

Bilgisayar ve Programlamaya Giriş



Hafta 7

C Programlamaya Giriş (biçimli gösterim)

`%5c` → _____

`a='x';
printf("%5d",a);` → _____x

`%5d` → _____

`a=2;
printf("%5d",a);` → _____2

`%5.1f` → _____

`a=2.5;
printf("%5.1f",a);` → __ 2 . 5

```
/*biçimli gösterim (tamsayı) ve karakter*/  
#include <stdio.h>  
int main()  
{  
    char a='b';  
    int b=12345;  
  
    printf("%7c\n",a);  
  
    printf("%7d\n",b);  
    printf("%3d\n",b);  
    return 0;  
}
```

```
/*biçimli gösterim (ondalıklı)*/  
#include <stdio.h>  
int main()  
{  
    float a=12345.67;  
    printf("a=%7.2f\n",a);  
    printf("a=%8.2f\n",a);  
    printf("a=%8.1f\n",a);  
    printf("a=%8.3f\n",a);  
    printf("a=%9.2f\n",a);  
    printf("a=%9.5f\n",a);//kaçınmak lazım  
    return 0;  
}
```

Tip dönüştürme (casting)

İstemsiz dönüştürme (implicit casting)

int tipi değişkenler bellekte 4 byte, long tipi değişkenler ise bellekte 8 byte yer tutarlar.

```
int a = 43;
```

```
long b ;
```

```
b = a ;
```

atamasını yaparsak, a'nın bellekteki adresinde yer alan 43 sayısını b'nin bellekteki adresine taşımış oluruz. Bu taşımayı 4 byte'lık bir yerden 8 byte'lık bir adrese yaptığımız için, veri kaybı söz konusu değildir. Dolayısıyla, yukarıdaki veri tipi dönüştürme her zaman geçerlidir. Bu tür dönüştürme istemsiz (implicit) dönüştürme denir.

İstemli dönüştürme (explicit casting)

```
int c ;
```

```
long d = 43 ;
```

```
c = (int)d ;
```

atamasını yaparsak, d'nin bellekteki adresinde yer alan 43 sayısını c'nin bellekteki adresine taşımış oluruz. Bu taşımayı 8 byte'lık bir yerden 4 byte'lık bir adrese yaptığımız için derleyiciyi büyük adresteki veriyi küçük adrese taşımaya ikna ediyoruz. Bu tür dönüştürme istemli (explicit) dönüştürme denir.

Tip dönüştürme (casting)

İstemli dönüştürme (explicit casting)

```
int x;
```

```
x = 97;
```

```
printf("%c ", (char)x); /*int'ten char'a istemli dönüştürme*/
```

```
float a = 5.25;
```

```
int b = (int)a; /*float'tan int'e istemli dönüştürme. Burada b'nin değeri 5 olur*/
```

```
char c = 'A';
```

```
int x = (int)c; /* char'dan int'e istemli dönüştürme. x değeri 65 olur: A'nın ASCII karşılığı*/
```

İstemli dönüştürme (Explicit casting)

```
/* casting */
#include<stdio.h>
int main()
{
    int bolunen = 8, bolen = 5;
    float bolum;
    bolum = bolunen / bolen; // (float)
    // C dilinde tamsayının tamsayıya
    bölümü yine tamsayı çıkar
    // bölünen ya da bölen den birinin
    ondalıklı olması gerekir Explicit
    casting
    printf("Sonuc: %f\n",bolum);
    return 0;
}
```

```
/* casting */
#include<stdio.h>
int main( )
{
    printf("Sonuc: %f\n", 2 / 4);
    //printf("Sonuc: %f\n", (float)2 / 4);
    // explicit casting
    return 0;
}
```

```
/* casting */
#include<stdio.h>
int main( )
{
    char c;
    c=97; // explicit casting
    printf("%c\n", c);
    printf("%d\n", c);
    //geçen hafta görmüştük
    return 0;
}
```

