

BÖLÜM 2

C PROGRAMLAMADA AKIŞ KONTROLÜ

GOTO: C programında programın herhangi bir yerinden bir yerine **şartsız** olarak atlanmasını sağlayan komuttur.

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

```
main () {    int a=5 ;
```

```
    dal1:
```

```
    printf ("\n %d",a ) ;
```

```
    a+= 2;
```

```
    goto dal1 ;
```

IF-ELSE Yapıları: Mantıksal bir ifadenin doğru yada yanlış olmasına göre farklı yerlere dallanmayı sağlar.

*** If Yapısı:**

if (koşul) işlem 1 ;
Y-D → işlem 2 ;

if (koşul) { işlemler;
Y-D → }
↓

Örn 2.1

```
#include <stdio.h>
main(){
int a=2,b=4;
    if( a>5) printf (“\n %d “,a) ;
    printf (“\n %d - %d “,a,b ) ; }
```

If - Else Yapısı

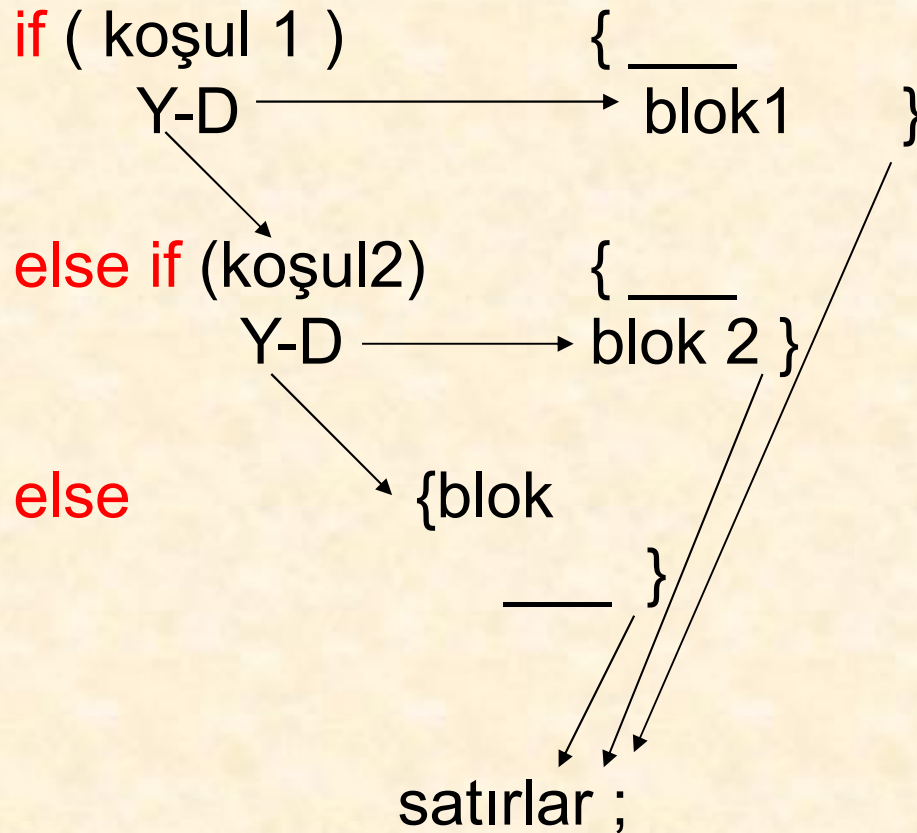
if (Koşul) işlem1 ;
 Y-D →
else işlem2 ;

if (Koşul) { blok1 }
 Y-D →
else { ____ }

Örnek 2.2

```
#include <stdio.h>
main(){
int a,b;
scanf("%d,%d",&a,&b);
if (a>b) printf ("ilk girilen sayı büyük");
else printf ("ikinci girilen sayı büyük");
}
```

