

3. VERİ GİRİŞ-ÇIKIŞI (4ncü hafta)

Bilgisayar belleğine veri aktarılabilmesi (veri okutulması) için `scanf()` veya `cin` (consol input un kısaltılmışı) komutları kullanılabilir. Sonuçlarının ekrana aktarılması (veri yazdırılması) için `printf()` veya `cout` (consol out un kısaltılmışı) komutları kullanılabilir.

3.1 `printf()` ve `cout` FONKSİYONLARI

Bu komutlar (C/C++) standart çıkış aygıtına (burada genellikle ekrana) veri veya mesaj yollamak amacıyla kullanılır. Yazım biçimleri

```
printf(format stringi, değer listesi);
```

ve

```
cout << string << veri;
```

şeklinde. Burada yazılan format stringi, değer listesi, string veya veri ilerleyen kısımlarda anlatılacaktır. Yazılan programlarda `printf()` fonksiyonu için `<stdio.h>` ve `cout` fonksiyonu için `<iostream.h>` header dosyalarının kullanılması gerekmektedir. Çizelge 3-1 ve Çizelge 3-2 de `printf` ve `cout` komutları ile kullanılacak Escape karakterleri ve biçimlendiricileri verilmektedir.

Çizelge 3-1 C/C++ ta `printf` ve `cout` ile kullanılan escape karakter komutları.

Escape Karakter		Escape Karakter	
<code>\n</code>	Newline (NL veya LF)-yeni satır	<code>\\</code>	Backslash (\)-bu simgeyi çıkartmak için kullanılır
<code>\t</code>	Horizontal tab (HT)-yatay kaydırma	<code>\?</code>	Soru işareti (?)-soru işaretini çıkartmak için kullanılır
<code>\v</code>	Vertical tab (VT)-düşey satır atlatma	<code>\'</code>	Tek tırnak (single quote, ')-bu simgeyi ekranda çıkartmak için kullanılır
<code>\b</code>	Backspace (BS)-geriye giderek silme	<code>\"</code>	Çift tırnak (")
<code>\r</code>	Carriage return (CR)-yeni satır	<code>\ooo</code>	Sayısal sekizli kod <i>ooo</i>
<code>\f</code>	Süreklî form (FF)-1 sayfa atlatma	<code>\xhhh</code>	Sayısal onaltılı kod <i>hhh</i>
<code>\a</code>	Alarm (BEL)-zil sesi		

Çizelge 3-2 `printf` komutunda yazım biçiminin ayarlanması.

<code>%f</code>	Kesirli veya kayan noktalı sayılar için kullanılır (float)
<code>%c</code>	Karakter değişkenler ve sabitler için kullanılır
<code>%s</code>	String (çoklu karakter grubu)
<code>%d</code>	Desimal, işaretli tam sayı
<code>%</code>	Yüzde işaretini ekrana yazdırmak için
<code>%10.4f</code>	Noktanın solunda 10 hane, noktanın sağında 4 hane ayrılacak.
<code>%5.7s</code>	Noktanın solunda 5 hane, noktanın sağında 7 hane ayrılır
<code>%ld</code> veya <code>%li</code>	Uzun tam sayılar için kullanılır
<code>%lf</code>	Uzun kesirli sayılar için kullanılır

Örnek 3-1 `printf("a sayısı %4.2f dir.\n", a);` komut satırında çift tırnaklar içindeki "a sayısı %4.2f dir.\n" format string'idir. "a sayısı", ekrana yazdırılacak

olan açıklama kısmıdır. %4.2f, ekrana yazdırılacak rakamın çıkış formatı yani görünüşüdür. Ekrana yazılacak sayının tipi kesirli (floating point) sayı olduğundan tam sayı kısmı 4 basamaklı olup, ondalık kısmına 2 hane ayrılmıştır. Sayı, "a sayısı" ifadesi yazıldıktan sonra ekrana gelecektir. Ekrana a değişkenindeki değer yazılacaktır. Ardından "dir." yazısı ekrana gelecektir. \n, (enter) ifadesi (Escape karakteri) işlendikten sonra imleç bir alt satıra geçer ve imleç satırın başına konumlandırılır.

Algoritma 3-1 Bir değişkendeki değerın yazdırılması.