Operatörler ve İşlem önceliği

İşlem	C aritmetik operatörü	Cebirsel İfade	C ifadesi
Toplama	+	f+7	f+7
Çıkarma	-	p-c	p-c
Çarpma	*	bm ya da b.m	b*m
Bölme	/	x/y ya da $\frac{x}{y}$	x/y
Mod	%	r mod s	r%s

Operatörler	İşlemler	İşlem önceliği
()	parantez	İlk değerlendirilir. (a*(b+c/d+e)) ifadesinde olduğu gibi iç içe parantezler varsa, en içteki parantezlerin önceliği vardır.
* / %	Çarpma Bölme Mod	İkinci sırada değerlendirilir. Eğer birden fazla işlem varsa soldan sağa doğru öncelik sırası vardır.
+ -	Toplama Çıkarma	Son sırada değerlendirilir. Eğer birden fazla işlem varsa soldan sağa doğru öncelik sırası vardır.

Operatörler ve İşlem önceliği

$z = p * r \% q + w / x - y;$



Öncelikle çarpma, mod alma ve bölme işlemleri soldan sağa olacak şekilde gerçekleşir, çünkü bunların toplama ve çıkarma işlemlerine göre yüksek öncelik sıraları bulunmaktadır. Toplama ve çıkarma işlemleri daha sonra yapılır. Bunlar da yine soldan sağa doğru gerçekleşir. Atama operatörü en son uygulanır çünkü diğer işlemlere göre daha alt seviye öncelik sırasına sahiptir.

Operatörler ve İşlem önceliği

```
*      8/4+2 => 2 + 2 => 4
*      8-4*2+-12 => 8 - 8 + -12 => -12
*      15*4/2%4*7 => 60/2%4*7 => 30%4*7 => 2*7 => 14
*      31+7/2-83%5*2-2 => 31+ 3 - 3 * 2 - 2 => 31 + 3 - 6 - 2 => 26
*      (31-7) * 2 + 83 / (5%2) => 24 * 2 + 83 / 1 => 48 + 83 => 131
```

```
/* işlem önceliği*/
#include <stdio.h>
int main()
{
    int sonuc1=31+7/2-83%5*2-2;
    int sonuc2=15*4/2%4*7;
    printf("\nsonuc1=%d ve\nsonuc2=%d dir.",sonuc1,sonuc2);
    return 0;
}
```

Operatörler ve İşlem önceliği

Cebirsel ilişkisel ya da eşitlik operatörü	C ilişkisel ya da eşitlik operatörü	C örnek koşulu	C koşulunun anlamı
<i>İlişkisel operatörler</i>			
>	>	x > y	x büyüktür y
<	<	x < y	x küçüktür y
≥	>=	x >= y	x büyük ya da eşittir y
≤	<=	x <= y	x küçük ya da eşittir y
<i>Eşitlik operatörleri</i>			
=	==	x == y	x eşittir y
≠	!=	x != y	x eşit değildir y

Operatörler

++ -- (artırma / eksiltme)

x=x+1; → ++x;

x=x-1; → --x;

x=100;

y=++x;

Önce x 101 olacak, sonra y 101 olacak

x=100;

y=x--;

önce y 100 olacak, sonra x 99 olacak

x=100;

y=x++;

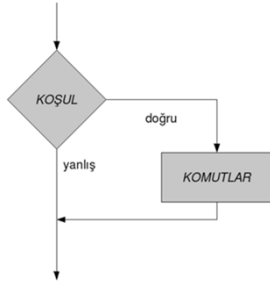
Önce y 100 olacak, sonra x 101 olacak

x=100;

y=--x;

Önce x 99 olacak, sonra y 99 olacak

if



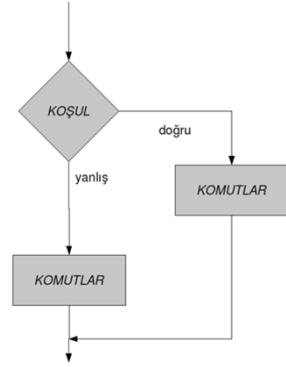
```
if (5>3)
    printf ("5 buyuktur");
```

if program içinde karar vermek için kullanılan en basit yapıdır.

