

CENG 111 ALGORİTMALAR VE PROGRAMLAMA

Doç. Dr. Tufan TURACI

tturaci@pau.edu.tr

- Pamukkale Üniversitesi
- Mühendislik Fakültesi
- Bilgisayar Mühendisliği Bölümü
- Hafta 6
- 25 Ekim 2022

DÖNGÜLER

- Döngü nedir?
- for Döngü Yapısı
- while Döngü Yapısı
- do-while Döngü Yapısı
- break Deyimi
- continue Deyimi
- İç İç Döngüler
- Sonsuz Döngüler

Döngü

- Programlamada en çok kullanılan yapılardan biri döngülerdir. Döngüler aracılığı ile belli şartlar gerçekleşinceye kadar veya gerçekleştikçe istenilen komut bloğunu çalıştırmak mümkündür.
- Döngü deyimleri, bir işlemi yerine getiren kod bloğunun belli bir koşul altında tekrar edilmesi için kullanılır.
- Bir değişkenin belirli bir değerden başlayıp, son değeri alıncaya kadar belirtilen işlemleri tekrarlaması döngü yapısına örnektir.

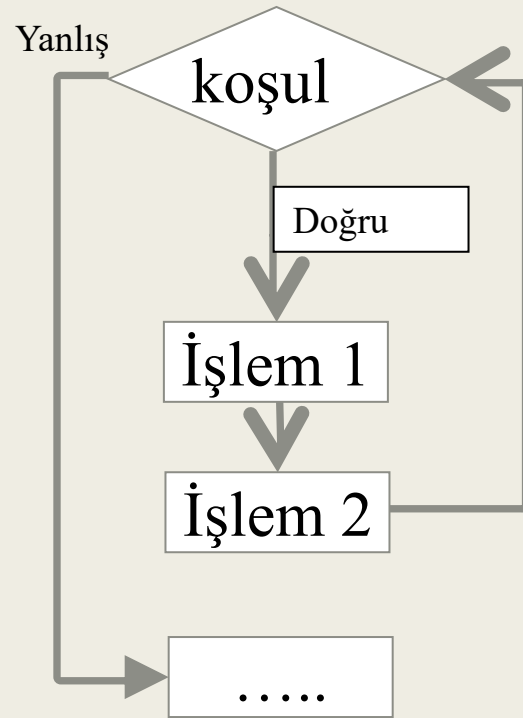
$n!$ hesaplanırken sayıların 1 den n 'ye kadar çarpılması.

- C programlama dilinde birden fazla döngü çeşidi bulunmaktadır.
- Döngü çeşitleri kullanım yerlerine göre kolaylıklar sağlamaktadır.
- C programlama dilinde üç tip döngü deyimi vardır:
 - for döngü yapısı
 - while döngü yapısı
 - do...while döngü yapısı
- Döngünün ne kadar çalışacağı belli ise genellikle for döngüsü kullanılır.
- Döngünün ne kadar çalışacağı belli değilse while ya da do-while döngüleri kullanılır.

for Döngü Yapısı

- **Belirli sayıda tekrarı** çalıştırmak için kullanılır. Sayaçlı döngü yapısı şeklinde de ifade edilir.
- Döngünün istenilen sayıda çalışması için **bir başlangıç** ve **bir bitiş** değeri belirtilmelidir.
- Aksi belirtilmedikçe sayaç birer birer artar.
- Kullanım Şekli:

```
for (başlangıç; koşul; artım)
{
    ...
    döngüde yapılacak işlemler;
    ...
}
```



```
int i;  
for ( $i = \text{başlangıç değeri}$ ; koşul;  $i = i + \text{artış miktarı}$ )  
{  
    ...  
    Yapılacak işlemler  
    ...  
}
```

