enb) enb=z;
 if(t>enb) enb=t;
 printf("Bu sayilarin en buyugu= %d",enb);
 getch();
 return 0;
}
```

```
1.sayiyi giriniz
30
2.sayiyi giriniz
140
3.sayiyi giriniz
35
4.sayiyi giriniz
20
Bu sayilarin en buyugu= 140
-----
Process exited after 11.12 seconds with return value 0
Press any key to continue . . .
```



## ÖRNEK 2:

Üç sayıyı küçükten büyüğe sıralayan programı yazınız.

```
#include<stdio.h>
#include<conio.h>
main()
{int x,y,z,t;
printf ("1.sayiyi giriniz\n" ); scanf("%d",&x);
printf ("2.sayiyi giriniz\n" ); scanf("%d",&y);
printf ("3.sayiyi giriniz\n" ); scanf("%d",&z);
if(x>y){t=x; x=y; y=t;}
if(x>z){t=x; x=z; z=t;}
if(y>z){t=y; y=z; z=t;}
printf("Bu sayilar  %d<%d<%d  şeklinde sıralanır",x,y,z);
getch();
return 0;
}
```

```
1.sayiyi giriniz
35
2.sayiyi giriniz
78
3.sayiyi giriniz
20
Bu sayilar  20<35<78  şeklinde sıralanır
-----
Process exited after 13.9 seconds with return value 0
Press any key to continue . . .
```

## Örnek 3:

### Program

```
#include<stdio.h>

#include<conio.h>

int main()
{ int a,b;

if (a < 10)
if (b < 10)
printf("1.kisi\n");
else
printf("2.kisi \n");
printf("3.kisi \n");

getch ();
return 0;
}
```

a=8 ve b=12 iken çıktı ne olur?

?

```
2.kisi
3.kisi
-----
```

a=12 ve b=8 iken çıktı ne olur?

?

```
3.kisi
-----
```

## Örnek 4:

### Program

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

int main()
{ int a,b;
  if (a < 10){
    if (b < 10)
      printf("1.kisi\n");}

  else{
    printf("2.kisi \n");
    printf("3.kisi \n");}
  getch ();
  return 0;
}
```

a=8 ve b=12 iken çıktı ne olur?

?

a=12 ve b=8 iken çıktı ne olur?

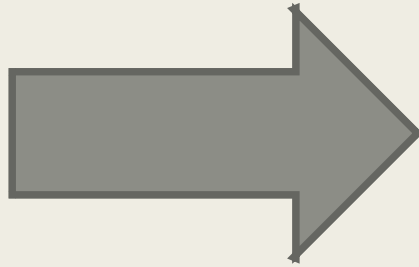
?

```
2.kisi
3.kisi
```

# if-else Kullanım Hataları

## Hata:

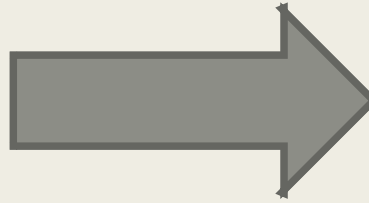
```
if a < 10
{
if (b < 10)
printf("1.kisi\n");
}
```



```
if (a < 10)
{
if (b < 10)
printf("1.kisi\n");
}
```

## Hata:

```
if (a < 10);  
printf("3.kisi \n");
```



a=15 için:

```
3.kisi
```

if (a < 10); komutundaki ; boş bir ifadedir.

## Hata:

`if (100<=x<=999) ...`

Sonuç her zaman doğrudur. Soldan sağa bakılır.

Girilen x değeri için  $(100 \leq x)$  değeri ya sıfır ya da bir olur.

Bu değerler daima 1000 den küçüktür ve  $(100 \leq x \leq 999)$  koşulu her zaman sağlanır.

## Koşul aşağıdaki şekilde olmalıdır:

`if ((x>=100) && (x<=999))...`

## **Hata:**

`if (x=y) ...`

atama işlemi yapar.