1. Başla,
2. tek duyarlı a=102.345,
3. a değişkenindeki değeri yaz,
4. Son.

Program 3-1 Değişkendeki değerin ekrana yazdırılması.

```
#include <stdio.h>
#include <conio.h>
main(){ float a=102.345;
    printf("a sayısı %4.2f dir.\n", a);
    printf("\n\n bir tuşa basınız ");
    getch(); return 0;
} // program sonu
```

Programın çalıştırılması sonucu ekran çıktısı aşağıdaki gibidir:

```
a sayısı 102.35 dir.
bir tuşa basınız
```

Örnek 3-2 Derece biriminde verilmiş bir açı değerinin radyan açı değerine dönüştürülmesini ve sonucun ekrana yazdırılması sağlayan (algoritma ve) program aşağıdaki gibi olabilir.

Algoritma 3-2 Derece cinsinden verilen açı değerinin radyan açı değerine dönüştürülmesi.

1. Başla,
2. RADYAN=3.14159/180.0
3. x i tanımla;
4. y yi tanımla;
5. x = 90.0;
6. y = RADYAN * x;
7. x, y değişkenlerindeki değerleri yaz;
8. Son.

Program 3-2 Derece-radyan aç dönüşümü.

```
/* Derece açsının radyan açsına dönüştürülmesi
   define deyiminin kullanılması */
#include <stdio.h>
#include <conio.h>
#define RADYAN 3.14159/180.0 /* radyan tanımlanır */
main()/* program başlangıcı */
{ float x, /* derece deęişkeni */
  y; /* radyan deęerini saklayacak deęişken */
  x = 90.0;
  y = RADYAN * x; /* derece -> radyana cevrilir */
/* ekrana mesaj yazılır */
  printf("%6.4f derece = %6.4f radyan \n", x, y);
  printf("\n bir tuşa basınız ");/* bir tuşa basılması beklenir */
  getch();
  return 0;
} // program sonu
```

Programın çalıştırılması sonucu ekran çıktısı aşağıdaki gibidir:

```
0.0000 derece = 1.5708 radyan
bir tuşa basınız
```

Yukarıdaki programda printf komutu ile çift tırnaklar içindeki "%6.4f derece = %6.4f radyan \n", x, y ile ekrana mesaj yazdırılmıştır. İlk %6.4f formatı ile ekrana yazılacak (burada x deęişkenindeki sayının tam kısmı için 6 hane ayrılır. Sayının kesirli kısmı dört haneli biçimde ekrana yazdırılır. y deęişkenindeki sayının deęeri %6.4f formatı ile ekrana yazdırılır. Daha sonra ekrana bir tusa basınız şeklinde mesaj gelir. Bu satırın amacı Windows işletim sisteminde çalışan DOS tabanlı bu programın ekranda görünmesini bir tuşa basana kadar beklemesini sağlamaktır. Bu komut fonksiyonu #include <conio.h> içerik dosyasından sağlanmaktadır.

Örnek 3-3 Fahrenheit sıcaklık skalasından Celcius (Santigrad) skalasına $C=(F-32)/1.8$ formülü ile dönüşüm için hazırlanmış (algoritma ve) bilgisayar programı aşağıda verilmektedir.

Algoritma 3-3 Fahrenheit sıcaklığının Celcius skalasına dönüştürülmesi.

1. Başla,
2. f=55
3. $c = (f-32.0)/1.8;$
4. f ve c deęişkenlerindeki deęerleri yazınız
5. Son.

Program 3-3 Fahrenheit-Santigrat dönüşümü.

```
/* Fahrenheit-Santigrat dönüşümü */
#include <stdio.h> /* standart giris cikis */
#include <conio.h> /* konsol giriş/çıkışı */
main()
{ float f, c; /* değişkenler tanımlanır */
  f = 55; /* Fahrenheit skalasındaki sıcaklık değeri */
  printf("\n");
  c = (f-32.0)/1.8; /* derece skalasındaki sıcaklık değeri */
  printf("%4.0f Fahrenheit=%6.1f Santigrad\n", f, c);
  getch();}
```

Programın çalıştırılması sonucu ekran çıktısı aşağıdaki gibidir:

```
55 Fahrenheit= 12.8 Santigrad
```

