

1. Birleşik (Compound) Operatörler

Bazı durumlarda, kontrol edeceğimiz koşul, tek bir parametreye bağlı değildir. Örneğin, bir kişinin yaşının 65'den küçük olup olmadığına bakabiliriz. Ama 65'den küçük ve 18 yaşından büyük olup olmadığına karar vermek istersek, o zaman Birleşik/Birleştirici Operatörleri kullanmamız uygun olacaktır. Birleşik operatör'ler aşağıdaki gibidir:

&&	and	ve
	or	veya
!	not	tersi

Bu operatörlerin mantıksal (logical) doğruluk tablosu da şu şekildedir:

p	q	p && q	p q	!p
0	0	0	0	1
0	1	0	1	1
1	0	0	1	0
1	1	1	1	0

2. Klavyeden alınacak bir değer, 18 ile 65 arasında olup olmadığını kontrol eden bir program yazınız:

```
/*Girilen yaşın, 18 ile 65 arasında olup olmadığını belirler.*/
```

```
#include<stdio.h>
int main( void )
{
```

```
    int girilen_yas;
    printf("Lütfen yaşınızı giriniz> ");
    scanf("%d",&girilen_yas);
```

```
    if( girilen_yas >= 18 && girilen_yas <= 65 )
        printf("Girilen yaş, 18 ile 65 arasındadır.\n");
```

```
//Girilen yaş 18 ile 65 arasında değilse, aşağıdaki else bloğu çalışır.
else {
```

```
    //Girilen yaş 18'den küçükse
    if( girilen_yas < 18 )
        printf("Girilen yaş, 18'den küçüktür.\n");
    //Girilen yaş 65'ten büyükse
    else
        printf("Girilen yaş, 65'ten büyüktür.\n");
}
```

3. Klavyeden girilen iki tam sayıdan, 2. sayı 1. sayıyı bölecek ve sonucu virgüllü sayı olarak gösterecek bir bölme işleminin programını hazırlayınız. (Şayet bölen 0 olarak verilirse, bölme işlemi yapılmamalıdır.)

4. Klavyeden bir sayı girildiğinde şayet sayı 18'den küçükse çocuk yaşı; 18-30 arasında genç yaşı; 30-50 arasında orta yaşı diye bir mesaj yazdıran programı yazınız.

Şimdi bunu sadece if yapısıyla kuruyor olsaydık, her seferinde yaşın uygun aralıklara düşüp düşmediğini kontrol eder ve ona göre sonucu ekrana bastırırdık. Ama bu son derece verimsiz bir yöntem olurdu. Çünkü zaten yaş bilgisinin genç olduğuna dair bir karar vermişsek, sonrasında tutup bunun yaşlı olup olmadığını kontrol etmenin bir anlamı olmayacaktır. Verilebilecek en kötü cevabı aşağıda bulabilirsiniz:

```
/*Sorulan soruya verilebilecek en kötü cevap.*/
```

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

```
int main( void )
```

```
{
    int girilen_yas;
    printf("Lütfen yaşınızı giriniz> ");
    scanf("%d",&girilen_yas);
    if( girilen_yas < 18 )
        printf("Daha çocuk yaştasınız, hayatın başındasınız.\n");
    if( girilen_yas >= 18 && girilen_yas <= 30 )
        printf("Gençliğin, güzelliği bambaşka!\n");
    if( girilen_yas > 30 && girilen_yas <= 50 )
        printf("Hepsini boş verin, olgunluk orta yaşta başlar!\n");

    return 0;
}
```

Yukarıdaki kodu if - else kullanarak daha etkili hale getirebiliriz

```
/*if - else yapısıyla daha efektif bir yapı*/
```

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

```
int main( void )
```

```
{
    int girilen_yas;
    printf("Lütfen yaşınızı giriniz> ");
    scanf("%d",&girilen_yas);

    if( girilen_yas < 18 )
        printf("Daha çocuk yaştasınız, hayatın başındasınız.\n");
    else if( girilen_yas >= 18 && girilen_yas <= 30 )
        printf("Gençliğin, güzelliği bambaşka!\n");
    else if( girilen_yas > 30 && girilen_yas <= 50 )
        printf("Hepsini boşverin, olgunluk ortayaşta başlar!\n");
    else
        printf("HATA: Girilen yaş tanımlı değildir!\n");

    return 0;
}
```