**Koşul aşağıdaki şekilde olmalıdır:**

`if (x==y)....`

# switch Çoklu Seçim Yapısı

## ■ switch

- *Bir değişken ya da bir ifadenin ayrı ayrı sabitlerle karşılaştırılması ve buna bağlı olarak farklı işlemlerin yapılması gibi bir dizi seçimin yer aldığı durumlarda kullanılır.*

### Format

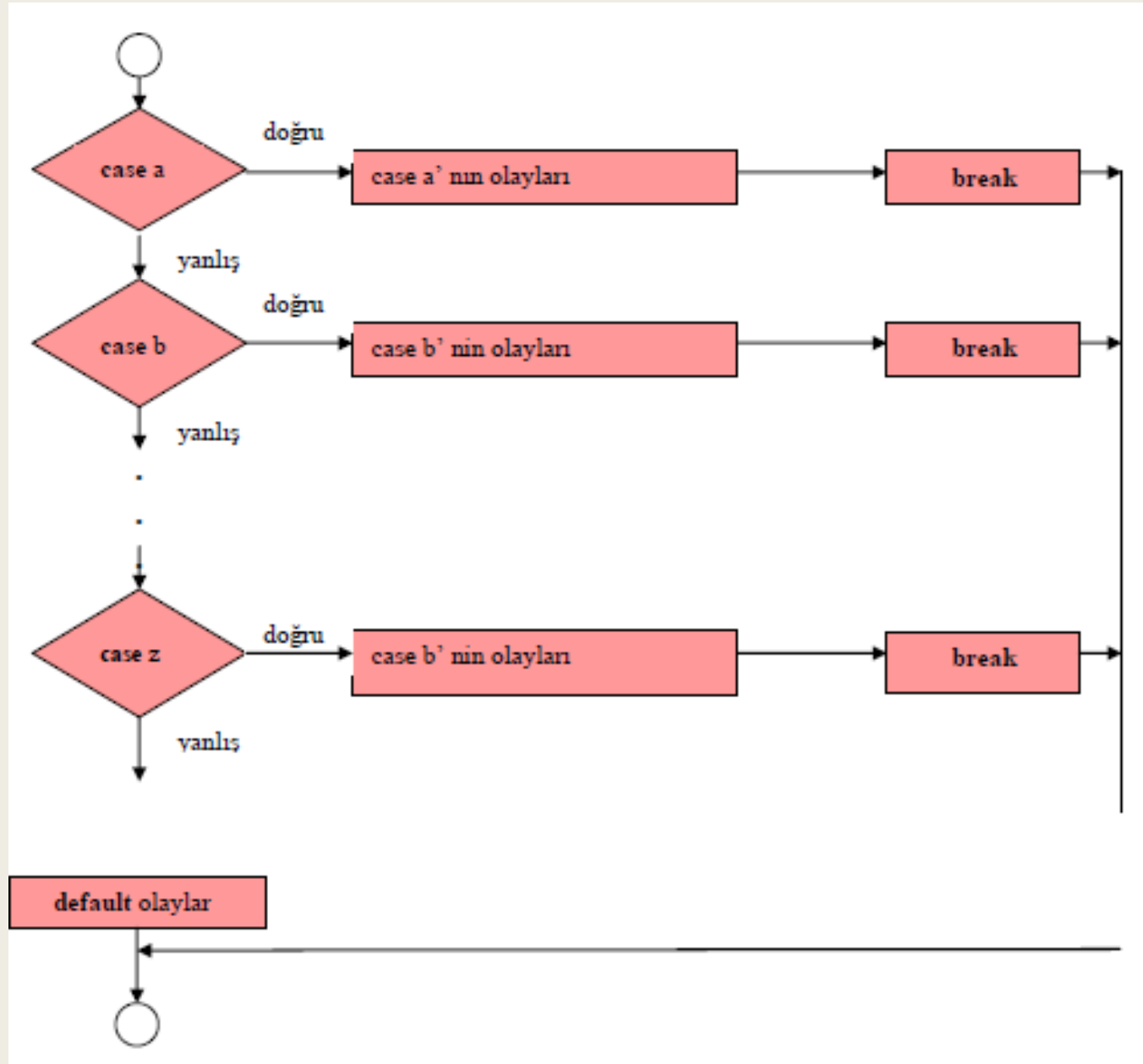
- *case ve tercihe bağlı default kısımlarında oluşur.*

```
switch ( ifade ){  
    case '1':  
        actions  
    case '2':  
        actions  
    default:  
        actions  
}
```

