



DÖNGÜLER (LOOPS)

- for Döngüsü
- foreach Döngüsü
- while Döngüsü
- do while Döngüsü

Adem AKKUŞ

Bilgisayar Mühendisi
Bilişim Teknolojileri Öğr.
Eğitmen

DÖNGÜLER

Programlama dilleri açısından bazı işlemlerin tekrar edilmesi gerekir. Bu işlemler döngü yapıları ile gerçekleştirilir.

Örnek bir mesajın 10 defa yazdırılması, 1'den n'e kadar olan sayıların toplamı, dizi işlemlerinde vb.

Aşağıdaki çıktıyı normal kod ve döngü ile oluşturalım.

[illegible]

DÖNGÜLER

Programlama dilleri açısından bazı işlemlerin tekrar edilmesi gerekir. Bu işlemler döngü yapıları ile gerçekleştirilir.

Örnek bir mesajın 10 defa yazdırılması, 1'den n'e kadar olan sayıların toplamı, dizi işlemlerinde vb.

Aşağıdaki çıktıyı normal kod ve döngü ile oluşturalım.

[illegible][illegible]

```
for (int i = 0; i < 10; i++)
{
    Console.WriteLine("C# Programlamaya Hoş geldiniz.");
}
```

[illegible]

DÖNGÜLER

Döngüler, bu tekrar işlemlerini hem daha hızlı hem de daha az kod satırıyla gerçekleştirir.

Dört döngü yapısı vardır:

1. for
2. foreach
3. while
4. do while

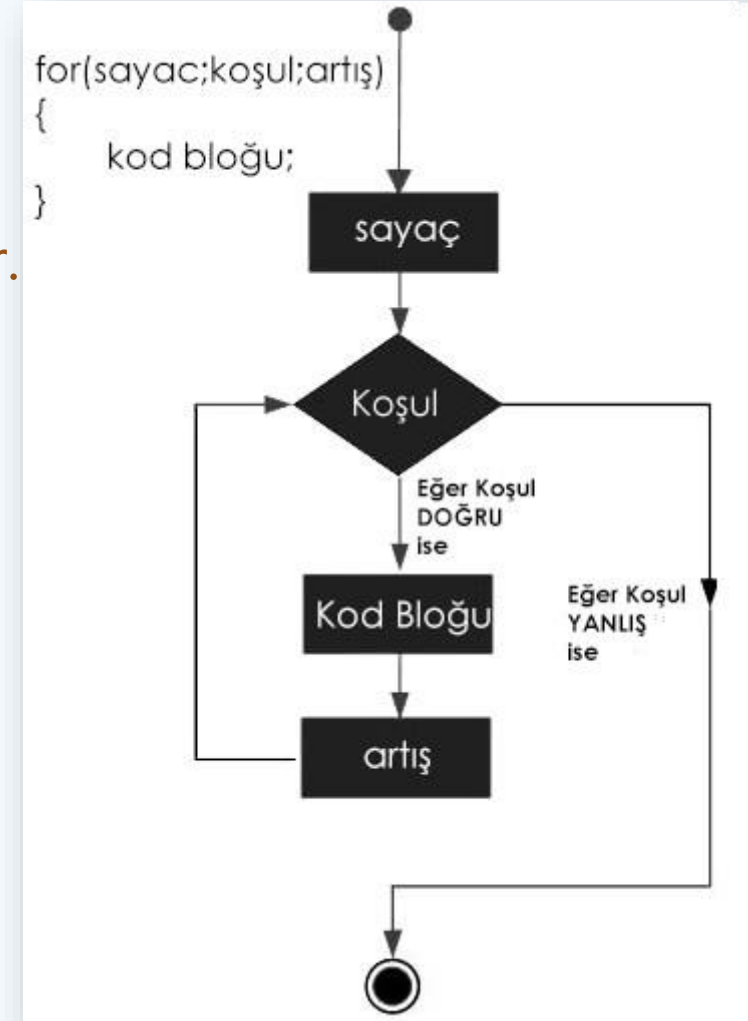
A stylized landscape illustration featuring rolling green hills in the foreground, a small tree with a brown trunk and purple and pink foliage on the left, and blue and white wavy hills in the background. The text 'for Döngüsü' is written in a brown, cursive font in the center-right area.

for Döngüsü

Adem AKKUŞ

for Döngüsü

- ❑ for yapısında önce döngü değişkenine ilk değer ataması yapılır.
- ❑ Daha sonra koşul sınanır.
- ❑ Eğer koşul sağlanıyorsa döngüye devam edilir.
- ❑ Koşul sağlanmıyorsa for yapısından sonraki komuttan işletilmeye devam edilir.



for Döngüsü

Bir ya da birden fazla satırdan oluşan kodun belirtilen şarta göre, belirtilen artış ya da azalış miktarına bağlı olarak tekrarlanmasını sağlayan döngüdür. Artış miktarı, bazı kaynaklarda , adım değeri olarak da ifade edilir.

for(başlangıç değeri; bitiş şartı; artış miktarı)

{

}

//tekar tekrar çalışacak kod

```
for (int i = 0 ; i < 10 ; i++)  
{  
    Console.WriteLine("C# Programlamaya Hoş geldiniz.");  
}
```

for Döngüsü

Karar Tablosu (decision table): Kod çıktılarını incelemek için kullanılabilir.