Yukarıdaki programda standart giriş-çıkış işlemlerini yapabilmek için #include <stdio.h> satırı yazılmıştır. Daha sonra main() ile ana programın başladığı belirtilmiştir. Fahrenheit ile Celcius sıcaklık skalaları arasında dönüşüm yapabilmek için tek dıyarlı gerçel değişkenler tanımlanmış ve f=55; satırı ile 55 değeri bilgisayar belleğine değişken aracılığı ile aktarılmıştır. printf("\n"); ile ekranda imleç bir satır aşağı düşer. Sonra Fahrenheit skalasından Celcius skalasına geçiş c=(f-32.0)/1.8; işlemi ile yapılır ve ekrana değişkenlerdeki değerler %6.1f ve %4.0f formatı ile yazdırılıp program sonlandırılır.

Örnek 3-4 Aşağıdaki algoritmada ve C++ programında 10 tabanındaki bir sayı klavyeden veri olarak bilgisayar belleğine aktarılmakta ve ekrana 16 lı sayı sisteminde yazdırılmaktadır. Burada C++ daki hazır fonksiyon kullanılmıştır.

Algoritma 3-4 Sayı sistemleri arasındaki dönüşüm.

1. Başla,
2. f ve g değişkenlerine 10 tabanına göre sayı giriniz,
3. f ve g yi 16 lı sayı sistemine dönüştür,
4. sonuçları yaz,
5. Son.

Program 3-4 10 tabanlı sayı sisteminden 16 tabanlı sayı sistemine geçiş.

```
// C++ : 10 tabanına göre girilen sayıların
// onaltılı sayı sistemine dönüştürülmesi
#include <iostream> // standart giriş/çıkış fonksiyonları
#include <iomanip>
```

```

#include <conio.h> // konsol giriş/çıkış fonksiyonları
using namespace std;
int main(){ int f, g; // degiskenler tanimlanir
    cout << "iki tane on tabanına göre sayı giriniz    :";
    cin >> f >> g; // deęişkenlere deęerler girilir
    cout << "\n onaltılı sayı sistemi esdegerleri :";
    cout.setf(ios::showbase); // veya cout << showbase; yazılabilir
    cout << hex << setw(15) << f << hex << setw(15) << g << "\n";
    getch(); // bir tusa basılması beklenir
} // program sonu

```

Programın çalıştırılması sonucu ekran çıktısı aşağıdaki gibidir:

```
iki tane on tabanına göre sayı giriniz    :24 8
```

```
onaltılı sayı sistemi esdegerleri :           0x18           0x8
```

Yukarıdaki programda standart giriş-çıkış işlemleri için kullanılmak üzere çeşitli fonksiyonları barındıran `#include <iostream.h>` ve girilen sayıları 16 lı sayı sistemine dönüştürmek için `#include <iomanip.h>` içerik dosyaları programın başlangıç satırlarına yazılmıştır. Bu içerik dosyaları ile ekrana mesaj yazdırma komutu olan `cout <<` ve klavyeden veri girişini sağlayacak olan `cin >>` komutları kullanılabilir hale getirilmiştir. Program satırlarındaki `//` simgeleri ile yapılan açıklamalara dikkat ediniz. `cout << "\n "` komutu imlecin yani mesajın bir alt satıra yazdırılması sağlanmıştır. Program ekrana `cout << "\n onaltılı sayı sistemi esdegerleri :"` ve `cout << hex << f << " " << g;` satırları ile ilgili mesajları yazdırdıktan sonra sonlanmaktadır.

Örnek 3-5 Dünya çevresinde belirtilen yükseklikte dairesel bir yörüngede dolanan bir uydunun dönme periyodunu $T = 2\pi \sqrt{\frac{R}{g}} \left(1 + \frac{h}{R}\right)^{3/2}$ formülüne göre hesaplayan algoritma ve bilgisayar programı aşağıda verilmektedir.

Algoritma 3-5 Uydunun periyodu.