- *break; yapıdan çıkmak için*



# switch Çoklu Seçim Yapısı



**Örnek1:** Klavyeden girilen iki say ve girilen matematiksel işleme göre sonuç veren C programı yazınız.

```
#include <stdio.h>
#include <conio.h>
int main()
{
    char islem;
    int s1, s2, s3;
    printf("Yapilacak islemi giriniz:\n ");
    scanf("%c",&islem);
    printf("iki sayi giriniz:\n");
    scanf("%d%d", &s1, &s2);
    switch (islem)
    {
        case '+': s3 = s1 + s2; break;
        case '-': s3 = s1 - s2; break;
        case '*': s3 = s1 * s2; break;
        case '/': s3 = s1 / s2; break;
        default : printf ("Hatali islem");
    }
    printf("\n Sonuc = %d %c %d = %d",s1,islem,s2,s3);
    getch();
    return 0;
}
```

```
Yapilacak islemi giriniz:
*
iki sayi giriniz:
5
9
Sonuc = 5 * 9 = 45
-----
Process exited after 10.7 seconds with return value 0
Press any key to continue . . .
```

```
Yapilacak islemi giriniz:
-
iki sayi giriniz:
60
25
Sonuc = 60 - 25 = 35
-----
Process exited after 23.36 seconds with return value 0
Press any key to continue . . .
```

**Örnek2 :** Klavyeden girilen iki basamaklı bir sayıyı Türkçe olarak yazan C programını yazınız.

Girdi: 78

Çıktı: Yetmiş Sekiz

```
#include<stdio.h>
#include<conio.h>
int main()
{
    int sayi,a,b;
    printf("2 basamakli bir sayi giriniz: ");
    scanf("%d",&sayi); // sayımızı sayi değişkenine atadık.
    if((sayi>=10) && (sayi<100)) // Sayı 10 dan büyük 100 den küçükse aşağıdaki işlemler yapılacak.
    {
        a=sayi/10; // Onlar basamağındaki sayıyı belirledik.
        b=sayi-(a*10);
        printf("Sayınız: ");
        switch(a) // Onlar basamağı için switch case işlemleri.
        {
            case 1:printf("On"); break;
            case 2:printf("Yirmi"); break;
            case 3:printf("Otuz"); break;
            case 4:printf("Kirk"); break;
            case 5:printf("Elli"); break;
            case 6:printf("Altmis"); break;
            case 7:printf("Yetmis"); break;
            case 8:printf("Seksen"); break;
            case 9:printf("Doksan"); break;
            case 0:printf(" "); break;
        }
    }
}
```

switch(b) // Birler basamağı için switch case işlemleri.

```
{
case 1:printf(" bir");break;
case 2:printf(" iki"); break;
case 3:printf(" uc"); break;
case 4:printf(" dort"); break;
case 5:printf(" bes"); break;
case 6:printf(" alti"); break;
case 7:printf(" yedi"); break;
case 8:printf(" sekiz"); break;
case 9:printf(" dokuz"); break;
case 0:printf(" "); break;
} // Birler basamağındaki sayıyı belirledik.
```

```
}
else // Sayı iki basamaklı değilse program sonlanacak.
{printf("Lutfen iki basamakli bir sayi giriniz.\n");}
getch();
return 0;
}
```

```
2 basamakli bir sayi giriniz: 78
Sayiniz: Yetmis sekiz
-----
Process exited after 12.63 seconds with return value 0
Press any key to continue . . .
```

```
2 basamakli bir sayi giriniz: 5
Lutfen iki basamakli bir sayi giriniz.
-----
Process exited after 3.284 seconds with return value 0
Press any key to continue . . .
```

```
2 basamakli bir sayi giriniz: 280
Lutfen iki basamakli bir sayi giriniz.
-----
Process exited after 6.663 seconds with return value 0
Press any key to continue . . .
```

