do - while

Bu deyimin **while** döngüsünden farkı koşulun döngü sonunda sınanmasıdır.

```
do  {
…
   Küme;
…
}
while (koşul);
```

Örn: Klavyeden girilen pozitif sayının karekökünü hesaplayan ve devam etmek istiyor musunuz sorusuna **e** veya **E** ile karşılık verildiğinde bu işlemi sürekli yapan C kodunu yazınız.

Örn: Klavyeden gireceğiniz pozitif bir sayının (n basamaklı) basamakları toplamını bulan ve ekranda gösteren C kodunu yazınız. Eğer klavyeden 0 (Sıfır) ya da negatif bir sayı girilirse tekrar sayı girilmesi istenecektir.

Örnek Program çıktıları:

Bir sayı girin : 1905
Basamakları toplamı : 15

Bir sayı girin : -695
Bir sayı girin : -123072
Bir sayı girin : 0
Bir sayı girin : 532904
Basamakları toplamı : 23

for

while ve **do-while** deyimlerinden 2 açıdan farklıdır. İlki döngü sayacı olması ve ona başlangıç değeri verilmesi; ikincisi döngü sayacının artırılmasıdır.

```
for(başlangıç ; koşul ; artım)  {
   …
   Küme;
   …
}
```

Örn: main( )

```
{
      int  i;        // i+=2
      for(i=1; i<=10; i++)
            printf("%d\n", i);

}
```

Çıktı (*)

```
1          1
2          3
3          5
.          7
.          9
10
```

****

Örn: main( )

```
{
      int  i;
      for(i=0; i<=20; i+=2)
            printf("%d\n", i++);

}
```

Çıktı

```
0
3
6
9
12
15
18
```

- 10'dan 1'e doğru geriye doğru sayan for döngüsü… (?)

- Klavyeden girilen bir sayının faktöryelini hesaplayan C programı…

- 1' den klavyeden girilecek n sayısına kadar tüm sayıları toplayan ve sonucu ekranda gösteren C programını yazınız?

  $1 + 2 + 3 + \ldots + n = ?$

- Üç basamaklı pozitif bir tamsayının basamaklarının küpleri toplamı sayının kendisine eşitse o sayıya **Armstrong sayı** denir. Örneğin, $371 = 3^3 + 7^3 + 1^3$ olduğundan 371 sayısı bir Armstrong sayıdır.

  Buna göre, üç basamaklı bütün Armstrong sayıları bulan ve ekranda gösteren C kodunu yazınız.