Aşağıdaki kod için karar tablosu uygulamaya çalışalım. iterasyon: döngünün her adımı.

```
int karesi;  
for (int i = 0; i < 5; i++)  
{  
    karesi = i * i;  
    Console.WriteLine(karesi);  
}
```

Kod çıktısı

```
C:\Users\PC\source\repos\Hafta6\  
0  
1  
4  
9  
16
```

iterasyon	i	bitiş şartı	karesi (i*i)	çıktı
1	0	True	0*0=0	0
2	1	True	1*1=1	1
3	2	True	2*2=4	4
4	3	True	3*3=9	9
5	4	True	4*4=16	16
	5	False		

for Döngüsü

Bir ya da birden fazla satırdan oluşan kodun belirtilen şarta göre, belirtilen artış ya da azalış miktarına bağlı olarak tekrarlanmasını sağlayan döngüdür.

Pozitif artış örnekler;

```
for (int i = 0 ; i < 10 ; i++)  
{  
    Console.WriteLine("C# Programlamaya Hoş geldiniz.");  
}
```

```
for (int i = 0 ; i < 10 ; i*=2)  
{  
    Console.WriteLine("C# Programlama.");  
}
```

```
int i;  
for (i = 0 ; i < 10 ; i++)  
{  
    Console.WriteLine("C# Programlama.");  
}
```

```
for (int i = 0 ; i < 10 ; i+=2)  
{  
    Console.WriteLine("C# Programlama.");  
}
```

```
int i = 0;  
for (; i < 10;) {  
    Console.WriteLine("C# Programlamaya Programlama.");  
    i++;  
}
```

for Döngüsü

Bir ya da birden fazla satırdan oluşan kodun belirtilen şarta göre, belirtilen artış ya da azalış miktarına bağlı olarak tekrarlanmasını sağlayan döngüdür.

Negatif artış örnekler;

```
for (int i = 10 ; i > 10 ; i--)  
{  
    Console.WriteLine("C# Programlama.");  
}
```

```
for (int i = 10 ; i > 10 ; i-=2)  
{  
    Console.WriteLine("C# Programlama.");  
}
```

```
for (int i = 10 ; i > 0 ; i/=2)  
{  
    Console.WriteLine("C# Programlama.");  
}
```

for Döngüsü - Örnek

0 – 20 arasında ardışık, tek, çift, üçün ve beşin katı olan sayıları ekrana yazdıralım.

```
Console.WriteLine("0-20 arası ardışık sayılar");
for (int i = 0; i <= 20; i++)
{
    Console.Write(i + ", ");
}

Console.WriteLine("\n 0-20 arası  ardışık çift sayılar");
for (int i = 0; i <= 20; i += 2)
{
    Console.Write(i + ", ");
}

Console.WriteLine("\n 0-20 arası ardışık tek sayılar");
for (int i = 1; i <= 20; i += 2)
{
    Console.Write(i + ", ");
}

Console.WriteLine("\n 0-20 arası 3 katı ardışık sayılar");
for (int i = 0; i <= 20; i += 3)
{
    Console.Write(i + ", ");
}

Console.WriteLine("\n 0-20 arası 5 katı ardışık sayılar");
for (int i = 0; i <= 20; i += 5)
{
    Console.Write(i + ", ");
}
```

```
C:\Users\PC\source\repos\Hafta6\ForDongusu_UcKatiSayilar\bin\Debug\net6.0\ForDongusu_UcKatiSayilar.exe
0-20 arası ardışık sayılar
0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20,

 0-20 arası  ardışık çift sayılar
0, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20,

 0-20 arası ardışık tek sayılar
1, 3, 5, 7, 9, 11, 13, 15, 17, 19,

 0-20 arası 3 katı ardışık sayılar
0, 3, 6, 9, 12, 15, 18,

 0-20 arası 5 katı ardışık sayılar
0, 5, 10, 15, 20,
```

for Döngüsü - Örnek

Kullanıcı tarafından girilen sayının faktöriyelini hesaplayıp ekrana yazdıran program.

Faktöriyel matematiksel temelleri;

- Negatif sayıların faktöriyeli yok.
- $0! = 1$,
- $1! = 1$
- $2! = 2.1$
- $n! = n.(n-1)!$
- $5! = 5.4! = 5.4.3.2.1 = 120$

for Döngüsü - Örnek

Kullanıcı tarafından girilen sayının faktöriyelini hesaplayıp ekrana yazdıran program.

```
int faktoriyel = 1;
int sayi;
Console.Write("Faktoriyeli alınacak sayı:");
sayi=Convert.ToInt32(Console.ReadLine());
for (int i = 1; i <=sayi; i++)
{
    faktoriyel *= i; //faktoriyel =faktoriyel* i;
}
Console.WriteLine($"{sayi} != {faktoriyel}");
```

Microsoft Visual Studio Debug Console

```
Faktoriyeli alınacak sayı:5
5 != 120
```

Microsoft Visual Studio Debug Console

```
Faktoriyeli alınacak sayı:8
8 != 40320
```

for Döngüsü - Örnek

Harfleri a-z ve A-Z ekrana yazdıran program.