Örnek 1: 1 ile 20 arasındaki sayıların yazdıran ve bu sayıların toplamını bulan bir C programı yazınız.

```
#include<stdio.h>
#include<conio.h>
int main()
{
int i,top=0;
for (i=1;i<=20;i++)
{
printf("i= %d\n",i);
top=top+i;
}
printf("Sayilarin toplami= %d",top);
getch();
return 0;
}
```

```
i= 1
i= 2
i= 3
i= 4
i= 5
i= 6
i= 7
i= 8
i= 9
i= 10
i= 11
i= 12
i= 13
i= 14
i= 15
i= 16
i= 17
i= 18
i= 19
i= 20
Sayilarin toplami= 210
-----
Process exited after 13.76 seconds with return value 0
Press any key to continue . . .
```

while Döngü Yapısı

- Bir koşulun sağlanması sonucunda belirli komutların tekrarlanması durumlarında while döngü yapısı kullanılır. Koşullu döngü yapısı da denir.
- Döngünün başlaması için ilk koşul mutlaka sağlanmalıdır. (doğru olmalıdır.)
- Koşul doğru olduğu sürece döngü devam eder.
- Kullanım Şekli:

while (koşul)

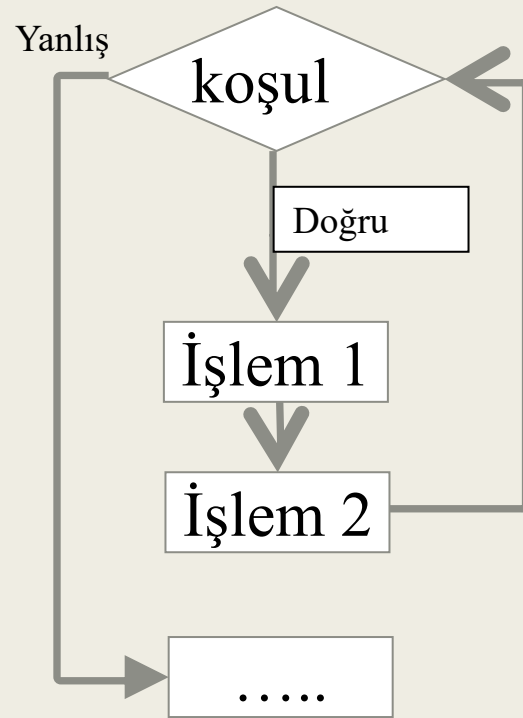
{

...

döngüde yapılacak işlemler;

...

}



```
while (koşul)
{
    ...
    Yapılacak işlemler
    ...
}
```

Örnek 1: Negatif bir tamsayı girilene kadar girilen tüm sayıların toplamını bulan bir C programı yazınız.

```
#include<stdio.h>
#include<conio.h>
int main()
{ int x,top=0,i=1;
printf("%d.ci tamsayiyi giriniz: ",i);
scanf("%d",&x);
while (x>=0)
{
top=top+x;
i++;
printf("%d.ci tamsayiyi giriniz: ",i);
scanf("%d",&x);
}
printf("Negatif sayi girilinceye kadar tum sayilarin toplami= %d ",top);
getch();
return 0;
}
```

```
1.ci tamsayiyi giriniz: 35
2.ci tamsayiyi giriniz: 20
3.ci tamsayiyi giriniz: 10
4.ci tamsayiyi giriniz: -5
Negatif sayi girilinceye kadar tum sayilarin toplami= 65
-----
Process exited after 13.12 seconds with return value 0
Press any key to continue . . .
```

Örnek 2: Belirsiz sayıda girilen sayıların toplamını bulan bir C programı yazınız.

```
#include<stdio.h>
#include<conio.h>
int main()
{ char cev='E'; int i=1,top=0,x;
while ((cev=='E') || (cev=='e') )
{
printf("%d.ci sayiyi giriniz: ",i);
scanf("%d",&x);
top=top+x;
i++;
printf("Baska sayi varsa \"E\" veya \"e\" giriniz, aksi halde farkli bir harf giriniz \n");
cev=getche();
printf("\n");
}
printf("\nSayilarin toplami= %d",top);
getch();
return 0;
}
```

getche() fonksiyonu klavyeden tek bir karakter okur.

