

4. DÖNGÜLER (7nci hafta)

Bir problemin çözümünde işlem veya işlemler birden fazla tekrarlanarak yapılacaksa bilgisayar programınızda döngüler kullanmak yararlı olacaktır (kendisini tekrarlayan program parçaları veya bölgeleri). Örnek olarak ortalama almak için sayıları toplamak, faktöriyel hesabı, katsayıları değiştirilen polinomların köklerini bulmak, eğri uydurma yani fit işlemleri yapılırken verilerin değerlendirilmesi, integral almak için toplamalar yapmak, diferensiyel hesabı yapılırken kendini tekrar eden işlemler, serbest düşen bir cismin zamana bağlı konumunu belirlemek zamanın artırılması, bir ortamda üreyen bakterilerin benzetişimini yaparken rasgele sayı türetilmesi, sayısal analizler yaparken sonuçların değerlendirilmesi, otel veya uçak için rezervasyon işlemleri yaparken isim, gidiş-dönüş tarihleri gibi bilgilerin değişik müşteri için yeniden yapılması, fatura girişi yapmak, depodaki ürünlerin takibi gibi işlemler tekrarlamalar/döngüler gerektirir. Bu amaç için değişik özelliklere sahip döngüler kullanılır. Bu döngüler aşağıda anlatılmaktadır.

4.1 for DÖNGÜSÜ

Tekrarlanacak işlem sayısı rakamsal olarak belli ise programda `for` döngüsü kullanılabilir.

de C++ da kullanılan `for` döngüsü şematik olarak gösterilmektedir.

de yuvarlaklar içindeki rakamlar (1) döngünün başlangıcını, (2) bitişini ve (3) ise adımlama değerlerini göstermektedir. Döngünün devam edip etmeyeceği (2) nolu kısımda kontrol yapılarak karar verilir. İşleme devam edilecekse yani sorgulamadaki koşul sağlanıyorsa veya olumlu (doğru) cevap alınıyorsa (4) nolu bölgedeki (döngü içindeki) işlemler yapılır ve (3) nolu aşamaya geçilerek (1) bölgesindeki sayaç değişkeninin adım miktarı kadar artırılır. Döngü koşul sağlandığı sürece devam eder. Koşul sağlanmıyorsa yani (2) nolu kısımdan hatalı sonuç elde ediliyorsa döngüden çıkılarak (5) nolu kısma geçilir. Yazım formatı aşağıdaki gibidir:

Şekil 4-1 for döngüsünün çalışması.

```
for (başlangıç değeri; devam etme şartı; artırma değeri)
```

Örnek 4-1 1 den 10 a kadar rakamları sırayla yan yana yazdırabilmek için for döngüsü kullanılabilir. C programı aşağıda verilmektedir. Burada döngünün başlangıç ve bitiş değerleri (1 ve 10) bilinmektedir. Koşul ise sayacın 10 dan küçük ve eşit olması durumudur.

Program 4-1 1 den 10 a kadar rakamların for döngüsü içinde yazdırılması.

```
/* for komutunun kullanıldığı örnek program */
#include <stdio.h>
main() { int dongu;
    for(dongu=1; dongu<=10; dongu=dongu+1) printf("%3d ", dongu );
    printf("\n");}
```

Yukarıdaki örnek programın çıktısı aşağıdaki gibidir:

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

Örnek 4-2 ($\sum_{i=1}^4 i^2$) ifadesini hesaplayan algoritma ve bir C++ programı aşağıda verilmektedir.

Algoritma 4-1 ($\sum_{i=1}^4 i^2$) ifadesini hesaplayan algoritma.

1. Başla
2. int i, t=0, n=4;
3. for (i=1; i<=4; i++) t = t + i*i;
4. Yaz t;
5. Son

Program 4-2 ($\sum_{i=1}^4 i^2$) ifadesininin hesaplanması.

```
/* i*i nin toplamı */
#include <stdio.h>
#include <conio.h>
main() { int i, t=0;
    for(i=1; i<=10; i=i+1) t=t+i*i;
    printf("\n Toplam = %4d", t);
    getch();
    return 0;}
```

Yukarıdaki örnek programın çıktısı aşağıdaki gibidir:

```
Toplam = 385
```

4.1.1 İç içe for döngülerinin kullanımı

Bazı durumlarda iç içe geçmiş for döngülerinin kullanılması gerekmektedir. Aşağıdaki örnekte sin x fonksiyonunun seri açılımını kullanarak değerinin hesaplanması iç içe geçmiş for döngüleri ile yapılmaktadır.