Harf char tipindedir. char tipi üzerinden artırma azaltma işlemleri yapılabilir.

```
for(char harf='a';harf<='z';harf++)  
{  
    Console.Write(harf+"-");  
}
```

```
C:\Users\PC\source\repos\Hafta6\ForDongusu_CharOrnek\bin\Debug\net6.0\ForDongusu_CharOrnek.exe  
a-b-c-d-e-f-g-h-i-j-k-l-m-n-o-p-q-r-s-t-u-v-w-x-y-z-
```

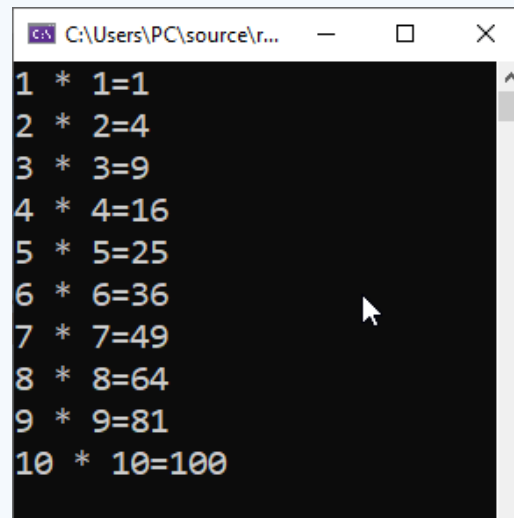
```
for(char harf='A';harf<='Z';harf++)  
{  
    Console.Write(harf+"-");  
}
```

```
C:\Users\PC\source\repos\Hafta6\ForDongusu_CharOrnek\bin\Debug\net6.0\ForDongusu_CharOrnek.exe  
A-B-C-D-E-F-G-H-I-J-K-L-M-N-O-P-Q-R-S-T-U-V-W-X-Y-Z-
```

for Döngüsü – İki Değişkenli Örnek

Bir for döngüsü içerisinde birden fazla değişken ya da şart tanımlaması yapılabilir.

```
for (int i=1, j =1 ; i<=10 & j<= 10; j++,i++)  
{  
    Console.WriteLine($"{i} * {j}={i*j}");  
}
```

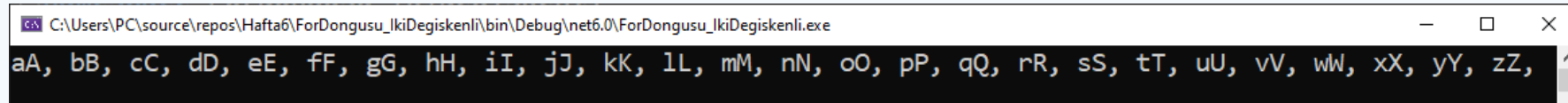


```
C:\Users\PC\source\r...  
1 * 1=1  
2 * 2=4  
3 * 3=9  
4 * 4=16  
5 * 5=25  
6 * 6=36  
7 * 7=49  
8 * 8=64  
9 * 9=81  
10 * 10=100
```

for Döngüsü – İki Değişkenli Örnek

Bir for döngüsü içerisinde birden fazla değişken ya da şart tanımlaması yapılabilir.

```
for (char i='a', j = 'A' ; i<='z' & j<= 'Z'; j++,i++)  
{  
    Console.Write($"{i}{j}, ");  
}
```



C:\Users\PC\source\repos\Hafta6\ForDongusu_IkiDegiskenli\bin\Debug\net6.0\ForDongusu_IkiDegiskenli.exe

aA, bB, cC, dD, eE, fF, gG, hH, iI, jJ, kK, lL, mM, nN, oO, pP, qQ, rR, sS, tT, uU, vV, wW, xX, yY, zZ,

for Döngüsü – Sonsuz Döngü

Sonsuz for döngüsü çeşitleri.

```
for (; ; )  
{  
    Console.WriteLine("Sonsuz döngü.");  
}
```

```
for (int i = 0; i < 10; )  
{  
    Console.WriteLine("Sonsuz döngü.");  
}
```

```
for (; true; )  
{  
    Console.WriteLine("Sonsuz döngü.");  
}
```

```
for (int i = 0; i < 10; i--)  
{  
    Console.WriteLine("Sonsuz döngü.");  
}
```

```
for (int i = 0; i > 0; i++)  
{  
    Console.WriteLine("Sonsuz döngü.");  
}
```

for Döngüsü – break Kullanımı

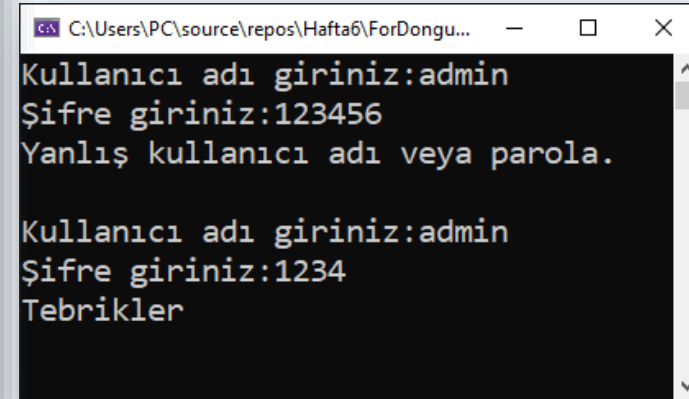
break keyword'ü; switch case ve döngüleri sonlandırmak için kullanılır. Ayrıca döngü içerisindeki if - else yapılarını sonlandırmak için de kullanılabilir.