5. Verilecek üç sayıdan en büyüğünü ekrana yazdıracak bir program yazınız.

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

```
int main( void )
```

```
{
    int sayi_1, sayi_2, sayi_3;
    printf("Üç sayı giriniz> ");
    scanf("%d %d %d", &sayi_1, &sayi_2, &sayi_3);
    if( sayi_1 >= sayi_2 && sayi_1 > sayi_3 )
        printf("%d en büyük sayıdır!\n", sayi_1);
}
```

```

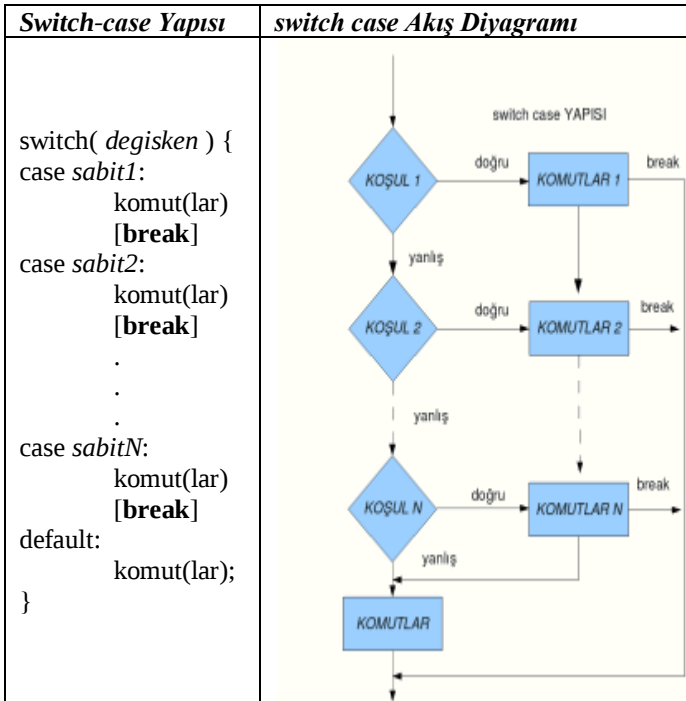
else if( sayi_2 >= sayi_1 && sayi_2 >= sayi_3 )
    printf("%d en büyük sayıdır!\n", sayi_2);
else if( sayi_3 >= sayi_1 && sayi_3 >= sayi_1 )
    printf("%d en büyük sayıdır!\n", sayi_3);

return 0;
}

```

6. swith - case ifadesi

switch - case, if - else if yapısına oldukça benzer bir ifadedir. Ancak aralarında iki fark vardır. Birincisi, switch - case yapısında, aralık değeri girmezsiniz. Direkt olarak ifadelerin bir şeylere eşit olup olmadığına bakarsınız. İkinci farksa, switch - case yapısında, illa ki uygun koşulun sağlanmasıyla yapının kesilmek zorunda olmayışdır. 'break' komutu kullanmadığınız takdirde, diğer şartların içindeki işlemleri de yapma imkanınız olabilir.



Bir not değerlendirme sistemi olsun. 100 - 90 arası A, 89 - 80 arası B, 79 - 70 arası C, 69 - 60 arası D, 59 ve altıysa F olsun. Eğer 100'den büyük veya negatif bir sayı girilirse, o zaman program hatalı bir giriş yapıldığını konusunda bizleri uyarın.

```

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

```

```

printf("HATA: Bilinmeyen bir değer girdiniz!\n");
    }
    return 0;
}

```

7. Arttırma (Increment) ve azaltma (decrement) işlemleri

x=y++ : y'nin değeri önce x'e atanır sonra da 1 arttırılır.
 x=y; y=y+1;
 x=++y : y'nin değeri önce 1 arttırılır sonra da x'e atanır .
 y=y+1; x=y;
 x=y-- : y'nin değeri önce x'e atanır sonra da 1 azaltılır.
 x=y; y=y-1;
 x=--y : y'nin değeri önce 1 azaltılır sonra da x'e atanır .
 y=y-1; x=y;

```

/*Bu programda arttırma, azaltma işlemleri önce yapılacaktır. */
#include<stdio.h>
int main( void )
{
    int i = 10, j = 60;
    printf("i = %d ve j = %d\n", ++i, --j);
    return 0;
}

```

Çıktı: i = 11 ve j = 59

Çünkü arttırma ve azaltma işlemleri ekrana bastırmadan önce yapılmış ve i ile j'nin değerleri değiştirilmiştir.

Şimdi programı değiştirip şöyle yazalım:

```

/*Bu programda arttırma, azaltma işlemleri sonra yapılacaktır. */

```

```

#include<stdio.h>
int main( void )
{
    int i = 10, j = 60;
    printf("i = %d ve j = %d\n", i++, j--);
    return 0;
}

```

Çıktı: i = 10 ve j = 60

8. While döngüsü ile girilen sayılarda tek ve çiftlerin kaç tane olduğunu bulmak.

```

#include <stdio.h>
void main(){

```

```

    int cift=0,tek=0,n,a;

```

```

while(n!=-1){ // -1 girilirse döngüden çık.
    printf("Bir sayı giriniz: \n");
    scanf("%d",&n);

    a=n%2;
    if(a==0){
        cift=cift+1;
    }//if

    if(a==1){
        tek=tek+1;
    }//if
} //while

```

```

printf("%d tane cift sayi vardır\n",cift);
printf("%d tane tek sayi vardır\n",tek);}

```