--- «conio.h» ile kullanılır.

```
1.ci sayiyi giriniz: 45
Baska sayi varsa "E" veya "e" giriniz, aksi halde farkli bir harf giriniz
e
2.ci sayiyi giriniz:
35
Baska sayi varsa "E" veya "e" giriniz, aksi halde farkli bir harf giriniz
E
3.ci sayiyi giriniz: 20
Baska sayi varsa "E" veya "e" giriniz, aksi halde farkli bir harf giriniz
Y

Sayilarin toplami= 100
-----
Process exited after 21.56 seconds with return value 0
Press any key to continue . . .
```

Döngü sayısı belli olan durumlarda da kullanılabilir...

Örnek 3: 1'den 20'ye kadar olan sayıların toplamını bulan bir C programı yazınız.

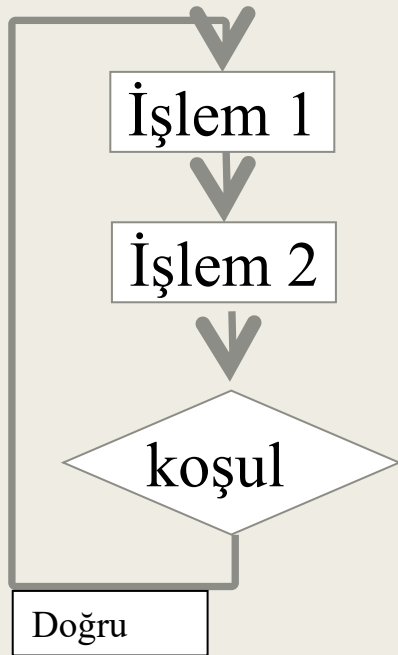
```
#include<stdio.h>
#include<conio.h>
int main()
{
    int i=1,top=0;
    while(i<=20)
    {
        printf("i= %d\n",i);
        top=top+i;
        i++;
    }
    printf("Sayilarin toplami= %d",top);
    getch();
    return 0;
}
```

```
i= 1
i= 2
i= 3
i= 4
i= 5
i= 6
i= 7
i= 8
i= 9
i= 10
i= 11
i= 12
i= 13
i= 14
i= 15
i= 16
i= 17
i= 18
i= 19
i= 20
Sayilarin toplami= 210
-----
Process exited after 2.11 seconds with return value 0
Press any key to continue . . .
```

do while Döngü Yapısı

- While döngü yapısı ile aynı mantıkta çalışır.
- While döngü yapısından farkı koşulun döngünün sonunda kontrol edilmesidir.
- Koşul kontrol edilmeden döngüye girilir ve en az bir kere çalışır.
- Koşul yanlış olunca döngüden çıkılır.
- Kullanım Şekli:

```
do
{
    ...
    döngüde yapılacak işlemler;
    ...
} while (koşul);
```



```
do
{
    ...
    Yapılacak işlemler
    ...
}while (koşul);
```

Örnek 1: Negatif bir tamsayı girilene kadar girilen tüm sayıların toplamını bulan bir C programı yazınız.

```
#include<stdio.h>
#include<conio.h>
int main()
{ int x,top=0,i=1;
printf("%d.ci tamsayiyi giriniz: ",i);
scanf("%d",&x);
if (x<0) {printf ("Negatif sayi girildiginden sonlandirilmistir...");
        goto son;}
do
{
top=top+x;
i++;
printf("%d.ci tamsayiyi giriniz: ",i);
scanf("%d",&x);
} while (x>=0);
printf("Negatif sayi girilinceye kadar tum sayilarin toplami= %d ",top);
son:
getch();
return 0;
}
```

```
1.ci tamsayiyi giriniz: 35
2.ci tamsayiyi giriniz: 20
3.ci tamsayiyi giriniz: -5
Negatif sayi girilinceye kadar tum sayilarin toplami= 55
-----
Process exited after 7.565 seconds with return value 0
Press any key to continue . . . _
```

```
1.ci tamsayiyi giriniz: -5
Negatif sayi girildiginden sonlandirilmistir...
-----
Process exited after 4.455 seconds with return value 0
Press any key to continue . . . _
```


Döngü sayısı belli olan durumlarda da kullanılabilir...