- Normalde for döngüsü belirtilen şart sağlandığı müddetçe döngü çalışır.
- Bazen herhangi bir durumda döngüden çıkılması istenebilir.
- Örnek, kullanıcı adı ve şifresini girdiği bir programda sonsuz döngü açılır.
- Kullanıcının girdiği şifre ve kullanıcı doğru olduğunda döngünün bitirilir.

for Döngüsü – break Kullanımı - Örnek

Örnek, kullanıcı adı ve şifresini girdiği bir programda sonsuz döngü açılır.

```
string? user, pass;  
string kullanıcıAdi = "admin";  
string sifre = "1234";  
for (; ; )  
{  
    Console.Write("Kullanıcı adı giriniz:");  
    user = Console.ReadLine();  
    Console.Write("Şifre giriniz:");  
    pass = Console.ReadLine();  
    if (user == kullanıcıAdi && pass == sifre)  
    {  
        Console.WriteLine("Tebrikler");  
        break; //döngü sonlanır.  
    }  
    else  
    {  
        Console.WriteLine("Yanlış kullanıcı adı veya parola.\n");  
    }  
}
```



```
C:\Users\PC\source\repos\Hafta6\ForDongu...  
Kullanıcı adı giriniz:admin  
Şifre giriniz:123456  
Yanlış kullanıcı adı veya parola.  
  
Kullanıcı adı giriniz:admin  
Şifre giriniz:1234  
Tebrikler
```

for Döngüsü – break Kullanımı - Ödev

Ödev : Kullanıcı tarafından sıfır girilene kadar, girilen sayıların toplamı ve ortalamasını ekrana yazdıran programı yazınız?

for Döngüsü – continue Kullanımı

continue keyword'ü,

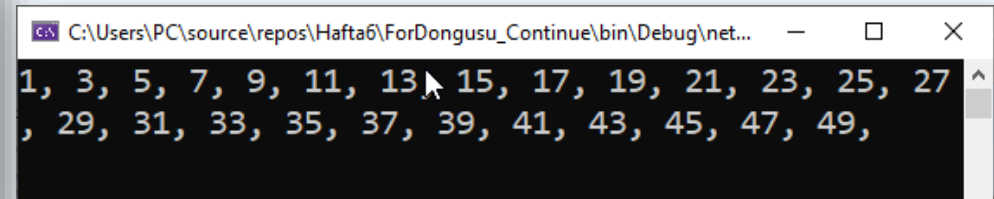
döngünün bu iterasyonunda hiçbir işlem yapmadan diğer iterasyona geçer. Yani döngü kaldığı yerden devam eder.

- ❑ Normalde for döngüsü belirtilen şart sağlandığı müddetçe döner.
- ❑ Bazen döngünün belirli bir iterasyonlarda boş dönmesi istenebilir
- ❑ continue deyimi kullanılırsa döngü iterasyonda içerisindeki deyimleri işletmeyecektir.

for Döngüsü – continue Kullanımı - Örnek

1 - 50 arasındaki sayılardan ardışık tek sayıları ekrana yazdıralım.

```
//1-50 arasındaki sayılardan, ardışık tek sayıları  
for (int i = 1; i <=50; i++)  
{  
    if (i % 2 == 0)  
        continue;  
    else  
        Console.Write(i.ToString()+" ");  
}
```



```
C:\Users\PC\source\repos\Hafta6\ForDongusu_Continue\bin\Debug\net...  
1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27  
, 29, 31, 33, 35, 37, 39, 41, 43, 45, 47, 49,
```

İç İçe for Döngüsü

Matris yapısındaki veriler üzerinde işlem yapabilmek için kullanılan yapıdır. Matris yapıları satır ve sütunlardan oluşan yapılardır. Örneğin tablolar gibi.

sütunlar

AD	SOYAD	SINIF
Osman	KIVRAK	6-A
Hasan	EKİNCİ	8-C
İsmail	ARIKAN	7-A

satırlar

İç İçe for Döngüsü

Matris yapısındaki veriler üzerinde işlem yapabilmek için kullanılan yapıdır. Matris yapıları satır ve sütunlardan oluşan yapılardır. Örneğin tablolar gibi.

satırlar

----->	AD	SOYAD	SINIF
----->	Osman	KIVRAK	6-A
----->	Hasan	EKİNCİ	8-C
----->	İsmail	ARIKAN	7-A

İç İçe for Döngüsü

Matris yapısındaki veriler üzerinde işlem yapabilmek için kullanılan yapıdır. Matris yapıları satır ve sütunlardan oluşan yapılardır. Örneğin tablolar gibi.