If – Elseif – Else Yapısı



Örnek 2.3

```
# include <stdio.h>
int a,b,c, min, max, ort ;

main () { scanf ("%d, %d, %d", &a ,&b,&c) ;
if (a<b) if (a<c) if (b<c) { min =a ; ort=b ; max=c ;}
                else {min =a ; ort=c; max=b ;}
        else {min=c; ort=a; max=b;}

else if (b<c) if (a<c) {min=b; ort=a; max=c;}
                else {min=b; ort=c; max=a;}

else {min=c; ort=b; max=a;}

printf ("\n %d %d %d ",min,ort,max) ; }
```

Örnek 2.4

// 1 den 10 e kadar olan sayıların toplamı

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

```
main( ) {
```

```
    int i,toplam;
```

```
    toplam=0; i=0;
```

etiket:

```
    i=i+1;
```

```
    toplam=toplam+i;
```

```
    if (i<10) goto etiket;
```

```
    printf( "toplam=%d\n",toplam);
```

```
}
```


Örnek 2.5

/* Girilen 3 sayıdan büyüğünü bulan program */

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

```
main() {  
    int a,b,c,buyuk;  
    printf("a sayısını giriniz:"); scanf("%d",&a);  
    printf("b sayısını giriniz:"); scanf("%d",&b);  
    printf("c sayısını giriniz:"); scanf("%d",&c);  
    buyuk=a;  
  
    if(buyuk<b) buyuk=b;  
    if(buyuk<c) buyuk=c;  
  
    printf ("büyük sayı=%d\n",buyuk);  
}
```

Örnek 2.6

/* Girilen sayının tek mi? Çift mi? Olduğunu bulan program*/

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

```
main() {  
    int a,b;  
    printf("a sayısını gir:"); scanf("%d",&a);  
    b=(a/2) * 2;  
    if (a==b) printf("sayi çift");  
    else     printf("sayi tek");  
}
```

Çözüm 1

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

```
main() {  
    int a,b;  
    printf("a sayısını gir:"); scanf("%d",&a);  
    b=(a%2);  
    if (b==0) printf("sayi çift");  
    else     printf("sayi tek");  
}
```

Çözüm 2

FOR Döngüsü

Tek bir işlem satırını veya birden fazla işlem satırından oluşan bloğun istenen sayıda tekrarlanması için kullanılır.

Kull. Şek. : `for (ilk değer; şart; arttırma) işlem satırı;`

Örn:

```
for (i=0 ; i<10 ; i++) printf (“\n %d”,i) ;
```

```
for (i=0, j=5 ; i<= j ; i++, j--) { a=i+j ;  
                                printf (“\n %d”, a) ; }
```

```
for ( ; ; ) sonsuz döngü
```

Örnek 2.7

```
/* For döngüsü */  
#include <stdio .h>  
  
main( ) {  
    int i,toplam;  
  
    toplam=0;  
  
    for (i=1;i<=100;i++)  
  
        toplam=toplam+i;  
  
    printf( "toplam=%d\n",toplam);  
  
}
```

Örnek 2.8

```
/* İç İç For döngüleri
```

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

```
int x,i,j ;
```

```
main () { for (x=1 ; x<2 ; x++) {
```

```
    for ( j=1 ; j<=4 ; j++) {
```

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

```
    }
```

```
 }
```

```
}
```

Örnek 2.9

/*Rasgele girilen 10 sayıdan negatif ve pozitif olanların sayısını bulma

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

```
int i,neg,poz,a;
```

```
main(){
```

```
neg=0;poz=0;
```

```
    for (i=1;i<11;i++) { printf("a Sayısını Giriniz");
```

```
        scanf("%d",&a);
```

```
        if (a<0) neg=neg+1;    // neg++;
```

```
        if (a>0) poz=poz+1;    // poz++;
```

```
    }
```

```
    printf("Negatif sayılar Top. = %d \nPozitif Sayılar Top.=%d",neg,poz);
```

```
    getch();
```

```
}
```

