TEMEL GİRİŞ/ÇIKIŞ FONKSİYONLARI

Temel giriş/çıkış fonksiyonları, bütün programlama dillerinde mevcuttur. Bu tür fonksiyonlar, kullanıcıya ekrana veya yazıcıya bilgi yazdırmasına, ve bilgisayara klavyeden veri girişi yapmasına izin verir. Temel giriş/çıkış fonksiyonları kullanılırken stdio.h başlık dosyası programın başına eklenmelidir. Bu kısımda, en çok kullanılan giriş/çıkış fonksiyonları anlatılacaktır.

Printf() Fonksiyonu

Standart C kütüphanesinde bulunan printf() fonksiyonu, değişkenlerin tuttuğu değerleri, onların adreslerini veya bir mesajı ekrana belli bir düzenle (format) standart çıkışa (stdout), yani ekrana yazdırmak için kullanılan fonksiyondur. Daha önce yazılan örnek programlarda printf() fonksiyonundan yararlanılmıştır. Aşağıda bu fonksiyonun nasıl kullanıldığı gösterilecektir.

Örneğin basit olarak ekrana Hata oluştu!.. şeklinde bir mesaj yazdırma işlemi:

- printf("Hata Oluştu!..");

şeklindedir. Çoğu zaman ekrana, programda kullanılan bir değişkenin değeri yazdırılmak istenebilir. Örneğin ekrana bir tamsayı değişkeninin içeriğini bastırımak için,

- ....
- int x = 12;
- printf("x in değeri %d dir", x);
- ....

gibi kullanılır. Bu program parçasının ekran çıktısı şöyle olacaktır:

- x in değeri 12 dir

Bu örnekte printf fonksiyonuna iki parametre aktarılmıştır. Birincisi ekranda gösterilecek ve çift tırnaklar arasına yazılan ifadeler, ikincisi ise ekranda sayısal değeri gösterilmek istenen değişken(x) dir.

Genel olarak printf fonksiyonunun parametreleri üç kısımdan oluşmaktadır:

printf(“Düz metin %Format ve tip belirlecileri \Kontrol karakterleri”, değişken listesi)

I. **Düz metin (literal string):** yazdırılmak istenen ileti olup karakterlerden olşan herhangibir metindir.
   - Örneğin: printf("Ondokuz Mayis Universitesi..."); şeklinde yazdırılabilir.

II. **Konrol karakterleri (escape squence):** \ işareti ile başlar. Değişkenlerin ve sabitlerin nereye yazılacağını belirtmek (örneğin imlecin bir alt satıra geçirilmesi gibi işlemlerin gerçekleştirilmesi) için kullanılır. Bu karakterler Tablo 4.1.1'de listelenmiştir.

Tablo 0.1 Kontrol karakterleri

Karakter	**Anlamı**
\a	Ses üretir (alert)
\b	imleci bir sola kaydır (backspace)
\f	Sayfa atla. Bir sonraki sayfanın başına geç (formfeed)
\n	Bir alt satıra geç (newline)
\r	Satır başı yap (carriage return)
\t	Yatay TAB (horizontal TAB)
\v	Dikey TAB (vertical TAB)
\"	Çift tırnak karakterini ekrana yaz
\'	Tek tırnak karakterini ekrana yaz
\\	\ karakterini ekrana yaz
%%	% karakterini ekrana yaz

- Örneğin: printf("\tParagraf boslugu birak... \n");

III. **Tip belirleyici (conversion specifier):** % işareti ile başlar ve değişken alanını ve tipini belirleyen karakterleri içermektedir. Değişkenin yazdırılması veya okunması için kullanılır (%d gibi). Printf() fonksiyonunda yazma yapılırken, ileride anlatılacak scanf() fonksiyonunda ise okuma yapılacaktır. Ekrana yazdırılmak istenen değişkenin tipi, % işaretinden sonra belirtilir.

- Örneğin: printf("x in değeri %d dir");

  C programlama dilinde kullanılan bazı tip belirleyiciler ve açıklamaları Tablo 4.1.2'de verilmiştir.

Tip karakterlerini kullanarak, birden çok veri tipi yazdırılabilir. Örneğin;

- ...
- int not= 12;
- float pi = 3.14;
- char kr = 'A';
- printf(" not = %d , pi = %f ve kr = %c dir", not, pi, kr);

- ...