sütunlar



AD	SOYAD	SINIF
Osman	KIVRAK	6-A
Hasan	EKİNCİ	8-C
İsmail	ARIKAN	7-A

İç İçe for Döngüsü

Matris yapısındaki veriler üzerinde işlem yapabilmek için kullanılan yapıdır. Matris yapıları satır ve sütunlardan oluşan yapılardır. Örneğin tablolar gibi.

sütunlar



AD	SOYAD	SINIF
Osman	KIVRAK	6-A
Hasan	EKİNCİ	8-C
İsmail	ARIKAN	7-A

İç İçe for Döngüsü

- ❑ Matris yapısında herhangi bir elemanın konumu satır ve sütun numarası ile belirlenir.
- ❑ İlk numara **satır** ikinci numara **sütunu** ifade eder.

1, 1	1, 2	1, 3
2, 1	2, 2	2, 3
3, 1	3, 2	3, 3
4, 1	4, 2	4, 3

İç İçe for Döngüsü

- ❑ Matris tipi yapılarda gezinirken iç içe 2 adet for döngüsü kullanılır.
- ❑ Bu döngülerden dıştaki döngü yavaş içteki ise hızlı döner.
- ❑ Yavaş dönen döngü **satır numaralarını**, hızlı olanı ise **sutun numaralarını** belirler.

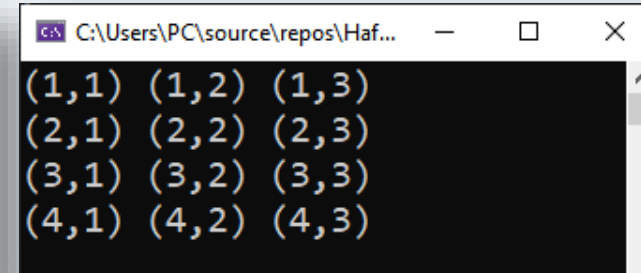
1, 1	1, 2	1, 3
2, 1	2, 2	2, 3
3, 1	3, 2	3, 3
4, 1	4, 2	4, 3

İç İçe for Döngüsü - Örnek

4 satır ve 3 sütun bir matris oluşturunuz?

(1, 1)	(1, 2)	(1, 3)
(2, 1)	(2, 2)	(2, 3)
(3, 1)	(3, 2)	(3, 3)
(4, 1)	(4, 2)	(4, 3)

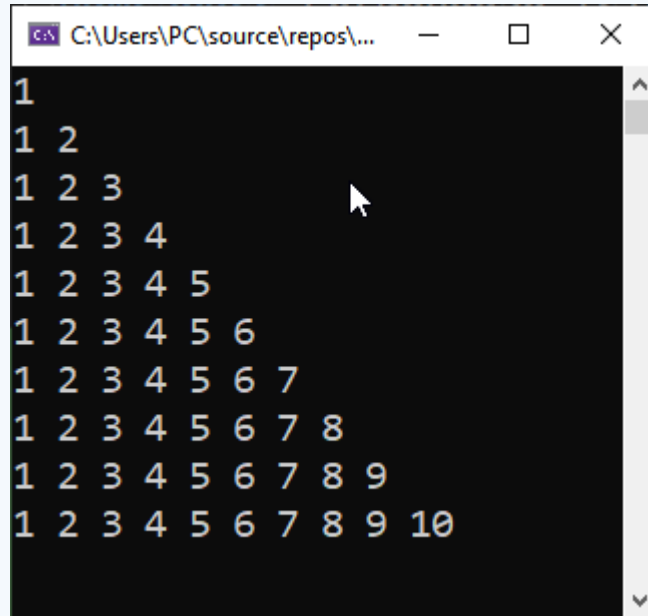
```
//iç içe for döngüsü  
//en dıştaki satır ve içeri doğru sütun vb şeklindedir  
for (int satir = 1; satir < 5; satir++)  
{  
    for (int sutun = 1; sutun < 4; sutun++)  
    {  
        Console.Write($"({satir},{sutun})");  
    }  
    Console.WriteLine();  
}
```



```
(1,1) (1,2) (1,3)  
(2,1) (2,2) (2,3)  
(3,1) (3,2) (3,3)  
(4,1) (4,2) (4,3)
```


İç İçe for Döngüsü - Örnek

Aşağıdaki gibi çıktı veren kodu yazınız?



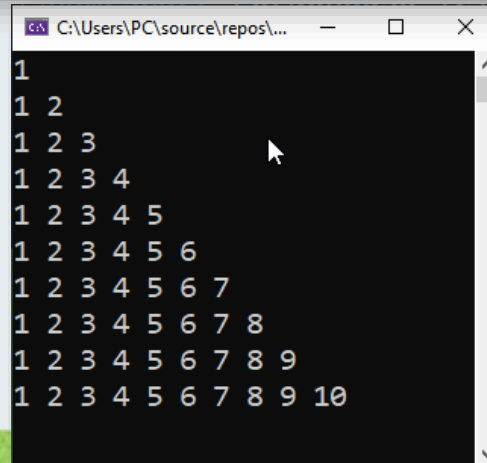
```
C:\Users\PC\source\repos\...  
1  
1 2  
1 2 3  
1 2 3 4  
1 2 3 4 5  
1 2 3 4 5 6  
1 2 3 4 5 6 7  
1 2 3 4 5 6 7 8  
1 2 3 4 5 6 7 8 9  
1 2 3 4 5 6 7 8 9 10
```

