Döngüler

Döngü (loop) deyimleri, bir kümenin belli bir koşul altında tekrar edilmesi için kullanılır. C programlama dilinde, while, do...while ve for olmak üzere üç tip döngü deyimi vardır. Diğer programlama dillerinde olduğu gibi, bu deyimlerle istenildiği kadar iç-içe döngü yapısı kullanılabilir.

While Döngüsü

Tekrarlama deyimidir. Bir küme ya da deyim while kullanılarak bir çok kez yinelenebilir. Yinelenmesi için koşul sınaması döngüye girilmeden yapılır. Koşul olumlu olduğu sürece çevrim yinelenir. Bu deyimin kullanımı aşağıda gösterilmiştir. Genel yazım biçimi:

```
while(koşul)
{
 ...
  döngüdeki deyimler; [küme]
 ...
}
```

Program içerisinde ise:

```
01: /* 07prg01.c: while döngüsü */
02:
03: #include <stdio.h>
04:
05: main()
06: {
07:    int x=0;
08:
09:    while(x <= 10)
10:       printf("%d\n",x++);
11:
12:  return 0;
13: }
```

do ... while Döngüsü

Bu deyimin while deyiminden farkı, koşulun döngü sonunda sınanmasıdır. Yani koşul sınanmadan döngüye girilir ve döngü kümesi en az bir kez yürütülür. Koşul olumsuz ise döngüden sonraki satıra geçilir. Bu deyimin kullanımı aşağıda gösterilmiştir. Genel yazım biçimi:

```
do{
   ...
   döngüdeki deyimler;
   ...
}while(koşul);
```

```
01: /* 07prg02.c: do-while yapısı */
02:
03: #include <stdio.h>
04:
05: main()
06: {
07:    int sayi;
08:
09:    do
10:    {
11:       printf("Bir sayi girin : ");
12:       scanf("%d",&sayi);
13:       printf("iki kati       : %d\n",2*sayi);
14:
15:    }while( sayi>0 );   /* koşul */
16:
17:    puts("Döngü sona erdi.");
18:
19:  return 0;
20: }
```

for Döngüsü

Bu deyim, diğer döngü deyimleri gibi bir kümeyi bir çok kez tekrarlamak için kullanılır. Koşul sınaması while da olduğu gibi döngüye girmeden yapılır. Bu döngü deyimin içinde diğerlerinden farklı olarak başlangıç değeri ve döngü sayacına sahip olmasıdır. Bu deyimin kullanımı aşağıda de gösterilmiştir Genel yazım biçimi:

```
for( başlangıç ; koşul ; artım )
{
 ...
  döngüdeki deyimler;
 ...
}
```

```c
/* 07prg03.c: for döngüsü ile faktoriyel hesabı.
*/

#include <stdio.h>

int main()
{
   long  i, n, faktor;

   printf("Faktoriyeli hesaplanacak sayi girin :
");
   scanf("%ld",&n);

   faktor=1;
   for(i=1; i<=n; i++){
         faktor *= i;     /* n! = 1 x 2 x 3 x
... x n */
   }

   printf("%ld! = %ld\n", n, faktor);
```

İç içe Geçmiş Döngüler

Bir program içinde birbiri içine geçmiş birden çok döngü de kullanılabilir. Bu durumda (bütün programlama dillerinde olduğu gibi) önce içteki döngü, daha sonra dıştaki döngü icra edilir.

Üç basamaklı, basamaklarının küpleri toplamı kendisine eşit olan tam sayılara Armstrong sayı denir. Örneğin: 371 bir Armstrong sayıdır çünkü 3^3 + 7^3 + 1^3 = 371. Program 7.5'de iç içe geçmiş üç for döngüsü ile bütün Armstrong sayıları bulup ekrana yazar.

```c
#include <stdio.h>

int main()
{
   int a,b,c, kup, sayi, k=1;

   for(a=1; a<=9; a++)
   for(b=0; b<=9; b++)
   for(c=0; c<=9; c++)
   {
       sayi = 100*a + 10*b + c;       /* sayi = abc (üç
basamaklı) */
       kup  = a*a*a + b*b*b + c*c*c;  /* kup  =
a^3+b^3+c^3        */

       if( sayi==kup ) printf("%d. %d\n",k++,sayi);
   }

return 0;
}
```

Sonsuz Döngü

Bir döngü işlemini sonsuz kere tekrarlarsa bu döngü sonsuz döngü olarak adlandırılır. Böyle bir döngü için, koşul çok önemlidir.

```
...
while(1)
{
   printf("Sonsuz döngü içindeyim...\n");
}
...

...
while(7>3)
{
   printf("Sonsuz döngü içindeyim...\n");
}
...
```

Her iki durumda da çevrimler, sonsuz döngü durumundadır. Çünkü while(1) ve while(7>3) ifadelerdeki koşullar daima olumludur. Bu durumda çevrim sonsuz döngüye girer.

break Deyimi

Bir C programında, bir işlem gerçekleştirilirken, işlemin sona erdirilmesi bu deyim ile yapılır. Örneğin, döngü deyimleri içindekiler yürütülürken, çevrimin, koşuldan bağımsız kesin olarak sonlanması gerektiğinde bu deyim kullanılır.

continue Deyimi

Bir döngü içerisinde continue deyimi ile karşılaşılırsa, ondan sonra gelen deyimler atlanır ve döngü bir sonraki çevrime girer.