Operatörler

Operatörler, programlama dillerinde matematiksel, mantıksal sınama ve karşılaştırma gibi işlemler için kullanılan bazı alfa nümerik karakterlerdir. Operatörlerin tam olarak ne iş yaptığı programı yazan kişi tarafından çok iyi bilinmelidir. Aksi halde çok küçük bir hata, çok büyük problemlere neden olabilir.

Aritmetiksel Operatörler

Değişkenler veya sabitler üzerinde temel aritmetik işlemler için kullanılırlar. Bu operatörler şunlardır:

Operatör	Açıklama	Örnek	Anlamı
+	toplama	x + y	x ve y nin toplamı
-	çıkarma	x - y	x ve y nin farkı
*	çarpma	x * y	x ve y nin çarpımı
/	bölme	x / y	x'in y'ye oranı
%	artık bölme	x % y	x'in y'ye bölümünden kalan
--	bir azalt	x--	x'in değerini bir azalt
++	bir artır	x++	X'in değerini bir artır

Aritmetiksel operatörler, tamsayı veya karakter değişkenleri için kullanılabilir. Ancak dikkat edilmelidir. Çünkü karakter değişkeni için örneğin ++ operatörü kullanıldığında, bu değişkenin içeriğindeki ASCII değeri 1 arttırılır. Bu da farklı bir karakter demektir. Ancak değişken tamsayı değişkeni ise bu değişken içindeki sayı değeri kaç ise ++ operatörü kullanıldığında, bu sayı 1 artar. Operatörün herhangi bir eşitlik ifadesi olmadan tek başına değişkenin sonunda veya başında kullanılmasının bir farkı yoktur.

Karşılaştırma Operatörleri

Bu operatörler değişken veya sabitleri birbirleriyle karşılaştırmada kullanılır. Yani sayısal veya karakter değişkenler birbiriyle karşılaştırılabilir ancak, karakter dizileri karşılaştırılamaz. Böylelikle değişik şartların yerine getirilmesi sağlanarak döngüsel veya

şartlı ifadelere bağlı programlarda kullanılabilir. C dilinde kullanılan karşılaştırma operatörleri ise şunlardır:

Operatör	Açıklama	Örnek	Anlamı
>	büyük mü?	x > y	x büyük mü y den?
<	küçük mü?	x < y	x küçük mü y den?
>=	büyük veya eşit mi?	x >= y	x y den büyük veya eşit mi?
<=	küçük veya eşit mi?	x <= y	x y den küçük veya eşit mi?
==	eşit mi?	x = y	x y ye eşit mi?
!=	farklı mı?	x != y	x y den farklı mı?

Karşılaştırma operatörleri az öncede değinildiği gibi şartlı ifadelerde ve döngülerde çok kullanılırlar. Karşılaştırma sonucunda doğru (true) yada yanlış (false) değeri gönderilir. Örneğin if 'li bir ifadede şart doğru ise blok içerisindeki işlemler yapılırken, şartın sonucu yanlış ise bu blok atlanır.

Mantıksal Operatörler

Bu operatörler, şartlı ifadeler ve döngülerde birden fazla sınanması gereken değişken veya sabit olması durumunda kullanılırlar. Bu operatörler şunlardır:

- ✓ && VE işlemi (AND)
- ✓ | | VEYA işlemi (OR)
- ✓ ! DEĞİL işlemi (NOT)

Atama Operatörleri

Bir değişkenin içerisine (kabul edilebilir) herhangi bir değeri eşitlemek için atama operatörleri kullanılır. Bu değer bir sabit değer olabileceği gibi, herhangi bir işlemin sonucu da olabilir.

Operatör	Açıklama	Örnek	Anlamı
=	atama	x = 7	X değişkenine 7 değeri atanmış
+=	ekleyerek atama	x += 3	x = x + 3
-=	eksilterek atama	x -= 5	x = x - 5
*=	çarparak atama	x *= 2	x = x * 2
/=	bölerek atama	x /= 4	x = x / 4
%=	bölüp kalanı atama	x %= 3	x = x % 3

sizeof Operatörü

Veri tiplerinin, değişkenlerin ve dizilerin bellekte kapladığı alan sizeof operatörü ile öğrenilebilir. Genel kullanım şekli;

sizeof(nesne) şeklindedir.

TEMEL GİRİŞ-ÇIKIŞ İŞLEMLERİ

Temel giriş/çıkış fonksiyonları, bütün programla 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.

printf() Fonksiyonu: Standart C kütüphanesine ait bir fonksiyondur. Bir mesajı veya değişkenlerle ilgili bir değeri ekrana yazdırmak için kullanılır.

