**^^C PROGRAMLAMA DİLİ ^^**

Veri Tipleri, Değişkenleri;

Bilgisayarın çalışması basitçe bellekteki veriler arası aritmetik ve mantıksal işlemler yapıp sonuçları yine belleğe aktarmaktır. Bilgisayara bağlı tüm arabirimlerin ekran, klavye gibi bellek adresleri vardır bu adresleri bu arabirimlere erişmek için kullanır ve bizde bu sayede sonuçları ekranda görebiliriz. Çok karmaşık gibi görünen bu yapı aslında basit temeller üzerine kuruludur. Birçok basit bileşen birleşerek çok karmaşık bir yapı oluşturur.

Bilgisayar programları çalışırken kullandığı verileri bilgisayarda bulunan belleğe kaydederler, bu belleğe RAM adı verilir. Bellek birçok bellek hücresinin bir araya gelmesi ile oluşur. Her hücre 8 bitten oluşur. Bit ise 0 ya da 1 olabilen ikilik sayı sisteminde tek haneyi tutabilin elektronik bir yapıdır, 8 bite 1 bayt denir ve adreslenebilen en küçük bellek birimidir. Herhangi bir bellek hücresine erişebilmek için o hücreye ait adrese ihtiyacımız vardır. Belleği bir matris olarak düşünürsek ki gerçekte de böyledir bu adres de belleğin fiziksel olarak satır numarası ve sütun numarasından oluşur. Bu iki sayının bir araya gelmesi ile her hücrenin kendine ait eşsiz bir adresi oluşur. Bilgisayar işlemcisi bu adresi kullanarak bu adresteki veriye ulaşır. C'de tanımladığımız her değişkenin eşsiz bir bellek adresi vardır. İstersek bu adresi bir fonksiyon ile görebiliriz.

Değişken kavramı tüm programlama dillerinde kullanılır. İçine veri kaydedilebilen bellek hücresi ya da hücre topluluğu anlamına gelir. Program içinde basitce bir isim verilerek tanımlanır ve o isimle veriye erişilir ya da yeni veri kaydedilir. C dilindeki temel veri tipleri aşağıda sıralanmıştır. Veri tiplerinin uzunlukları çalışma ortamına göre değişir aşağıdaki veri tipleri  32 bitlik bir sisteme göre verilmiştir.

#### **C Programlama Dilinde Veri Tipleri**

|  |  |  |
| --- | --- | --- |
| **Veri Tipi** | **Uzunluk** | **Limit** |
| Char | 1 bayt | -127 , +127 |
| unsigned char | 1 bayt | 0 , 255 |
| short | 2 bayt | −32767 , +32767 |
| unsigned short | 2 bayt | 0 , 65535 |
| int | 4 bayt | −2147483647 , +2147483647 |
| unsigned int | 4 bayt | 0 , 4294967295 |
| long | 4 bayt | −2147483647 , +2147483647 |
| unsigned long | 4 bayt | 0 , 4294967295 |
| float | 4 bayt | 1.2E-38 , 3.4E+38 |
| double | 8 bayt | 2.3E-308 , 1.7E+308 |
| long double | 10 bayt | 3.4E-4932 , 1.1E+4932 |

Yukarıda yazılı olan bütün veri tipleri aynı zamanda c dilinde birer anahtar kelimedir. C'de değişken tanımlanırken önce veri tipi daha sonra değişken ismi yazılır. Unsigned ön eki ise değişkenin sadece pozitif sayılar için kullanılabileceğini ifade eder. Yazılmadığı taktirde değişkenin hem pozitif hemde negatif sayılarda kullanılabileceği anlamına gelir.

### **Değişken Tanımlama Kuralları**

* Değişken isimleri harf ve rakam dışında yalnızca alt tire içerebilir.
* Değişken isminde rakam kullanılabilir ama değişken ismi rakamla başlayamaz.
* Değişken isimleri küçük büyük harfe duyarlıdır.
* Aynı kapsam içinde aynı isimle yalnızca bir değişken tanımlanabilir.

|  |  |
| --- | --- |
|  | İnt sayac; *// sayaç değişkeni* |

Yukarıdaki satırda sayac isminde ve int türünde bir değişken tanımladık bunun anlamı sayac değişkenine −2147483647 , +2147483647 aralığında bir sayı kaydedebiliriz. C dilinde if,for,while dışında her ifadeden sonra noktalı virgül konur. Daha sonraki çift slaş ise açıklama satırı anlamına gelir ve bu satıra yazdıklarınız derlemede dikkate alınmaz, bilgilendirme amaçlıdır. Çok satırlı açıklamalar ise /\* ile başlar ve \*/ ile biter bunlar arasına yazılan herşey açıklama olarak değerlendirilir.