Örnek 2: 1'den 20'ye kadar olan sayıların toplamını bulan bir C programı yazınız.

```
#include<stdio.h>
#include<conio.h>
int main()
{
    int i=1,top=0;
    do
    {
        printf("i= %d\n",i);
        top=top+i;
        i++;
    } while(i<=20);
    printf("Sayilarin toplami= %d",top);
    getch();
    return 0;
}
```

```
i= 1
i= 2
i= 3
i= 4
i= 5
i= 6
i= 7
i= 8
i= 9
i= 10
i= 11
i= 12
i= 13
i= 14
i= 15
i= 16
i= 17
i= 18
i= 19
i= 20
Sayilarin toplami= 210
-----
Process exited after 1.427 seconds with return value 0
Press any key to continue . . .
```

break Deyimi

- Döngü işlemi devam ederken döngünün koşuluna bağlı olmaksızın döngüden çıkılmasını sağlayan deyimdir.
- Döngü içinde bu deyime sıra geldiğinde, break ardından döngü sonuna kadar olan tüm deyimler atlanır.
- Tüm döngü tiplerinde kullanılabilir.
- Kullanım biçimi:

break;

En genel kullanımı bir döngüden istenen anda çıkmaktır.

Örnek: break kullanımına bir örnek...

```
#include <stdio.h>
#include <conio.h>
int main()
{
    int x;
    for ( x = 1; x <= 10; x++ ) {
        if ( x == 5 )
            break;
        printf( "%d ", x );
    }
    printf( "\nDonguden x= %d oldugunda cikildi\n", x );
    getch();
    return 0;
}
```

```
1 2 3 4
Donguden x= 5 oldugunda cikildi

-----
Process exited after 12.6 seconds with return value 0
Press any key to continue . . .
```

continue Deyimi

- Bir döngüyü terk etmeden bir adımın atlanması söz konusu olduğunda kullanılan deyimdir.
- Kullanım biçimi aşağıdaki gibidir:

continue;

Örnek: continue kullanımına bir örnek...

```
#include <stdio.h>
#include <conio.h>
int main()
{
    int x;
    for ( x = 1; x <= 10; x++ ) {
        if ( x == 5 )
            continue;
        printf( "%d ", x );
    }
    printf( "\ncontinue, 5 degerinin atlanmasi icin kullanildi\n" );
    getch();
    return 0;
}
```

```
1 2 3 4 6 7 8 9 10
continue, 5 degerinin atlanmasi icin kullanildi

-----
Process exited after 9.256 seconds with return value 0
Press any key to continue . . .
```

İç-içe döngüler

Bir programda iç içe bir çok döngü kullanılabilir.

Program önce en içteki döngüden başlar ve dıştaki döngüye doğru gider.

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

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

```
int main()
```

```
{int i,j;
```

```
  for ( i = 1; i <= 3; i++ )
```

```
  {
```

```
    for ( j = 1; j <= 5; j++ )
```

```
    {
```

```
      printf( "%d --- %d\n", i,j);
```

```
    }
```

```
  }
```

```
  getch();
```

```
  return 0;
```

```
}
```



Dıştaki döngü



İçteki döngü

```
1 --- 1
1 --- 2
1 --- 3
1 --- 4
1 --- 5
2 --- 1
2 --- 2
2 --- 3
2 --- 4
2 --- 5
3 --- 1
3 --- 2
3 --- 3
3 --- 4
3 --- 5
```

```
-----
```

```
Process exited after 15.34 seconds with return value 0
Press any key to continue . . .
```