**Örnek 3:** Klavyeden 't' veya 'T' harfi girildiğinde 'T harfi girdiniz' aksi halde 'T harfi girmediniz'

şeklinde çıktı veren ve C programı yazınız.

```
#include <stdio.h>
#include <conio.h>
int main()
{
    char x;
    printf("Bir harf giriniz:\n");
    scanf("%c",&x);
    switch (x)
    {
        case 't' : printf ("T harfi girdiniz"); break;
        case 'T' : printf ("T harfi girdiniz"); break;
        default : printf ("T harfi girmediniz");
    }
    getch();
    return 0;
}
```

```
Bir harf giriniz:
T
T harfi girdiniz
-----
Process exited after 3.965 seconds with return value 0
Press any key to continue . . .
```

```
Bir harf giriniz:
t
T harfi girdiniz
-----
Process exited after 3.224 seconds with return value 0
Press any key to continue . . .
```

```
Bir harf giriniz:
a
T harfi girmediniz
-----
Process exited after 2.847 seconds with return value 0
Press any key to continue . . .
```

## switch-case Kullanım Hataları

- break deyiminin kullanılması unutulması.
- switch-case bloğundaki { } sembollerinin kullanılmalarının unutulması.
- default deyiminin ve içeriğinin kullanılmalarının unutulması.
- Kontrol edilen değişken ile verilen sabitlerin farklı tipte olması.
- int ve char veri türleri dışında değişken tanımlamaya çalışmak (double, string)



# if-else ve switch-case Karşılaştırılması

1. switch-case deyimi char ve int veri türlerini destekler. if-else de böyle bir durum yoktur.
2. switch-case deyiminde aralık kontrolü yapılamıyor. if-else de yapılabilir.  
     $x < 25$   
     $x < 50$
3. switch-case deyiminde performans genelde daha iyidir.
4. switch-case deyiminde kodu okumak if-else deyimine göre daha kolaydır.

# ? Karşılaştırma Operatörü

Bu operatör, if-else karşılaştırma deyiminin benzer şekilde çalışır, fakat kullanımı kısıtlıdır.

Kullanımı:

(koşul) ? deyim1 : deyim2;

- Eğer koşul olumluysa deyim1, olumsuzsa deyim2 değerlendirilir.
- deyim1 ve deyim2 de atama işlemi yapılamaz.
- Ancak koşul deyiminde atama işlemi yapılabilir.
- deyim1 ve deyim2 yerine fonksiyon da kullanılabilir.

## Örnek1:

```
#include <stdio.h>
#include <conio.h>
int main()
{
    int sayi = 15;
    ( sayi < 10 ) ? printf("1.kisi") : printf("2.kisi");

    getch();
    return 0;
}
```

```
2.kisi
-----
Process exited after 1.924 seconds with return value 0
Press any key to continue . . . _
```

```
#include <stdio.h>
#include <conio.h>
int main()
{
    int sayi = 5;
    ( sayi < 10 ) ? printf("1.kisi") : printf("2.kisi");

    getch();
    return 0;
}
```

```
1.kisi
-----
Process exited after 2.151 seconds with return value 0
Press any key to continue . . .
```

## Örnek 2:

```
#include <stdio.h>
#include <conio.h>
int main()
{
    int a,x=5,y=3;
    a=(x>y) ? x : y;
    printf ("sayinin degeri= %d", a);
    getch();
    return 0;
}
```

```
sayinin degeri= 5
-----
Process exited after 4.005 seconds with return value 0
Press any key to continue . . .
```

```
#include <stdio.h>
#include <conio.h>
int main()
{
    int a,x=3,y=5;
    a=(x>y) ? x : y;
    printf ("sayinin degeri= %d", a);
    getch();
    return 0;
}
```

```
sayinin degeri= 5
-----
Process exited after 2.093 seconds with return value 0
Press any key to continue . . .
```

# goto deyimi

Bir C programında akışın yönünü değiştirmek için kullanılan komuttur.