şeklinde yazılabilir.

printf() fonksiyonu esnektir. Parametreler herhangi bir C de deyimi olabilir. Örneğin x ve y nin toplamı şöyle yazılabilir:

- printf("%d", x+y);

Tablo 0.2 Tip belirleyiciler ve açıklamaları

Tip Karakteri	**Açıklama**	**Örnek**
%d or %i	İşaretli tam sayı	392
%U	İşaretsiz tam sayı	7235
%O	İşaretsiz oktal tam sayı	610
%x	İşaretsiz heksadesimal tam sayı	7fa
%X	İşaretsiz heksadesimal tam sayı (büyük harf)	7FA
%f	Gerçel sayı	392.65
%F	Gerçel sayı (büyük harf)	392.65
%e	Üstel sayı	3.9265e+2
%E	Üstel sayı (büyük harf)	3.9265E+2
%g	Gerçel ya da üstel sayı olarak sayıyı en kısa şekilde ifade eder.	392.65
%G	Gerçel ya da üstel sayı olarak sayıyı en kısa şekilde ifade eder (büyük harf).	392.65
%a	Heksadesimal gerçel sayı	-0xc.90fep-2
%A	Heksadesimal gerçel sayı (büyük harf)	-0XC.90FEP-2
%c	Karakter	a
%s	Karakter dizisi	sample

Değişken alanı w bir tamsayı olmak üzere değişkenin alanının kaç karakterden oluştuğunu gösterir. Gerçek değişken olması durumunda gösterim w.k şeklinde olup k noktadan (virgülden) sonra kaç hane olduğunu gösteren bir tamsayıdır.

Scanf() Fonksiyonu

Birçok programda ekrana verilerin bastırılmasının yanısıra klavyeden veri okunması gerekebilir. scanf() fonksiyonu klavyeden veri okumak için kullanılan fonksiyondur. scanf() fonksiyonu printf() gibidir ve Tablo 4.1.1 ve Tablo 4.1.2'de verilen karakterleri kullanır. Ancak parantez içinde düz metin bulunmaz. Genel kullanımı

scanf("%Format ve tip belirlecileri \Kontrol karakterleri", değişken listesi)

şeklindedir. Örneğin klaveden bir x tamsayısı okumak için:

- scanf("%d", &x);

satırını yazmak yeterli olacaktır. Burada & işareti adres operatörü olarak adlandırılır. Klavyeden iki farklı sayı okunmak istendiğnde scanf() fonksiyonu şöyle kullanılabilir:

- scanf("%d %f",&x,&y);

Bu durumda veriler klavyeden

- 16 1.56

ya da

- 16, 1.56

veya

- 16
- 1.56

şeklinde girilebilir. Burada fonksiyon aşağıdaki gibi yazılsaydı,

```
scanf("%d*%f",&x,&y);
```

bu durumda kullanıcı tarafından verilerin klavyeden 16*1.56 şeklinde girilmesi gerekecekti.

Puts() Fonksiyonu

Ekrana yazdırılacak ifade bir karakter dizisi ise, printf()'e alternatif olarak puts( ) fonksiyonu kullanılabilir. Ancak puts( ) , ekrana bu karakter dizisi yazdıktan sonra, imleci alt satıra geçirir. Örneğin;

- printf("Sevgi varlığın mayasıdır. \n");

ile

- puts("Sevgi varlığın mayasıdır.");

kullanımları eşdeğerdir. Puts() fonksiyonu Tablo 4.1.1 de verilen kontrol karakterleri ile de kullanılabilir.

- puts("Bu birinci satır... \nBu ikinci satır.");

çıktısı:

- Bu birinci satır...
- Bu ikinci satır.

şeklinde olacaktır.

Gets() Fonksiyonu

Klavyeden bir karakter dizisi okumak için kullanılır. Okuma işlemi yeni satır karakteriyle(\n) karşılaşılıncaya kadar sürer. puts() - gets() arasındaki ilişki, printf() - scanf() arasındaki gibidir. Yani,

- scanf("%s", str);

ile

- gets(str);

aynı anlamdadır.

Getchar() Fonksiyonu

Bu fonksiyon ile standart girişten bir karakter okunur. Programı istenen bir yerde durdurup, bir karakter girinceye kadar bekletir. Örneğin;

- for(i=0; i<10; i++)
- {
- getchar();
- printf("%d\n",i);
- }
- ...

Yukarıdaki program parçası 0-9 arası sayıları sırasıyla ekranda göstermek için kullanılır.Fakat her rakamı yazdırılmadan önce klavyeden herhangi bir karakter girip [Enter] tuşuna basılması beklenir.Bu bekleme getchar() fonksiyonu ile gerçekleştirilir.

Formatlı Çıktı

Bundan önceki programlardaki değişkenler serbest biçimde (free format), yani derleyicinin belirlediği biçimde ekrana yazdırılmıştı. Bazen giriş ve çıkışın biçimi kullanıcı tarafından belirlenmesi gerekebilir. Bu işlem;

- Tamsayılarda %d yerine %wd
- Gerçel sayılarda %f yerine %w.kf
- Stringlerde %s yerine %ws

biçimindeki kullanım ile sağlanır. Burada w yazılacak olan sayının veya alfasayının alan genişliği olarak adlandırılır. Gerçel bir değişken ekrana yazılacaksa, değişkenin virgülden (noktadan) sonra kaç basamağının yazdırılacağı k sayısı ile belirlenir. Ancak w > k + 2 olmalıdır. Bir sonraki bölümdeki örnekler ile formatlı çıktı konusu daha iyi anlaşılacaktır.