Örnek 1: İki basamaklı tüm asal sayıları listeleyen bir C programı yazınız.

```
#include<stdio.h>
#include<conio.h>
int main()
{int i, j, bs;
for (i=10;i<=99;i++)
{ bs=0;
for (j=1;j<=i;j++)
{
if(i%j==0) bs++;
}
if(bs==2) printf("%d bir asal sayidir.\n", i);
}
getch();
return 0;
}
```

```
11 bir asal sayidir.
13 bir asal sayidir.
17 bir asal sayidir.
19 bir asal sayidir.
23 bir asal sayidir.
29 bir asal sayidir.
31 bir asal sayidir.
37 bir asal sayidir.
41 bir asal sayidir.
43 bir asal sayidir.
47 bir asal sayidir.
53 bir asal sayidir.
59 bir asal sayidir.
61 bir asal sayidir.
67 bir asal sayidir.
71 bir asal sayidir.
73 bir asal sayidir.
79 bir asal sayidir.
83 bir asal sayidir.
89 bir asal sayidir.
97 bir asal sayidir.
```

```
-----
Process exited after 5.349 seconds with return value 0
Press any key to continue . . .
```


Örnek 2: 3 basamaklı sayılar içerisinde basamaklarının küplerinin toplamı kendisine eşit olan sayıları bulan bir C programı yazınız.

```
#include <stdio.h>
#include <conio.h>
int main()
{ int i, a, b, c;
  for (i=100; i<=999; i++)
  { a = i/100 ; /* sayının yüzler basamağı*/
    b = (i-a*100)/ 10 ; /* sayının onlar basamağı*/
    c = i-(a * 100 + b * 10) ; /* sayının birler basamağı*/
    if ( i == (a*a*a + b*b*b + c*c*c) ) printf ("%d \n" , i) ;
  }
  getch() ;
  return 0;
}
```

```
153
370
371
407

-----
Process exited after 2.022 seconds with return value 0
Press any key to continue . . .
```

Örnek 3: Klavyeden girilecek bir n tamsayısı için Aşağıdaki görüntüyü elde eden programı while ve do-while döngülerini birer kere kullanarak yazdıran bir program yazınız.

```
bir n sayisi giriniz=10
1
12
123
1234
12345
123456
1234567
12345678
123456789
12345678910

-----
Process exited with return value 0
Press any key to continue . . . _
```

```
#include<stdio.h>
#include<conio.h>
int main()
{
int i=1,j,n;
printf ("bir n sayisi giriniz: ");
scanf("%d= ",&n);
printf ("\n");
while(i<=n)
{
j=1;
do
{
printf ("%d",j);
j++;
}while (j<=i);
printf ("\n");
i++;
}
getch();
return 0;
}
```

```
bir n sayisi giriniz: 5

1
12
123
1234
12345

-----
Process exited after 2.821 seconds with return value 0
Press any key to continue . . .
```

Aynı soruyu iç-içe for döngüleriyle yapalım...

```
#include<stdio.h>
#include<conio.h>
int main()
{
    int i,j,n;
    printf ("bir n sayisi giriniz: ");
    scanf("%d= ",&n);
    printf ("\n");
    for (i=1;i<=n;i++)
    {
        for (j=1;j<=i;j++)
        {
            printf ("%d",j);
        }
        printf ("\n");
    }
    getch();
    return 0;
}
```

```
bir n sayisi giriniz: 5

1
12
123
1234
12345

-----
Process exited after 2.821 seconds with return value 0
Press any key to continue . . .
```

Sonsuz Döngüler

Bir döngü işlemini sonsuz kere tekrarlırsa bu döngü sonsuz döngü olarak adlandırılır.

Örnekler:

```
#include<stdio.h>
#include<conio.h>
int main()
{
    for(;;)
        printf("Sonsuz dongu...\n");
    getch();
    return 0;
}
```

```
#include<stdio.h>
#include<conio.h>
int main()
{
    while(10>5)
    {
        printf("Sonsuz dongu...\n");
    }
    getch();
    return 0;
}
```

```
#include<stdio.h>
#include<conio.h>
int main()
{
    while(1) {
        printf("Sonsuz dongu...\n");
    }
    getch();
    return 0;
}
```

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.