1. Başla,
2. `PI = acos (-1.);`
3. `R = 6.378e6 metre;`
4. `g = 9.8 metre/s2;`
5. `h = 150 km`
6. `h = h * 1000 metre;`
7. `a = 2 * PI;`
8. `b = sqrt (R / g); // (Re/g) nin karekökü`
9. `c = 1. + h / R;`

```
10. d = pow (c, 1.5); // c nin 3/2 nci kuvveti
11. T = a * b * d; // periyot saniye cinsinden
12. Yaz T;
13. Son.
```

Program 3-5 Uydunun dünya çevresindeki dönme periyodunun hesaplanması.

```
#include <iostream> // C++ standart giriş/çıkış fonksiyonları
#include <iomanip> // giriş/çıkış manipülasyon fonksiyonları
#include <cmath> // kompleks matematik fonksiyonları
#include <stdio.h> // C standart giriş/çıkış
#include <conio.h> // C standart giriş/çıkış
using namespace std; // içerik dosyalarının uzantıları yazılmaz.
int main () { // ana programın başladığı satır
    const double PI = acos (-1.); // Pi sabiti tanımlanır
    const double R = 6.378e6; // yerkürenin yarıçapı
    const double g = 9.8; // yerçekimi ivmesi
    double h; // 1000 m yi km çevirme
    double T; // bu programda nesne periyot değişkenidir.
    // saniye birimindedir.
    // çıkış formatı belirlemek için aşağıdaki satırlar yazılmıştır
    cout << fixed << setprecision (2);
    cout << "Uydu yüksekliğini Km cinsinden giriniz : ";
    cin >> h;
    h = h * 1000; // yükseklik metreye dönüştürülür
    // eşitlikteki diğer terimler hesaplanır
    double a = 2 * PI;
    double b = sqrt (R / g); // (R/g) nin karekökü
    double c = 1. + h / R;
    double d = pow (c, 1.5); // c nin 3/2 nci kuvveti
    // periyot hesaplanır
    T = a * b * d; // periyot saniye cinsinden
    // sonuçlar Km ve saniye olarak verilir
    cout << "\n" << h/1000 << "kilometre yükseklikteki \n"
        << "uydu dünya çevresindeki 1 turunu " << endl;
    cout << setprecision (4) << T << " saniyede tamamlar.";
    // setprecision(4) virgülden sonraki rakamların
    // hane sayısını belirler
    cout << endl;
    system("PAUSE"); // programın windows ekranında kalması sağlanır
    return 0;}
```

Programın çalıştırılması sonucu ekran çıktısı aşağıdaki gibidir:

Uydu yüksekliğini Km cinsinden giriniz : 40

40.00kilometre yükseklikteki
uydu dünya çevresindeki 1 turunu
5116.6049 saniyede tamamlar.
Press any key to continue . . .

Örnek 3-6 Biçimli (formatlı) giriş-çıkış işlemleri için aşağıdaki programda iki sayının çarpımı ve çarpım sonuçları biçimlendirilmiş şekilde verilmektedir.

Algoritma 3-6 İki sayının çarpımı.

1. Başla,
2. rakam1=5
3. rakam2=10,
4. toplam = rakam1 * rakam2;
5. toplam değişkenindeki değeri yazınız.
6. Son

Program 3-6 Biçimli giriş-çıkış.

```
#include <iostream> // standart giriş çıkış işlemleri için
#include <iomanip> // formatlı ekran çıktısı için
using namespace std;
int main () { // program başlangıcı
    int rakam1; // adet verisi için değişken tanımlanır
    float rakam2, toplam; // değişkenlerin tanımlanmaları
    cout << "Adet sayısını giriniz :";
    cin >> rakam1; // klavyeden veri girişi yapılır
    cout << "Birim fiyatını giriniz :";
    cin >> rakam2; // klavyeden veri girişi yapılır
    toplam = rakam1 * rakam2;
    cout << fixed << rakam1 << " adet x ";
    cout << setprecision (3) // bundan sonra yazılacak
    // kesirli sayılardaki
    // virgülden sonraki basamak sayısı belirlenir
    << setw (4) // bundan sonra gösterilecek değer için
    //(rakam2 için 4 hanelik) genişlik değeri belirtilir
    << rakam2 << " TL = " << setw (10) << toplam
    << " TL " << endl;
    return 0; //işletim sistemine dönüşü sağlayan satır
} // program sonu
```