## Kullanımı:

etiket:

.....

.....

.....

goto etiket;

**Örnek 1:** 1 den 5'e kadar olan sayıları ekrana yazdıran C programını yazınız.

```
#include<stdio.h>
#include<conio.h>
int main()
{ int s=0;
  bas:
  s=s+1;
  printf ("%d\n",s);
  if (s<5) goto bas;
  getch ();
  return 0;
}
```

```
1
2
3
4
5
-----
Process exited after 1.79 seconds with return value 0
Press any key to continue . . .
```

**Örnek 2:** 1 den 10'a kadar olan sayıların toplamını ekrana yazdıran C programını yazınız.

```
#include<stdio.h>
#include<conio.h>
int main()
{ int s=0,top=0;
bas:
s=s+1;
top=top+s;
printf ("%d\n",s);
if (s<10) goto bas;
printf ("Sayilarin toplami= %d",top);
getch ();
return 0;
}
```

```
1
2
3
4
5
6
7
8
9
10
Sayilarin toplami= 55
-----
Process exited after 5.945 seconds with return value 0
Press any key to continue . . .
```

### Örnek 3:

Bir öğrencinin bir dersten aldığı vize ve final notları veriliyor. Bu notlara göre geçme notunu hesaplayan ( $\text{Geçme Notu} = \text{Vize} \cdot 0.4 + \text{Final} \cdot 0.6$ ), final ya da geçme notlarından birinin ya da ikisinin 50'den küçük olması durumunda bütünleme notunu isteyen ve buna göre geçme notunu hesaplayan, sonuçta geçme notunu ve GEÇTİ ya da KALDI şeklinde mesaj veren programı hazırlayınız.

```
#include<stdio.h>
#include<conio.h>
int main()
{ int vize,final,but;
float ort;
printf("Vize notunu giriniz: "); scanf("%d",&vize);
printf("Final notunu giriniz: "); scanf("%d",&final);
ort= vize*0.4 + final*0.6;
if((ort<50) || (final <50))
{ printf("Butunleme notunu giriniz: "); scanf("%d",&but);
ort= vize*0.4 + but*0.6;
if((ort<50) || (but <50))
{printf("Kaldi\n");
goto son;}
else { printf("Gecti\n");
printf("Gecme notu= %.3f \n",ort);
goto son;}
}
else printf("Gecti\n");
printf("Gecme notu= %.3f \n",ort);
son:
getch();
return 0;
}
```

```
Vize notunu giriniz: 45
Final notunu giriniz: 70
Gecti
Gecme notu= 60.000

-----
Process exited after 10.44 seconds with return value 0
Press any key to continue . . .
```

```
Vize notunu giriniz: 80
Final notunu giriniz: 45
Butunleme notunu giriniz: 50
Gecti
Gecme notu= 62.000

-----
Process exited after 9.514 seconds with return value 0
Press any key to continue . . .
```

```
Vize notunu giriniz: 45
Final notunu giriniz: 40
Butunleme notunu giriniz: 35
Kaldi

-----
Process exited after 9.223 seconds with return value 0
Press any key to continue . . .
```



# Kaynaklar

- C: How to Program Third Edition Harvey M. Deitel ; Paul J. Deitel.
- C Programlama Dili Dr. Rıfat Çölkesen Papatya Yayıncılık.
- Problem Solving and Program Design in C, 7/E Jeri R. Hanly; Elliot B. Koffman.
- C Programlama dili; İbrahim Güney; Nobel Yayıncılık.
- Algoritma Geliştirme ve Programlamaya Giriş, Fahri Vatansever, Seçkin yayıncılık
- C Programlama Ders Notları, A. Kadir YALDIR, Pamukkale Üniversitesi ders notları.