The image shows a screenshot of a Windows command prompt window. The title bar indicates the path is C:\Users\PC\source\repos\... The window contains a series of numbers printed in a nested loop pattern. The first line is '1'. The second line is '1 2'. The third line is '1 2 3'. The fourth line is '1 2 3 4'. The fifth line is '1 2 3 4 5'. The sixth line is '1 2 3 4 5 6'. The seventh line is '1 2 3 4 5 6 7'. The eighth line is '1 2 3 4 5 6 7 8'. The ninth line is '1 2 3 4 5 6 7 8 9'. The tenth line is '1 2 3 4 5 6 7 8 9 10'. A mouse cursor is visible over the third line.

İç İçe for Döngüsü - Örnek

Aşağıdaki gibi çıktı veren kodu yazınız?

```
for (int i = 1; i <=10; i++)  
{  
    for (int j = 1; j <=i; j++)  
    {  
        Console.Write($"{j} ");  
    }  
    Console.WriteLine();  
}
```

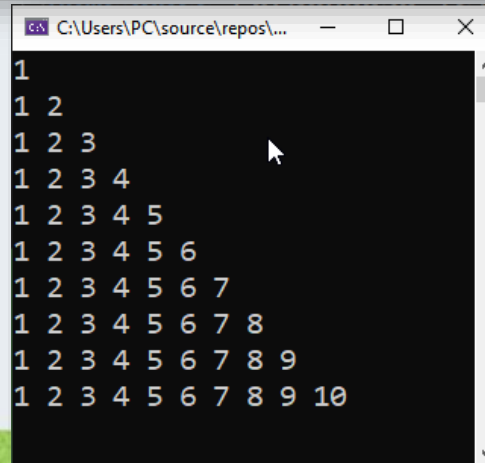


```
1  
1 2  
1 2 3  
1 2 3 4  
1 2 3 4 5  
1 2 3 4 5 6  
1 2 3 4 5 6 7  
1 2 3 4 5 6 7 8  
1 2 3 4 5 6 7 8 9  
1 2 3 4 5 6 7 8 9 10
```

İç İçe for Döngüsü - Örnek

Aşağıdaki gibi çıktı veren kodu yazınız?

```
for (int i = 1; i <=10; i++)  
{  
    for (int j = 1; j <=i; j++)  
    {  
        Console.Write($"{j} ");  
    }  
    Console.WriteLine();  
}
```



```
1  
1 2  
1 2 3  
1 2 3 4  
1 2 3 4 5  
1 2 3 4 5 6  
1 2 3 4 5 6 7  
1 2 3 4 5 6 7 8  
1 2 3 4 5 6 7 8 9  
1 2 3 4 5 6 7 8 9 10
```

İç İçe for Döngüsü – Çarpım Tablosu

Çarpım tablosunu iç içe for döngüsü ile yapmaya çalışalım.

```
for (int i = 1; i <=10; i++)  
{  
    for (int j = 1; j <=10; j++)  
    {  
        Console.WriteLine($"{i}*{j}={i*j}");  
    }  
    Console.WriteLine();  
}
```

1*1=1
1*2=2
1*3=3
1*4=4
1*5=5
1*6=6
1*7=7
1*8=8
1*9=9
1*10=10

2*1=2
2*2=4
2*3=6
2*4=8
2*5=10
2*6=12
2*7=14
2*8=16
2*9=18
2*10=20

3*1=3
3*2=6
3*3=9
3*4=12
3*5=15
3*6=18
3*7=21
3*8=24
3*9=27
3*10=30

4*1=4
4*2=8
4*3=12
4*4=16
4*5=20
4*6=24
4*7=28
4*8=32
4*9=36
4*10=40

5*1=5
5*2=10
5*3=15
5*4=20
5*5=25
5*6=30
5*7=35
5*8=40
5*9=45
5*10=50

6*1=6
6*2=12
6*3=18
6*4=24
6*5=30
6*6=36
6*7=42
6*8=48
6*9=54
6*10=60

7*1=7
7*2=14
7*3=21
7*4=28
7*5=35
7*6=42
7*7=49
7*8=56
7*9=63
7*10=70

8*1=8
8*2=16
8*3=24
8*4=32
8*5=40
8*6=48
8*7=56
8*8=64
8*9=72
8*10=80

9*1=9
9*2=18
9*3=27
9*4=36
9*5=45
9*6=54
9*7=63
9*8=72
9*9=81
9*10=90

10*1=10
10*2=20
10*3=30
10*4=40
10*5=50
10*6=60
10*7=70
10*8=80
10*9=90
10*10=100

for Döngüsü - Ödev

Aşağıdaki görüntü verecek çarpım tablosu kodlarını yazınız.

```
Microsoft Visual Studio Debug Console
```