Örnek 3-7 Aşağıdaki algoritma ve C programı bir dairenin alanını (πr^2 formülüne göre) hesaplar. Programda ilk satırlarında kullanılacak fonksiyonlara sahip olan içerik dosyaları belirtilmiş ve ayrıca

PI sabiti tanımlanmıştır. Bu kısımdaki tanımlamalar yani değişken tanımlamaları genel (global) amaçlı tanımlamalardır. Global değişkenler ana program ve alt programlar içinde her yerde kullanılabilir. Ayrıca tanımlanmalarına gerek kalmaz. Programda yarıçap değerine göre dairenin alanı hesaplanmıştır. Program ekrana uygun mesajları yazdıktan sonra sonlanır. Ayrıntılar için açıklama satırlarını okuyunuz.

Algoritma 3-7 Dairenin alanının hesaplanması.

1. Başla,
2. $PI = 3.1415926$,
3. $yaricap=2.0$,
4. $alan=PI*yaricap*yaricap$
5. alan değişkenindeki değeri yazınız
6. Son

Program 3-7 Dairenin alanını hesaplanması.

```
/* program dairenin alanını hesaplar */
/* önişlem komutları aşağıda verilmektedir */
#include <stdio.h> /* standart giriş/çıkış */
#include <math.h> /* matematiksel fonksiyon kullanımı */
#include <conio.h> /* getch(); fonksiyonunu kullanmak için */
#include <float.h> /* yüksek duyarlı sayı kullanımı */
/* sabit değerlerin tanımlanması */
const double PI = 3.1415926;
int main(void) /* main fonksiyonun başladığı belirtilir */
{ /* yarıçap ve alan değişkenleri deklare edilir */
    double yaricap, alan;
    yaricap = 2.0; /* yaricap değişkenine değer atanır */
    alan = PI*pow(yaricap, 2); /* daire alanı hesaplanır */
    /* ekrana bilgilerin yazdırılması */
    printf("r =%f, Alan = %f birim kare, \n", yaricap, alan);
    getch();
    return 0;
} /* programın sonlandırılması */
```

Örnek 3-8 Aşağıdaki algoritma ve C programı bir kürenin hacmini ($\frac{4}{3}\pi r^3$) formülüne göre hesaplar. pi sabiti include ifadelerinden hemen sonra tanımlanmıştır.

Algoritma 3-8 Küre hacminin hesaplanması.

1. Başla
2. $\pi = 3.1415926$;
3. YariCap deęişkenine deęer giriniz;
4. $V = (4.0/3.0) * \pi * \text{YariCap} * \text{YariCap} * \text{YariCap}$;
5. V deęişkenindeki deęeri yazınız;
6. Son

Program 3-8 Kürenin hacminin hesaplanması.

```
#include <stdio.h>
#include <conio.h>
#define pi 3.14159
int main(void)
{ int YariCap; float v;
printf("Yarı çap :");scanf("%d",&YariCap);
v=(4/3)*pi*YariCap*YariCap*YariCap;
printf("Hacim : %f birim^3\n", v);
getch();}
```

Örnek 3-9 Aşağıdaki C++ programında bir verinin (burada π) biçimli çıktısı örneęi verilmektedir.

Program 3-9 C++ biçimli giriş-çıkış işlemi.

```
// #include "stdafx.h" // MS Visual Studio 2005
#include <iostream> //
#include <iomanip> //
#include <fstream> //
#include <conio.h>
using namespace std; // yukarıdaki header dosyalarının
// uzantılarını yazmaya gerek kalmaz
int main(){ double pi=3.14159265358;
cout << setprecision(5) << setw(10);
cout << pi << endl;
getch(); // program bir tuşa basılmasını bekler
return 0;} // işletim sistemine aktarılacak deęer
```

Çizelge 3-3 Sistem başlangıç dosyaları.

stdio.h	Standart giriş çıkış fonksiyonlarını içerir (C)
iostream.h	Standart giriş çıkış fonksiyonları içerir (C++)
math.h	Matematiksel fonksiyonlar içerir (C/C++)
stdlib.h	Genel fonksiyonlar içerir (standard library)
time.h	Zaman fonksiyonlarını içerir