

# BİLGİSAYAR BİLİMLERİNE GİRİŞ I

-8-

# Soru

2

- New York metropolitanında nüfus 19 milyon, İstanbul'da ise 14 milyon olarak belirleniyor.
- İstanbul nüfusunun yılda %3, New York nüfusunun ise %2 artış göstereceği tahmin ediliyor.
- Buna göre kaç yıl sonra İstanbul'un New York'tan kalabalık olacağını hesaplayan C programını yazınız.

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

```
main()
```

```
{
```

```
    double newyork,istanbul;
```

```
    int yil;
```

```
    newyork=19000000;
```

```
    istanbul=14000000;
```

```
    yil=0;
```

```
    do
```

```
    {
```

```
        newyork=newyork*1.02;
```

```
        istanbul=istanbul*1.03;
```

```
        yil=yil+1;
```

```
    } while(istanbul<newyork);
```

```
    printf("%d",yil);
```

```
}
```

# FOR Döngü Deyimi

3

- Diğer döngü deyimleri gibi bir deyimi birçok kez tekrarlamakta kullanılır.
- Koşul sınanması *while*'da olduğu gibi çevrime girmeden yapılır.
- *while* ve *do...while* deyimlerinden iki açıdan farklıdır:
  - ▣ ilki, döngü sayacı olması ve ona başlangıç değeri verilmesi;
  - ▣ ikincisi döngü sayacının arttırılmasıdır.

Genel yazım biçimi,

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

# Örnek: 1-10 arasındaki sayıların yazdırılması

4

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

```
main()
```

```
{
```

```
    int i;
```

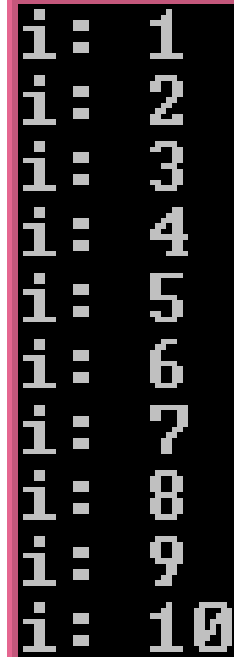
```
    for(i=1;i<=10;i++)
```

```
    {
```

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

```
    }
```

```
}
```



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

# Örnekler

5

- 1'den 100'e kadar birer birer arttırma

**for** ( **i=1; i<=100; i++** )

- 100'den 1'e kadar birer birer azaltma

**for** ( **i=100; i>=1; i--** )

- 7'den 77'ye kadar yedişer yedişer arttırma

**for** ( **i=7; i<=77; i+=7** )

- 2, 5, 8, 11, 14, 17, 20 değerlerini alacak biçimde değiştirme

**for** ( **j=2; j<=20; j+=3** )

# Örnek: Başlangıç bitiş değişkenleri arasındaki sayıların toplamı

6

```
#include<stdio.h>
main()
{
    int s, baslangic, bitis, top=0;
    printf("Baslangic sayisini giriniz:");
    scanf("%d", &baslangic);
    printf("Bitis sayisini giriniz:");
    scanf("%d", &bitis);

    for(s=baslangic; s<=bitis; s++)
    {
        top=top+s;
    }
    printf("Toplam=%d",top);
}
```

# Örnek: $n!$ bulduran C programı

7

```
#include<stdio.h>
main()
{
    int f=1,i,n;

    printf("Faktoriyeli alinacak sayiyi giriniz:");
    scanf("%d",&n);

    for(i=1;i<=n;i++)
    {
        f=f*i;
    }
    printf("Faktoriyel=%d",f);
}
```

# Alıştırmalar

8

- 1'den 100'e kadar olan sayıların toplamını ve ortalamasını bulduran programı yazınız.
- 1'den 100'e kadar olan *tek* sayıların toplamını ve ortalamasını bulduran programı yazınız.
- 1'den 100'e kadar olan *çift* sayıların toplamını ve ortalamasını bulduran programı yazınız.
- 1'den başlayıp toplamı 300'ü geçene kadar ardışık sayıların toplamını bulunuz.
- Girilen bir sayının tam bölenlerini bulan programı yazınız.
- $a^b$  değerini hesaplayan programı yazınız.



# İç içe döngüler

9

- Bir döngü içerisinde başka bir döngü bulunuyorsa, bu tür yapılara *İç-içe döngüler* denir.
- Bu durumda içteki döngü dıştaki döngünün her adımında yeniden çalıştırılacaktır.

# Örnek

10

```
#include <stdio.h>
main()
{
    int i,j,s=0;

    for(i=1;i<=4;i++)
    {
        for(j=1;j<=3;j++)
        {
            s=s+1;
        }
    }
    printf("Sonuc=%d",s);
}
```

Dıştaki döngü 4 kere tekrarlanacaktır.

İçteki döngü de dıştaki döngünün her tekrar edilişinde 3 kere tekrar edilecektir.

Böylece içteki döngünün gövdesini oluşturan atama ifadesi 12 kere tekrar edilmiş olacaktır.

Dıştaki *for* döngüsünden çıkıldığında, s değişkeni içerisindeki değer 12 olacaktır.

