**Veri türleri**

C'de aşağıda gösterilen 5 temel veri çeşidi vardır:

|  |  |
| --- | --- |
| **C temel veri türleri** | |
| **Anahtar kelime** | **Veri türü** |
| **char** | Karakter |
| **int** | 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 veri türü değiştiricileri** | |
| **Değiştirici** | **Uygulandığı veri türü** |
| **signed** | char, int |
| **unsigned** | char, int |
| **long** | int, double |
| **short** | int |

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.

Pozitif bir sayıyı negatif bir sayıya çevirmek için, sayıdaki bütün bit değerlerinin tersi alınır ve elde edilen sayıya 1 değeri eklenir. Biri signed diğeri unsigned olan iki adet char ve iki adet short int veri türünden değişken tanımlayalım:

signed char cd; // Alabileceği maksimum değer : 127 => 0 1 1 1 1 1 1 1

unsigned char ucd; // Alabileceği maksimum değer : 255 => 1 1 1 1 1 1 1 1

short int sid; // Alabileceği maksimum değer : 32767 => 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1

unsigned short int usid; // Alabileceği maksimum değer : 65535 => 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

cd ve sid değişkenleri için el solda yer alan değer işaret bit'i olarak kullanıldığından, değişkenlerin alabileceği maksimum değer sırasıyla 127 ve 32767 olacaktır. ucd ve usid değişkenleri en solda yer alan bit'i sayı değerleri için kullanabileceğinden, değişkenlerin alabileceği maksimum değer sırasıyla 255 ve 65535 olacaktır.

C'de, unsigned, short veya long int değerlerin bildirimi yapılırken int ifadesinin kullanılması tamamen isteğe bağlıdır. Başka bir ifade ile, bu tip tanımlamalarda int ifadesini kullanmayabiliriz. Örneğin, aşağıdaki işlem satırlarının her ikisi de id adlı unsigned int bir değişken tanımlar:

unsigned int id;

unsigned id;

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 veri türleri tablosu** | | | | | |
| **Veri türü** | **Bit genişliği** | **Alt sınır** | **Üst sınır** | **Format tanımlayıcısı** | |
| printf() | scanf() |
| char | 8 | -128 | 127 | %c | %c |
| unsigned char | 8 | 0 | 255 | %c | %c |
| signed char | 8 | -128 | 127 | %c | %c |
| int | 16 veya 32 | -32.768 veya -2.147.483.648 | 32.767 veya 2.147.483.647 | %d | %d |
| unsigned int | 16 veya 32 | 0 | 65.535 veya 4.294.967.295 | %u | %u |
| short int | 16 | -32.768 | 32767 | %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-038 | 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 |

Aşağıdaki program çalıştırıldığı bilgisayardaki tanımlı veri türlerinin byte ve bit olarak genişlikleri ile **limits.h** ve **float.h** başlık dosyalarında tanımlı alt ve üst sınır değerlerini ekrana yazar.

Örnek

#include <stdio.h>

#include <limits.h>

#include <float.h>

int main(void)

{

printf("char -> byte: %d bit: %d min: %d max: %d\n", sizeof(char), sizeof(char)\*8, CHAR\_MIN, CHAR\_MAX);

printf("unsigned char -> byte: %d bit: %d min: %d max: %d\n", sizeof(unsigned char), sizeof(unsigned char)\*8, (unsigned char) 0, UCHAR\_MAX);

printf("signed char -> byte: %d bit: %d min: %d max: %d\n\n", sizeof(signed char), sizeof(signed char)\*8, SCHAR\_MIN, SCHAR\_MAX);

printf("int -> byte: %d bit: %d min: %d max: %d\n", sizeof(int), sizeof(int)\*8, INT\_MIN, INT\_MAX);

printf("unsigned int -> byte: %d bit: %d min: %u max: %u\n", sizeof(unsigned int), sizeof(unsigned int)\*8, (unsigned int) 0, UINT\_MAX);

printf("short int -> byte: %d bit: %d min: %hd max: %hd\n", sizeof(short int), sizeof(short int)\*8, SHRT\_MIN, SHRT\_MAX);

printf("unsigned short int -> byte: %d bit: %d min: %hu max: %hu\n\n", sizeof(unsigned short int), sizeof(unsigned short int)\*8, 0, USHRT\_MAX);

printf("long int -> byte: %d bit: %d min: %ld max: %ld\n", sizeof(long int), sizeof(long int)\*8, LONG\_MIN, LONG\_MAX);

printf("long long int -> byte: %d bit: %d min: %lld max: %lld\n", sizeof(long long int), sizeof(long long int)\*8, LLONG\_MIN, LLONG\_MAX);

printf("unsigned long int -> byte: %d bit: %d min: %lu max: %lu\n", sizeof(unsigned long int), sizeof(unsigned long int)\*8, (unsigned long int) 0, ULONG\_MAX);

printf("unsigned long long int -> byte: %d bit: %d min: %llu max: %llu\n\n", sizeof(unsigned long long int), sizeof(unsigned long long int)\*8, (unsigned long long int) 0, ULLONG\_MAX);

printf("float -> byte: %d bit: %d min: %g max: %g\n", sizeof(float), sizeof(float)\*8, -FLT\_MAX, FLT\_MAX);

printf("double -> byte: %d bit: %d min: %lg max: %lg\n", sizeof(double), sizeof(double)\*8, -DBL\_MAX, DBL\_MAX);

printf("long double -> byte: %d bit: %d min: %Lg max: %Lg\n", sizeof(long double), sizeof(long double)\*8, -LDBL\_MAX, LDBL\_MAX);

return 0;

}