

C

Konsol Giriş Çıkış Fonksiyonları

Dr. Öğr. Üyesi M. Ozan AKI

Stringlerde Özel Karakterler

\n : Bir sonraki satırın ilk sütununa git

\b : Backspace (önceki karakteri sil)

\a : Bip sesi

**** : Ters bölü karakteri

\t : Tab karakteri

\v : Düşey tab (bir alt satıra geç)

\r : satır başına git

\' : Tek tırnak işareti

\" : Çift tırnak işareti

\f : Sayfa sonu karakteri

\ddd : Oktal notasyon ile ASCII karakter

\xdd : Hexadecimal notasyon ile ASCII karakter

Format Dizileri

%[Flag][Genişlik][.Uzantı][Öntip]Tip

Tip : d, u, o, x, X, f, e, c, s, p

Flag : + , - , #

Genişlik : Sayısal Değer

.Uzantı : .Sayısal Değer

Öntip : l, h, F, N

Tip Karakterleri

%d, %i : İşaretli Tamsayılar

%u : İşaretsiz Tamsayılar

%o : Oktal notasyonda tamsayılar

%x, %X : Hexadecimal notasyonda tamsayı

%f : Reel (Ondalık) Sayılar için

%e : Reel (Ondalık) Sayıların üstel yazımı

%c : Tek bir ASCII karakter yazdırılır

%s : String (char dizisi) yazdırılır

%p : Pointer adresi yazdırmak için kullanılır

Flag Karakterleri

- : Bilgi çıkışı her zaman sola yaslı olur

+ : Pozitif sayılarında (+) işaretinin,
Negatif sayılarda (-) işaretinin görülmemesini sağlar.

Boşluk : Pozitif sayılarda (+) işareti yerine boşluk yazdırılır

: %o ile kullanıldığında sayının önünde 0 yazar
%x ile kullanıldığında sayının önüne 0x yazar
%f, %e ile kullanıldığında ondalık nokta daima görünür

Genişlik

Ekrana yazdırılan bilgilerin belirli bir alan içerisinde yazdırılmasını sağlar. Ekrana yazılan bilgi, eğer genişlik kullanılmışsa varsayılan olarak sağa yaslı yazılır.

n : Kaç karakter alan olacağını belirtir

0n : Sayısal gösterimlerde sayının önünde kalan boşluğu sıfır ile doldurur

***** : Genişlik değerinin argüman listesinden alır.

Uzantı

Genişlik gibidir. Ancak ekrana yazılacak bilgiler alana sığmaz ise üzerinde kesme ya da yuvarlama yapar.

.0 : Ondalıklı sayılarda decimal nokta görüntülenmez, sayı yuvarlanarak sadece tam kısmı yazdırılır.

.n : Ondalıklı sayılarda kesirli kısmında yuvarlama yapılır, String ifadelerde Uzantı kadar string kesilerek yazdırılır.

.* : Uzantı değerini argüman listesinden alır.

Öntip

Değişkenlerin long, short gibi uzunluk bildirimlerini sağlar.

I : Tamsayılar için long, float için ise double bildirimi yapmayı sağlar.

F : Far tipinde pointer tanımlar

N : Pointer sadece offset adresini yazar

Örnekler

```
printf("%-5d",a);
```

```
printf("% 5x",a);
```

```
printf("%10f",a);
```

```
printf("%05u",a);
```

```
printf("%.0f",a);
```

```
printf("%.3f",a);
```

```
printf("%.10s",a);
```

```
printf("%c",a);
```

```
printf("%.2lf",a);
```

```
printf("%-.5d",a);
```

```
printf("%Np",a);
```

```
printf("%-5hd",a);
```

```
printf("%0d",a);
```

```
printf("%s",a);
```

Örnekler

```
printf("Toplam %d dir\n", toplam);
```

```
printf("%d x %d = %d", a, b, c);
```

```
printf("%d Hex: %x", sayi, sayi);
```

```
printf("Adres: %p", &sayi);
```

```
printf("\t%.2f\t%.2f", sayi1, sayi2);
```

```
printf("\nBir hata olustu\n\a");
```

scanf()

int scanf(char* format, argümanlar);

Klavyeden bilgi okumak amacıyla kullanılır. Atanan veri öğelerinin sayısını geri döndürür.