# Alıştırmalar

11

- Aşağıdaki değerleri hesaplayan programı yazınız.

$$t = \frac{1}{1} + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} \dots + \frac{1}{78}$$

$$t = \frac{1}{1} - \frac{1}{2} + \frac{1}{3} - \frac{1}{4} + \dots - \frac{1}{78}$$

- pi değerinin hesaplanması

$$\frac{\pi^2}{6} = \frac{1}{1^2} + \frac{1}{2^2} + \frac{1}{3^2} + \frac{1}{4^2} + \dots$$

# Alıştırmalar

12

- Aşağıdaki matrisleri oluşturan programı yazınız

1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1

1	1	1	1	1
2	2	2	2	2
3	3	3	3	3
4	4	4	4	4
5	5	5	5	5

1	2	3	4	5
1	2	3	4	5
1	2	3	4	5
1	2	3	4	5
1	2	3	4	5

0	1	1	1	1
1	0	1	1	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

1	1	1	1	0
1	1	1	0	1
1	1	0	1	1
1	0	1	1	1
0	1	1	1	1

0	0	0	0	0
1	0	0	0	0
1	1	0	0	0
1	1	1	0	0
1	1	1	1	0

# break Deyimi

13

- **while, for, do/while** ve **switch** ile kullanıldığında o yapıdan çıkış sağlar.
- Program yapıdan sonraki ilk ifadeyi çağırarak devam eder.
- En genel kullanımı bir döngüden istenen anda çıkmak ve döngünün belli bir kısmından çıkmaktır.

# Örnek:

14

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

```
main()
```

```
{ int x;
```

```
  for ( x = 1; x <= 10; x++ ) {
```

```
    if ( x == 5 )
```

```
      break;
```

```
      printf( "%d ", x );
```

```
  }
```

```
  printf( "\nDonguden x= %d oldugunda cikildi\n", x );
```

```
}
```

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

# continue Deyimi

15

- **while, for** ve **do/while** yapıları içinde çalıştığında döngü gövdesinin kalan kısmı atlanır.
- **while** ve **do/while** yapılarında döngü koşulu hemen kontrol edilir.
- **for** yapısında artırma deyimi çalıştırılır daha sonra ise döngü koşulu kontrol edilir.

# Örnek

16

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

```
main()
```

```
{
```

```
    int x;
```

```
    for ( x = 1; x <= 10; x++ ) {
```

```
        if ( x == 5 )
```

```
            continue;
```

```
        printf( "%d ", x );
```

```
    }
```

```
    printf( "\ncontinue, 5 degerinin atlanmasi icin kullanildi\n" );
```

```
}
```

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



# Uygulama 1

17

- ❑ **Kendisi hariç** bütün pozitif tamsayı çarpanlarının toplamı kendisine eşit olan sayılara **mükemmel sayı** denir.
- ❑ Örneğin, 28 sayısının kendisi hariç pozitif çarpanları toplamı  $1+2+4+7+14 = 28$  olduğundan 28 sayısı mükemmel bir sayıdır.
- ❑ Buna göre klavyeden girilen bir sayının mükemmel sayı olup olmadığını söylenen C programını yazınız.

# Çözüm

18

```
#include<stdio.h>
main()
{
    int sayi, i, toplam=0;
    printf("Bir sayi giriniz : ");
    scanf("%d",&sayi);
    for(i=1; i<sayi; i++)
    {
        if(sayi%i == 0) toplam += i;
    }
    if(toplam == sayi)
        printf("Mukemmel sayidir.");
    else
        printf("Mukemmel sayi degildir.");
}
```

# Uygulama 2

19

- Üç basamaklı **rakamları birbirinden farklı** tüm sayıları ekranda gösteren ve bu kurala uygun kaç tane sayı olduğunu söyleyen C programını yazınız.
- **Örnek çıktı :** 102 103 104 105 106 107 108 109 120  
123 . . . . . 980 981 982 983 984 985 986 987  
Bu kurala uygun 648 sayı vardır.

# Çözüm

20

```
#include <stdio.h>
main()
{
    int i, a, b, c, sayac=0;
    for(i=100;i<=999;i++)
    {
        a = i/100;          // yüzler basamağı
        b = (i%100)/10;     // onlar basamağı
        c = i%10;           // birler basamağı
        if(a!=b && a!=c && b!=c)
        {
            printf("%5d",i);
            sayac++;
        }
    }
    printf("\n\nBu kurala uygun %d sayi var" ,sayac);
}
```

# Uygulama 3

21

- Aşağıdaki programın çıktısını yazınız.

```
#include <stdio.h>

main()
{
    int i, sayi=48, a=350;
    for(i=2; i<8; i++) {
        if(sayi%i == 0) continue;
        a /= i;
    }
    printf("%d", a);
}
```

# Çözüm

22

i	sayi	a
2	48	350
3	48	350
4	48	350
5	48	70
6	48	70
7	48	10 (Döngü bitti)

Çıktı = 10

# Uygulama 4

23

- Aşağıdaki programın çıktısını yazınız.

```
#include <stdio.h>

main()
{
    int i, sayi=28, x=3;
    for(i=13; i>0; i-=2) {
        if(sayi%i == 0) break;
        x += i;
    }
    printf("%d", x);
}
```

# Çözüm

24

i	sayi	x
13	28	3
11	28	16
9	28	27
7	28	36 (Döngüden çık)

Çıktı = 36



# Uygulama 5

25

- Aşağıdaki programın çıktısını yazınız.

```
main()
{
    int i, j, t=0;
    for(i=1; i<6; i+=2) {
        for(j=4; j>0; j-=2) {
            if(i>j) t+=i;
        }
    }
    printf("%d", t);
}
```

# Uygulama 5

26

- Aşağıdaki programın çıktısını yazınız.

```
main()
{
    int i, j, t=0;
    for(i=1; i<6; i+=2) {
        for(j=4; j>0; j-=2) {
            if(i>j) t+=i;
        }
    }
    printf("%d", t);
}
```

Çözüm: 13

# Sorular???