If..Else deyimi bir Boolean ifadeyi (doğru/yanlış) değerlendirerek program denetimi belirtilen deyim gruplarına geçirir.

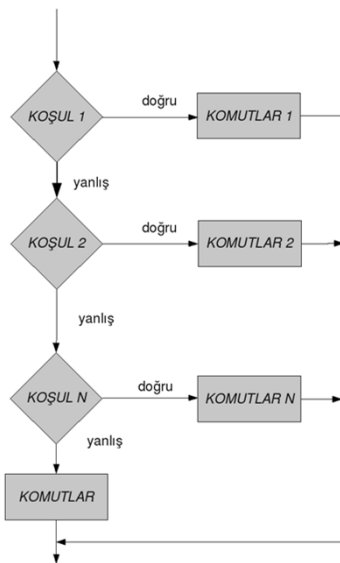
Bir koşulun doğruluğunun aksi durumunda ne yapacağımızı belirtmek istersek bunun için if-else yapısını kullanırız.

if else



```
if (a>b)
    printf ("a buyuktur");
else
    printf ("b buyuktur");
```

if + else if + (else)



```
if( girilen_yas < 18 )
    printf("cocuk\n");
else if( girilen_yas >= 18 && girilen_yas <= 30 )
    printf("Genc\n");
else if( girilen_yas > 30 && girilen_yas <= 50 )
    printf("olgun\n");
else if( girilen_yas > 50 && girilen_yas <= 65 )
    printf("cok olgun\n");
else
    printf("yaşlı\n");
```

if

```
/* pozitif / negatif */
#include<stdio.h>
int main( )
{
    int sayi;
    printf("Lütfen bir tam sayı giriniz> ");
    scanf("%d",&sayi);
    if( sayi > 0 )
        printf("Girilen sayı pozitifdir\n");
    else if ( sayi < 0 )
        printf("Girilen sayı negatiftir\n");
    else
        printf("Girilen sayı sıfırdır\n");
    return 0;
}
```

```
/* harf puanı hesaplama */
#include <stdio.h>
int main()
{
    int puan;
    printf("puani giriniz ");
    scanf("%d",&puan);

    if(puan >= 90)
        printf("A");
    else if ( puan >=80)
        printf("B");
    else if ( puan >=70)
        printf("C");
    else if ( puan >=50)
        printf("D");
    else
        printf("F");

    return 0;
}
```

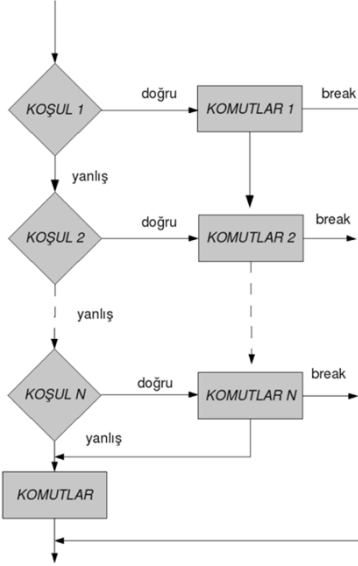
if

if (kosul)
 x=y;
else
 x=z;

} x = (kosul) ? y : z ;

```
/* if */
#include<stdio.h>
int main( )
{
    int sayi1,sayi2,buyuk;
    printf("Lütfen ilk sayiyi giriniz> ");
    scanf("%d",&sayi1);
    printf("Lütfen ikinci sayiyi giriniz> ");
    scanf("%d",&sayi2);
    buyuk = (sayi1>sayi2)?sayi1:sayi2;
    printf("Buyuk sayi : %d dir\n",buyuk);
    return 0;
}
```

switch-case



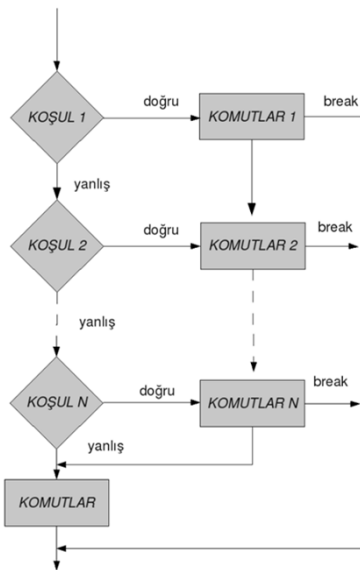
switch – case seçme yapısı bir değişkenin içeriğine bakarak programın akışını birçok seçenektan birisine yönlendiren bir karşılaştırma deyimidir.