Şimdi oluşturduğumuz değişkene bir sayı kaydedelim.

|  |  |
| --- | --- |
|  | Sayac = 95 ; |

Yukardaki tüm veri tiplerinde değişkenleri bu şekilde tanımlayabiliriz. Aşağıda değişkenlerle çarpma işlemi yapılıyor.

|  |  |
| --- | --- |
|  | İnt sonuc = 0 ; //sonuç değerini tutacak değişken  İnt a = 7; //a değişkeni  İnt b = 9 ; //b değişkeni  sonuc= a\*b ; //değişkenler arası çarpma işlemi sonuç değişkeninin değeri  7\*9=63 tür. |

sonuc=a=b;

Bunun gibi bir ifade kullanırsak işlem sırasına göre b'nin içeriği a'ya aktarılır a'nın içeriğide sonuc değişkenine aktarılır. Sonucta bütün değişkenlere b'nin içeriği olan 9 kaydedilir.

C dilinin C ile özdeşleşmiş standart bir kitaplığı vardır. Bu kitaplık içindeki birçok fonksiyon diğer programlama dillerine de geçmiştir. En sık kullanılan fonksiyon ekrana bir şey yazmak için kullanılan "printf" fonksiyonudur. Standart kitaplık  C'nin ayrılmaz bir parçasıdır. İlerleyen bölümlerde kitaplık fonksiyonlarını sıkça kullanacağız.

## **Veri türleri;**

C'de aşağıda gösterilen 5 temel veri çeşidi vardır:

|  |  |
| --- | --- |
| **C TEMEL VERİ TÜRLERİ** | |
| ANAHTAR KELİME | VERİ TÜRÜ |
| Char | Karakter |
| İnt | Tamsayı |
| Float | Kayan noktalı sayı |
| Double | Çift duyarlıklı kayan noktalı sayı |
| Void | Değersiz |

## **Veri türü değiştiricileri (modifiers);**

Sadece char, int ve double veri türlerinin önüne aşağıda gösterilen değiştiricileri koyarak, veri tiplerinin işaret (+/-) durumunu, byte olarak boyutlarını ve sınırlarını değiştirebiliriz. C’da kullanılan veri türü değiştiricileri ve uygulandıkları veri türleri aşağıda gösterilmektedir:

|  |  |
| --- | --- |
| **C VERİ TÜRÜ DEĞİŞTİRİCİLERİ** | |
| DEĞİŞTİRİCİ | UYGULANDIĞI VERİ TÜRÜ |
| Signed | Char , int |
| Unsigned | Char , int |
| Long | İnt , double |
| Short | İnt |

Değişken tanımlamalarında signed değiştirici ifadesi kullanıldığında, tanımlanan değişkenin ikili sayı sistem değeri içinde en üst sırada yer alan bit'i işaret bit'i olarak kullanılır. Bu değer 0 olursa sayı pozitif, 1 olursa sayı negatif olur. Ancak, bu durumda bir bit sayının işaretini belirlemek için ayrıldığından tanımlanan değişkene atanacak değer otomatik olarak azalır.

|  |
| --- |
| ! Bilgisayarlarda, bir sayının ikili sayı sisteminde gösteriminde en solda yer alan bit, sayının negatif veya pozitif olduğunu göstermek üzere, işaret bit'i olarak kullanılır. |

C dilinde temel veri çeşitleri kullanılarak oluşturulabilen bütün veri çeşitleri ile byte uzunlukları ve sınırlarının genel olarak kullanılan değerleri ile printf() ve scanf() fonksiyonları ile kullanılan format tanımlayıcıları aşağıdaki tabloda gösterilmektedir:

C'nin diğer önemli bir özelliği de, char değişkenlerin ASCII karakter seti dışında değerler taşıyabilmesidir. Eğer, -128 ile 127 arasında kalan küçük int değerlerle işlem yapıyorsak, bu int değerleri atamak için int değişken yerine char bir değişken kullanabiliriz.

## **C veri türleri tablosu;**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| C VERİ TÜRLERİ TABLOSU | | | | | |
| VERİ TÜRÜ | BİT GENİŞLİĞİ | ALT SINIR | ÜST SINIR | FORMAT TANIMLAYICISI | |
| Printf() | Scanf() |
| Char | 8 | -128 | 127 | %c | %c |
| Unsigned char | 8 | 0 | 255 | %c | %c |
| Signed char | 8 | -128 | 127 | %c | %c |
| İnt | 16 veya 32 | -32.768 veya -2.147.483.648 | 32.768 veya 2.147.483.648 | %d | %d |
| Unsigned char | 16 veya 32 | 0 | 65.535 veya 4.294.967.295 | %u | %u |
| Short int | 16 | -32.768 | 32.767 | %hd | %hd |
| Unsigned short int | 16 | 0 | 65.535 | %hu | %hu |
| Long int | 32 | -2.147.483.648 | 2.147.483.647 | %ld | %ld |
| Long long int (C99) | 64 | -9.223.372.036.854.775.808 | 9.223.372.036.854.775.807 | %lld | %lld |
| Unsigned long int | 32 | 0 | 4.294.967.295 | %lu | %lu |
| Unsigned long long int (C99) | 64 | 0 | 18.446.744.073.709.551.615 | %llu | %llu |
| Float | 32 | 1.17549e-083 | 3.40282e+038 | %f | %f |
| Double | 64 | 2.22507e-308 | 1.79769e+308 | %f | %lf |
| Long double | 96 | 3.3621e-4932 | 1.18973e+4932 | %Lf | %Lf |