1*1=1	2*1=2	3*1=3	4*1=4	5*1=5
1*2=2	2*2=4	3*2=6	4*2=8	5*2=10
1*3=3	2*3=6	3*3=9	4*3=12	5*3=15
1*4=4	2*4=8	3*4=12	4*4=16	5*4=20
1*5=5	2*5=10	3*5=15	4*5=20	5*5=25
1*6=6	2*6=12	3*6=18	4*6=24	5*6=30
1*7=7	2*7=14	3*7=21	4*7=28	5*7=35
1*8=8	2*8=16	3*8=24	4*8=32	5*8=40
1*9=9	2*9=18	3*9=27	4*9=36	5*9=45
1*10=10	2*10=20	3*10=30	4*10=40	5*10=50
6*1=6	7*1=7	8*1=8	9*1=9	10*1=10
6*2=12	7*2=14	8*2=16	9*2=18	10*2=20
6*3=18	7*3=21	8*3=24	9*3=27	10*3=30
6*4=24	7*4=28	8*4=32	9*4=36	10*4=40
6*5=30	7*5=35	8*5=40	9*5=45	10*5=50
6*6=36	7*6=42	8*6=48	9*6=54	10*6=60
6*7=42	7*7=49	8*7=56	9*7=63	10*7=70
6*8=48	7*8=56	8*8=64	9*8=72	10*8=80
6*9=54	7*9=63	8*9=72	9*9=81	10*9=90
6*10=60	7*10=70	8*10=80	9*10=90	10*10=100

for Döngüsü - Ödev

Aşağıdaki görüntüleri verecek C# kodlarını yazınız.

1

2

```
*  
**  
***  
****  
*****  
******  
*******  
*****  
*****  
*****  
*****
```

3

[illegible]

4

5



foreach Döngüsü

Adem AKKUŞ

foreach Döngüsü

Diziler ve listelerde (koleksiyonlar) kullanılır. Özellikle eleman sayısı bilinmediği durumlarda kullanılır. Diziler ve listeler konusunda detaylı anlatılacaktır

foreach(değişken_tipi değişken_adı in liste/dizi)

```
{  
    foreach (string sehir in sehirler)  
    {  
        //işlemler Console.WriteLine($"{sehir} ");  
    }  
}
```

Not: Değişken türü bilinmiyorsa, türü yerine **var** keyword'ü kullanılabilir.

```
foreach (var sehir in sehirler)  
{  
    Console.WriteLine($"{sehir} ");  
}
```


A stylized landscape illustration featuring rolling green hills in the foreground, a small tree with a brown trunk and purple and pink foliage on the left, and blue and white wavy bands representing hills or clouds in the background.

while Döngüsü

Adem AKKUŞ

while Döngüsü

Döngü, koşulu **True** olduğu müddetçe çalışır. Koşul **False** olduğunda döngü sonlanır.
for döngüsünden farklı olarak;

- Sayaç değişkeni döngü dışında tanımlanmalı,
- Artış miktarı döngü içerisinde yapılmalıdır.

sayaç_değişkeni_tanımlama

While(şart)

{

//işlemler

//sayaç artış miktarı

}

while Döngüsü - Örnek

0 – 20 arasındaki sayıları while döngüsü ile yazdıralım.

```
#region ardışık
//ardışık tamsayılar
byte sayac = 0;
while (sayac <=20)
{
    Console.Write(sayac+" ");
    sayac++;
}
#endregion
```

```
C:\Users\PC\source\repos\Hafta6\WhileDongusu_Ornek_1\bin\Debug\net6.0\...
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
```

```
#region ardışık çift
//ardışık çift tamsayılar
byte sayac = 0;
while (sayac <= 20)
{
    Console.Write(sayac + " ");
    sayac += 2;
}
#endregion
```

```
C:\Users\PC\source\repos\Hafta6\WhileDongusu_Ornek_1\bin\Debug\net6.0\...
0 2 4 6 8 10 12 14 16 18 20
```

```
#region ardışık tek
//ardışık tek tamsayılar
int sayac = 1;
while (sayac <= 20)
{
    Console.Write(sayac + " ");
    sayac += 2;
}
#endregion
```

```
C:\Users\PC\source\repos\Hafta6\WhileDongusu_Ornek_1\bin\Debug\net6.0\...
1 3 5 7 9 11 13 15 17 19
```

while Döngüsü - Ödev

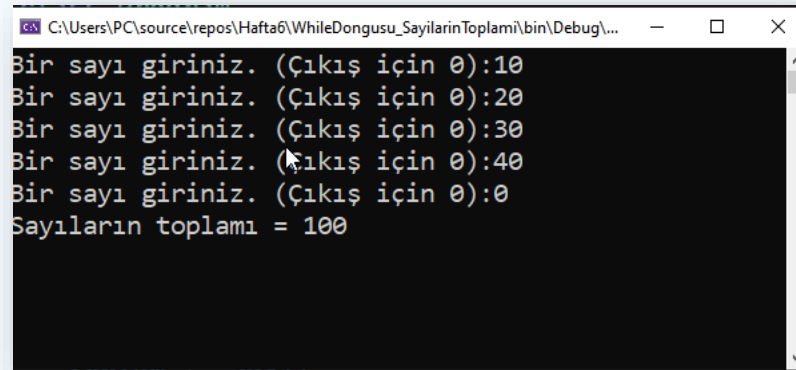
Kullanıcı tarafından girilen üç basamaklı sayının rakamları toplamını ekrana yazdıran program kodlarını yazınız? Örnek 765 sayısı için $7+6+5=18$ yazacak.

```
C:\Users\PC\source\repos\Hafta6\WhileDongusu_Ucbasamakli\bin\Debug\net6..  
Üç basamaklı bir sayı giriniz:345  
345 nin rakamları toplamı:12
```

```
C:\Users\PC\source\repos\Hafta6\WhileDongusu_Ucbasamakli\bin\Debug\net6..  
Üç basamaklı bir sayı giriniz:998  
998 nin rakamları toplamı:26
```