Örnek 2.10

/*Arka arkaya girilen rasgele 10 tamsayının ortalaması ile bu sayılardan en büyük ve en küçük olanının ortalamasını bularak elde edilen bu iki ortalamanın farkını alan*/

```
# include <stdio.h>
# include <math.h>
int i,a,ek,eb,tp;
float b,fark,ort;
main(){
printf("a sayısını gir.:""); scanf( "%d",&a);
eb=a; ek=a, tp=a;
    for (i=2;i<11; i++) {
        printf("a sayısını gir..."); scanf("%d",&a);
        if(ek>a) ek=a; if(eb<a) eb=a;
        tp=tp+a;
    }

b=(ek+eb)/2; ort=tp/10; fark=ort-b;
printf("fark=%f",fark);
}
```


Örnek 2.11

/*Ondalıklı kısmı iki haneli girilen pozitif bir rasyonel sayının ondalıklı kısmının sayı değerinin bir tam kare olup olmadığını bulma

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

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

```
#include <math.h>
```

```
float a,k;
```

```
int b,c,sayi,deger;
```

```
main() {
```

```
    scanf( "%f",&a);
```

```
    b=a;
```

```
    k=a-b;
```

```
    sayi=k*100;
```

```
    deger=sqrt(deger);
```

```
    printf("%f %d %f %d %d",a,b,k,sayi,deger);
```

```
    deger=deger*deger;
```

```
    if(sayi==deger) printf("\nbu sayi tam karedir..");
```

```
    else printf("\nbu sayi tamkare degildir..");
```

```
    getch();
```

```
}
```


While döngüsü

while (mantıksal ifade) { işlem satırı }

ifade ile belirtilen şart doğru olduğu sürece blok içerisindeki satırlar tekrarlanır şart yanlış olduğu anda döngüden çıkar.

```
while( Koşul) { _____  
    y, d _____  
    _____ }  
    _____  
    _____
```

```
while ( n) { _____  
    _____ }
```

n=0 olduğu anda sonuç yanlıştır ve döngüden çıkar.
n<>0 iken döngü çalışır.

```
while (1) { _____ }
```

sonsuz döngü

Örnek 2.12

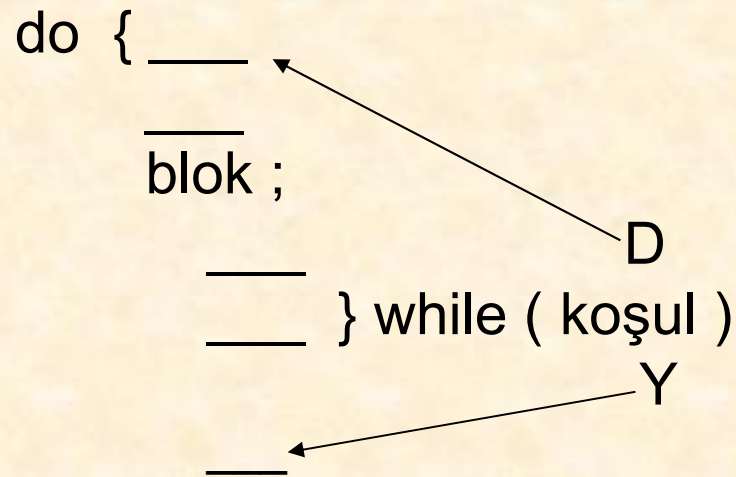
```
# include <stdio.h>
# include <conio.h>
int i ;

main () {
    i=1;
    while (i<20) { printf ("\ni=%d",i) ;
                    i++; }

    printf("i'nin değeri=%d",i);
    getch();
}
```

Do – While Döngüsü

While döngüsü gibidir. Ancak do-while döngüsünde önce program direkt olarak döngüye girer ve satırları çalıştırır. Daha sonra koşula bakılır. Koşul doğru olduğu sürece döngü bloğu tekrarlanır.