Örnek 4-3 $\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \frac{x^9}{9!} - \dots$ ifadesinin hesaplanması için iç içe for döngüleri kullanılabilir. Eşitliğin sağındaki ifade pay ve paydalardan oluşmaktadır. Paydaki değerler x in kuvvetininin 1, 3, 5, ... şeklinde değiştiğini, paydadaki değerler 1!, 3!, 5!, vs şeklinde değiştiğini ve

terimlerin işaretlerinin sırası ile +, - şeklinde değiştiğini göstermektedir. Buradan yukarıdaki eşitliğin sağındaki işlemler için döngü kullanılabileceği anlamının çıkarılabileceğidir. Paydalardaki faktöriyeler önce hesaplanabilir (en içteki döngü bu olabilir), daha sonra bölme işlemine ve terimlerin toplanmasına geçilebilir. Aşağıdaki algortimada bu işlemler (for döngüsü kullanılarak) gösterilmektedir. İçteki for döngüsü faktöriyeli hesaplamakta ve bu döngü bitince bölme ve toplama işlemlerine (terimlerin işaretleri değiştirilerek) geçilmektedir. Programdada for döngüsü kullanılarak benzer işlemler yapılmaktadır.

Algoritma 4-2 $\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \frac{x^9}{9!} - \dots$ ifadesini hesaplayan algoritma.

```
1. Başla
2. int f, i, n=1, k=10;
3. float a, x, t=0;
4. Yaz " Açığı derece cinsinden giriniz : "; a;
5. x=3.14159*a/180.0;
6. for (n=1; n<k; n++) {
    f=1;
    for (i=1; i<=(2*n-1); i++) f=f*i;
    t=t+pow(-1, n-1)*pow(x, 2*n-1)/(float)f;
}
7. Yaz a, t, sin(x);
8. Son
```

Program 4-3 $\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \frac{x^9}{9!} - \dots$ ifadesini hesaplayan C++ programı.

```
#include <iostream>
#include <math.h>
#include <conio.h>
using namespace std;
main (){
    int n, // seri deęiskeni
        k=5, // seri üst sınır deęeri
        i;
    long int f; // faktöriyel hesabı
    float a, t=0, x;
    cout <<" n 0 ile 90 derece arasında açı deęeri giriniz : " << endl;
    cout <<" Bu deęerlerin dışında pogram sonlanır " << endl;
    cin >> a;
    x=a*3.14159/180.; /* Dereceyi radyana dönüştür */
    t = 0; // toplam deęiskeni
    k = 10; // ust sinir
    for (n=1; n<k; n++)
```

```

    {
        f = 1;
        for (i = 1; i<=2*n-1; i++ ) f = f * i;
        t=t+pow(-1, n-1)*pow(x, 2*n-1)/(float) f;
    }
cout << endl;
cout << "sin(" << a << ")=" << t << ", hazır fonksiyon=" << sin(x)<<
endl;
getch(); return 0;
}

```

Programın çalıştırılması sonucu elde edilen ekran çıktısı aşağıdaki gibidir:

n 0 ile 90 derece arasında açı değeri giriniz :

Bu değerlerin dışında program sonlanır

34

sin(34)=0.559192, hazır fonksiyon=0.559192

Örnek 4-4 $(\sum_{i=1}^4 \frac{1}{i^2})$ ifadesinin hesaplanması için sadece 1 tane for döngüsü kullanılması yeterli olacaktır. Yapılacak işlem, 1 den 4 e kadar olan tam sayıların önce karelerini alıp sonra bunların terslerini alıp toplamak şeklinde olacaktır. Aşağıdaki programda bu işlem (for döngüsü kullanılarak) yapılmıştır.

Program 4-4 $(\sum_{i=1}^4 \frac{1}{i^2})$ ifadesini hesaplayan program.

```

#include <stdio.h>
#include <conio.h>
main() { int i, n=4;
        float t=0;
        for(i=1; i<=n; i++) t=t+1.0/((float) i*i);
        printf("Sonuç = %2.3f ", t);
        getch();
        return 0;}

```