## **Veri çeşitlerini geçici olarak değiştirme (Type cast);**

Bazı durumlarda ihtiyaç duyarsanız bir değişkenin veri çeşidini geçici olarak değiştirebiliriz. Bunun için aşağıda genel yapısı verilen type cast özelliğini kullanabiliriz:

|  |
| --- |
| (veri-türü ) değişken ; |

## **İfadelerde veri çeşidi değişimi;**

C'de, farklı veri çeşitlerini aynı ifade içinde kullanabiliriz. Çünkü, C'de aynı ifade içinde farklı veri çeşitlerini düzenli bir şekilde kullanarak sonuçta tek bir veri çeşidi elde eden bir sistem vardır.

C'de, bir ifade içinde bir char veya short int veri kullanıldığında, bu verilerin değerleri içinde bulundukları ifadenin çalışması esnasında int değere çevrilir. int değişkenler yerine char değişkenlerin kullanılmasının nedeni budur. Bu değişim otomatik olarak yapılmaktadır.

Otomatik veri çeşidi değişimleri yapıldıktan sonra, program ifadelerde yer alan bütün verileri aynı ifade içindeki en büyük verinin çeşidine çevirir. Bu işlem aşağıda belirtilen kurallara göre yapılır:

|  |  |  |
| --- | --- | --- |
| İlk veri çeşidi | Diğer verilerin çeşidi | Elde edilen veri çeşidi |
| Long double  Double  Float  Unsigned long  Long  Unsigned int | …………….  ……………..  ………………  ………………..  ……………….  ………………… | Long double  Double  Float  Unsigned long  Long  Unsigned int |

# float ve double arasındaki fark nedir?

“float” boyutu 4 bayttır. “double” boyutu ise 8 bayttır. “float” 6 basamaklı değerleri desteklerken, “double” 14 basamaklı değerleri desteklemektedir. Yani projenizde “float” veya “double” değeri kullanırken değişkenin alabileceği değerin basamak sayısını göz önünde tutabilirsiniz. Yanlış kullanım sonucunda kod bloklarınız yanlış değerler ile çalışabilir.

# Değişkenler

**Değişkenler** bilgisayarın geçici belleğinde bilginin saklandığı gözlere verilen sembolik adlardır. Bir C programında, bir değişken tanımlandığında bu değişken için bellekte bir yer ayrılır. Her değişkenin tuttuğu değerin nasıl bir veri olduğunu gösteren bir veri tipi vardır.

* **Değişkenler tanımlanırken bazı kurallara uymak zorunludur. Bunlar;**

Değişken adları en fazla 32 karakterden oluşabilir. 32 karakterden uzun değişken adları ilk  
32 karakteri değerlendirilir. Geriye kalan karakterler işleme tabi tutulmaz.

Değişken adları ingiliz alfabesinde bulunan karakterler (A-Z) veya (a-z) yada rakamlar (0–9) ile yazılmalıdır. . **Türkçe karakterler**, **özel karakter** veya **boşluk**  ,**karakteri** kullanılamaz.

**Değişken adları herhangi bir rakam ile başlayamaz.** Ilk karakter bir harf olmalıdır.  
Sonrakiler rakamlardan oluşabilir.

**Temel Veri Tipleri;**

C dilinde üç tane temel veri tipi vardır;

1. Sayı
2. Karakter
3. String

Bunlar uygulamada alt gruplarına ayrılırlar. Alt grupların oluşturulmasında ortak bir standarttan sözedilemez. Farklı derleyiciler farklı sınıflandırmalar ve farklı adlar kullanmıştır. Onları alışılmış adlarına ve bellekte kapladıkları büyüklüklere göre sınıflandırmak yeterli olacaktır.

**İlkel Veri Tipleri;**

|  |  |
| --- | --- |
| Tamsayılar  Karakterler  Mantıksal  Kesirli sayılar  Yapısal tipler | İnt  Char  False == 0 , True != 0  Float , double  Array, string, functions, struct, files, pointers |

İlkel veri tipleri deyince, bir dilde çok kullanıldığı için dile gömülü gelen, dolayısıyla kütüphneden çağrılması gerekmeyen veri tipleri akla gelir.