while Döngüsü - Ödev

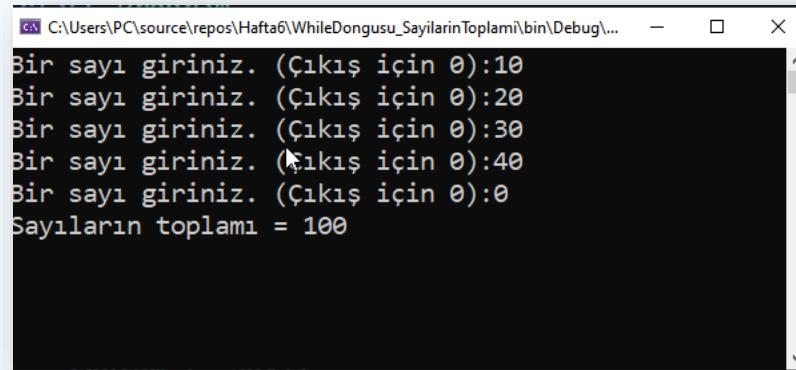
Kullanıcı tarafından sıfır girilene kadar tüm sayıların toplamı ve ortalamasını ekrana yazdıran programı yazınız?



```
C:\Users\PC\source\repos\Hafta6\WhileDongusu_SayilarinToplami\bin\Debug\...  
Bir sayı giriniz. (Çıkış için 0):10  
Bir sayı giriniz. (Çıkış için 0):20  
Bir sayı giriniz. (Çıkış için 0):30  
Bir sayı giriniz. (Çıkış için 0):40  
Bir sayı giriniz. (Çıkış için 0):0  
Sayıların toplamı = 100
```

while Döngüsü - Ödev

Kullanıcı tarafından sıfır girilene kadar tüm sayıların toplamı ve ortalamasını ekrana yazdıran programı yazınız?



```
C:\Users\PC\source\repos\Hafta6\WhileDongusu_SayilarinToplami\bin\Debug\...  
Bir sayı giriniz. (Çıkış için 0):10  
Bir sayı giriniz. (Çıkış için 0):20  
Bir sayı giriniz. (Çıkış için 0):30  
Bir sayı giriniz. (Çıkış için 0):40  
Bir sayı giriniz. (Çıkış için 0):0  
Sayıların toplamı = 100
```

for Döngüsü - while Döngüsü Karşılaştırma

```
for (int i=0;i<=5;i++)  
{  
    //Yapılacak işler  
}
```

```
int i=0;  
While(i<=5)  
{  
    //Yapılacak işler  
    i++;  
}
```

- Döngü değişkenine ilk değer ataması for döngüsünde döngü ifadesinde, while da ise döngüden önce yapılır.
- Döngü değişkeninin değerinin arttırılması, for döngüsünde döngünün ifadesinde, while döngüsünde döngü içerisinde yapılır.

Do while Döngüsü

while döngüsünden farkı, her halükarda döngü en az bir defa çalışır. Şart en sonda kontrol edilir.

```
do
{
    //Yapılacak işler.
    //Döngü değişkenine
    müdahale
}
While (Şart);
```

```
int i=0;
do
{
    i++;
    Console.WriteLine(i);
}
While (i<=10);
```

Genel Çalışma Ödevleri

- Klavyeden girilecek bir sayının asal sayı olup olmadığını ekrana basan bir program yazınız.
- Dört basamaklı bir sayının rakamları ters yazılıp, 4 ile çarpılırsa, çıkan sonuç, kendisiyle aynıdır. Dört basamaklı bu sayıyı bulunuz.
- 3 basamaklı iki adet sayıyı klasik toplama yöntemini kullanarak hesaplayan programı bulunuz.

Genel Çalışma Ödevleri

- . Hipotenüs'u 500 birime kadar olan dik üçgenlerin, kenar uzunluklarını gösteren bir program yazınız. Kenar uzunlukları, sadece tam sayı olacaktır.
- Fibonacci serisinin ilk iki elemanı 0 ve 1'dir. Bundan sonra gelen sayılar, kendisinden önceki iki sayının toplamıyla bulunur.
Yani seri elemanları 0 1 1 2 3 5 8 13 21 ... şeklinde gitmektedir.
Verilecek adım sayısına göre, Fibonnaci serisinin elemanlarını gösterecek bir program yazınız.
- Girilen 4 basamaklı bir sayının okunuşunu veren programı geliştiriniz.