Format dizisinde üç sınıf sözkonusudur

- Biçim Belirteçleri
- Boşluk karakterleri
- Boşluk haricindeki karakterler

scanf() – Biçim karakterleri

printf fonksiyonunda kullanılan ve % ile belirtilen veri tipleri, scanf fonksiyonu için de geçerlidir. Okunmak istenen veri tipi format dizisinde % karakteri ile belirtilir. Örn.

scanf(«%d»,&i); // işaretli tamsayı okur

scanf(«%c»,&c); // sadece bir karakter

scanf(«%f»,&f); // ondalıklı sayı okur

scanf() – Biçim karakterleri

printf fonksiyonunda kullanılan ve % ile belirtilen veri tipleri, scanf fonksiyonu için de geçerlidir. Okunmak istenen veri tipi format dizisinde % karakteri ile belirtilir. Örn.

scanf(«%d»,&i); // işaretli tamsayı okur

scanf(«%c»,&c); // sadece bir karakter

scanf(«%f»,&f); // ondalıklı sayı okur

scanf() – Boşluk Karakteri

scanf format dizisinde kullanılan bir boşluk karakteri, girdi akışındaki ardışık tüm boşlukların gözardı edilmesini sağlar.

scanf, boşluk (space, tab, linefeed) olmayan ilk karaktere kadar girdinin okunmasını ancak işlem yapılamamasını sağlar.

scanf() – Boşluk Dışındaki Karakterler

scanf format dizisinde kullanılan bir karakteri, girdi akışından okur ve bu karakteri depolamadan okuma işlemine devam eder.

Eğer beklenen karakter girdiden okunamaz ise işlem sonlandırılır.

scanf() – Tarama Kümeleri

scanf format dizisinde sabit bir karakter yerine, okunması olası bir karakter kümesi tanımlanabilir.

Bu karakterler, % belirtecinden sonra [...] köşeli parantez içerisine tanımlanır.

Örneğin, tarih bilgisi okurken, ayıraç olarak /, -, . Gibi karakterlerin seçimlik olması, aşağıdaki gibi sağlanabilir;

```
scanf(«%u%[/-.] %u%[/-.] %u»,&gun,&ay,&yil);
```


scanf() – Girdiyi Gizlemek

Scanf girdi akışında, bir veri tipini okunmak isteniyor ancak herhangi değişkene atanmak istenmiyorsa * belirteci kullanılabilir. Örneğin,

```
scanf(«%d%*%d»,&x,&y);
```

Format dizi ile, iki tamsayı arasındaki bir karakter gözardı edilerek giriş okunmaktadır.

scanf() – Değişkene Değer Aktarımı

Format dizisinin yanına virgül ile ayrılmış ve format dizisindeki biçim niteleyicisi kadar sayıda değişken adresi verilmesi gerekir.

Karakter dizilerinden oluşan string ifadelerde, dizi adı tek başına adres gösterdiğinden ayrıca başına & operatörü konulmasına gerek yoktur.

Örnekler

```
scanf("%d", &sayi);
```

```
scanf("%d/%d/%d",&gun,&ay,&yil);
```

```
scanf("%s",ad);
```

```
scanf("%s %s", ad, soyad)
```

```
printf("Devam edecek misiniz?");
```

```
int i = scanf("%[ehEH]", &cevap);
```

```
if(i) ...
```

sprintf() ve sscanf()

sprintf(char *buf, char *format, ...)

printf fonksiyonundan tek farkı, konsol ekranı yerine çıktıları buf ile belirtilen diziye yazmasıdır.

sscanf(char *buf, char *format, ...)

scanf fonksiyonundan tek farkı, bilgileri klavyeden okumak yerine buf ile belirtilen diziden okumasıdır.

getchar() putchar()

int getchar();

Ekrandan (Klavyeden) Tek karakter okur. **Enter** tuşuna basılana kadar bekler, karakteri ekranda yazar ve bu karakterin kodunu döndürür.

putchar(int c);

c ile verilen ASCII kodundaki karakteri ekrana basar.

`getch() getch() <conio.h>`

`int getch();`

Ekrandan (Klavyeden) Tek karakter okur.

Herhangi bir tuşa basılana kadar bekler
ve bu karakterin kodunu döndürür.

Karakter ekranda görünmez.

`int getch();`

Tek farkı, girilen karakteri ekranda
göüntülemesidir.

gets() puts()

char* gets(char *str);

Ekrandan (Klavyeden) Karakter dizisi (string) okur. Enter tuşuna basılana kadar olan karakterleri string içerisine kopyalar

puts(char *str);

Bir karakter dizisini (string) ekranda görüntüler