Örnek 2.13

```
# include <stdio.h>
# include <stdio.h>
int i ;

main () {
i=1;
    do { printf ("\ni=%d",i) ;
        i++; } while (i<20);

    printf("i'nin değeri=%d",i);
    getch();
}
```

Örnek 2.14

```
# include <stdio.h>
int x,y ;

main () {
clrscr () ;
    do { printf("sayi giriniz=");
        scanf("%d ",&x) ;
        y = (x<1|| x>9);
        if ( y ) printf ("\n hatalı seçim") ;
    } while (y);
printf ("\n dogru seçim ") ;
getche () ; }
```

Switch-Case Yapısı

Char ya da int tipi değişkenlerin değerlerine bağlı olarak programın işleyişini dallandıran (yapıdır.) bir kalıptır.

```
switch (ifade) { case etiket: blok1;  
                case etiket: blok2;  
                .  
                .  
                default: blokn;}
```

Örnek 2.15

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

main() { int d1;
printf ( "Bir Sayı Giriniz [1-5]");
scanf ( "%d", &d1);
switch (d1) { case 1: printf ("Bir");break;
              case 2: printf ("İki");break;
              case 3: printf ("Üç");break;
              case 4: printf ("Dört");break;
              case 5: printf ("Beş");break;
              }

getch();
}
```

break ve continue

for, while, do-while gibi döngü içlerinde kullanılan komutlardır. Break döngüden çıkılmasını sağlarken continue ise döngü bloğuna ait sonraki bölümü atlayarak döngü sonuna gidilmesini, dolayısı ile döngü şartının kontrol edilmesini sağlar.

Örnek 2.16

```
while (1) { scanf ("%d",&x);  
            if (x<0) break;  
            printf ("sayi=%d",x);  
  
            }
```

Örnek 2.17

```
for (i = 0; i<100; i++){  
    e = getchar ();  
    if ('0'<=c && c<='9')  
        continue;  
    putchar (c);  
}
```

Örnek 2.18

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

```
# include <math.h>
```

```
float a,b,c,x1,x2,d ;
```

```
char y ;
```

```
main () { scanf ("%f,%f,%f",&a,&b,&c) ;
```

```
    d=b*b-4*a*c ;
```

```
    y= ( d<0.0)*1 + (d== 0.0 )*2 + (d>0.0)*3 ;
```

```
    switch (y){ case 1:    printf ("\n gerçek kök yok") ;  
                break ;
```

```
                case 2 :    x1 = -b /2(*a) ;  
                printf ("\n kök=%f",x1) ; break ;
```

```
                case 3 :    x1 = (-b -sqrt(d)) /(2*a) ;  
                x2 = (-b + sqrt(d)) /(2*a) ;  
                printf ("\n kökler=%f,%f",x1,x2); }
```

```
getche () ; }
```

Örnek 2.19

```
/* İki sayının okek'ini bulan program*/
# include <stdio.h>
# include <conio.h>
main() {
    clrscr();
    int a,b,buyuk,i;
    printf("1.sayiyi giriniz:");
    scanf("%d",&a);
    printf("2.sayiyi giriniz:");
    scanf("%d",&b);
    if(a>b) buyuk=a;
    else buyuk=b;
    for(i=buyuk; ;i++)
        if(i%a==0 && i%b==0) {
            printf("okek=%d",i);
            break;    }
    getch();
}
```

Örnek 2.20

// İki sayının obebini bulan program

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

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

```
main() { clrscr();
```

```
    int a,b,x,i;
```

```
    printf("1.sayiyi giriniz:");
```

```
    scanf("%d",&a);
```

```
    printf("2. sayiyi giriniz:");
```

```
    scanf("%d",&b);
```

```
    if(a<b) x=a;
```

```
    else x=b;
```

```
    for(i=x;;i--)
```

```
        if(a%i==0 && b%i==0){
```

```
            printf("obeb=%d",i);
```

```
            break; }
```

```
getche();
```

```
}
```