Düz metin olarak kullanımı printf(“Merhaba Dünya”); şeklindedir.

Kontrol karakterleri ile kullanımı printf(“\tMerhaba Dünya…\n”); şeklindedir. Bazı kontrol karakterleri ve anlamları aşağıdaki tabloda verilmiştir.

Karakter	Anlamı
\a	Ses üretir (bip sesi)
\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
\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

Tip belirleyiciler yardımıyla kullanımı printf(“x sayısının değeri %d dir.”,x); şeklindedir.

Karakter	Yazdırılacak Veri Tipi
%c	char
%s	Karakter dizisi (string olmadığı için char)
%d	int
%f	float
%lf	double

printf() fonksiyonunun geri dönüş tipi integerdır. Bu özellik pek kullanılmasa da bilinmesinde fayda vardır. Kod bloğunun çıktısı ekrana yansıdığında sayi değişkeninin değerinin 13 olduğunu göreceksiniz.

```c
#include<stdio.h>

int main()
{
    int sayi;

    sayi=printf("merhaba dunya");

    printf("\n%d",sayi);

    return 0;
}
```

scanf()Fonksiyonu: Yazmış olduğumuz kodlarda çokça kullanacağımız bir fonksiyondur. Klavyeden girilen değerlerin program tarafından yakalanması için kullanılır. Yakalanan klavye değeri ilgili değişkene

atanırken kullanılır. Kısacası klavyeden veri okumak için kullanılan fonksiyondur. scanf() fonksiyonu da printf fonksiyonu gibi kontrol karakterlerini ve tip belirleyicileri kullanır. Örneğin klavyeden girilen x tamsayısını okumak için

scanf("%d",&x);

yazmak yeterli olacaktır. Burada & işareti adres operatörü olarak adlandırılır.

Klavyeden iki sayı arka arkaya girildiği zamanda sayıları okuma işlemi yandaki gibidir.

```c
#include<stdio.h>

int main()
{
    int x,y,z;

    printf("iki sayi giriniz...");
    scanf("%d %d",&x,&y);
    z=x+y;
    printf("\n%d",z);

    return 0;
}
```

puts() Fonksiyonu: Ekrana yazdırılacak ifade bir karakter topluluğu ise, printf()'e alternatif olarak puts() fonksiyonu kullanılabilir. Ancak puts(), ekrana bu karakter topluluğu yazdıktan sonra, imleci alt satıra geçirir. Yani:

printf("Merhaba Dünya.\n");

puts("Merhaba Dünya."); aynı şeylerdir.

Kontrol karakterleri puts fonksiyonu ile beraber kullanılabilir.

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

scanf("%s",str);

gets(str); aynı şeylerdir.

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);
}
```

Yandaki 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, 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 alan genişliği olarak adlandırılır. Gerçel bir değişken ekrana yazılacaksa, değişkenin virgülden sonra kaç basamağının yazdırılacağı ksayısı ile belirlenir. Ancak w > k + 2 olmalıdır.

```
#include<stdio.h>

int main()
{
    char c1='A', c2='c';
    int x=11, y=16;
    float f=56.7;
    double d=456.78912345;

    printf("+-------+-------+\n");
    printf("|%7c|%-7c|\n",c1,c2);
    printf("|%-7d|%7d|\n",x,y);
    printf("|%7.2f|%-7.1lf|\n",f,d);
    printf("+-------+-------+\n");

    return 0;
}
```

```
+-------+-------+
|      A|c      |
|11     |     16|
|  56.70|456.8  |
+-------+-------+
```

Yandaki kodu incelediğimizde ilk önce değişik tiplerde değişkenlerin tanımlandığını ve bunlara bir değer atandığını göreceğiz. Yazdırma işlemlerine başladığımızda ise beş adet printf fonksiyonu olduğunu görüyoruz. İlk ve son printf fonksiyonları standart ekran çıktısı vermektedir. İkinci printf fonksiyonunda ilk başta 7 karakterlik yer ayırıp içerisine sağa yaslı c1 karakterini sonra bir ayraç ve yine 7 karakterlik yer ayırdığımızı ve bununda sola yaslı c2 karakteri olduğunu görüyoruz. Üçüncü printf fonksiyonunda yine 7 karakterlik yer ayrılmış ve içerisine sola yaslı x değişkeni sonra sağa yaslı y değişkeni yerleştirilmiş. Dördüncü printf fonksiyonunda yine 7 karakterlik yer ayrılmış içerine float f değişkeni

ondalık kısmı 2 karakter olacak şekilde sağa yaslı yerleştirilmiş. Yanına ise sola yaslı double tipli d değişkeni ondalık kısmı bir karakter olacak şekilde yerleştirilmiştir.