Değişkenin içeriği hangi sabit ile uyuşursa ona ait işlem kümesi ve arkasındaki bütün işlem kümeleri yürütülür. Ancak küme deyimleri arasında **break** kullanılırsa, daha sonraki tüm işlem kümeleri atlanarak *switch* bloğunun sonuna gidilir.

switch-case

vs

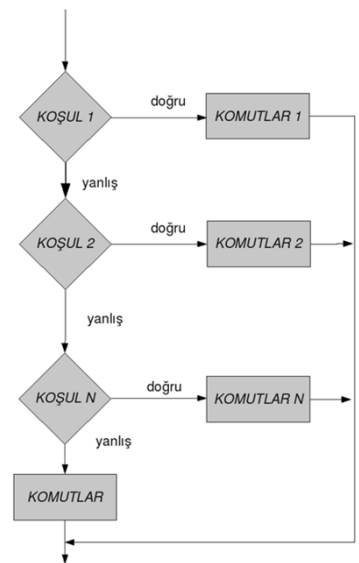
if + else if + (else)



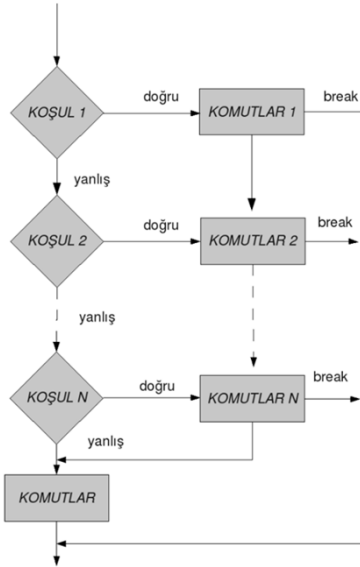
If program içinde karar vermek için kullanılan en basit yapıdır.

If..Else deyimi bir Boolean ifadeyi (doğru/yanlış) değerlendirerek yolunu belirler.

Switch deyimi de if deyimi gibi programın koşullara göre yönelmesini sağlar. Özellikle bir değişkenin aldığı birçok farklı değere göre yapılacak farklı işler varsa if yerine switch-case kullanılabilir.



switch-case



```
#include <stdio.h>
int main()
{
    char a;

    printf("1-5 arasi bir rakam giriniz: ");
    scanf("%c",&a);

    switch (a) {
        case '1': printf("cok zayif \n"); break;
        case '2': printf("zayif \n"); break;
        case '3': printf("orta \n"); break;
        case '4': printf("iyi \n"); break;
        case '5': printf("pekiyi \n"); break;
        default: printf("yanlis secim \n");
    }
    return 0;
}
```

switch

```
/* kodu oynayalım */
#include <stdio.h>
int main()
{
    int a=2;
    switch(a)
    {
        case 1:
            printf("X"); //break;
        case 2:
            printf("X"); //break;
        case 3:
            printf("X"); //break;
        default :
            printf("X"); //break;
    }
    return 0;
}
```

```
/* harf puanı hesaplama */
#include<stdio.h>
int main( )
{ int not;
    printf("Lütfen notu giriniz> ");   scanf("%d",&not);
    switch( not / 10 ) {
        case 10:
        case 9: printf("A\n"); break;
        case 8: printf("B\n"); break;
        case 7: printf("C\n"); break;
        case 6: printf("D\n"); break;
        case 5:
        case 4:
        case 3:
        case 2:
        case 1:
        case 0: printf("NOT: F\n"); break;
        default:
            printf("Notu tekrar giriniz!\n");
    }
    return 0;}
```