

Tipos de enteros en C++

Tipo de dato	Nombre abreviado	Número usual de bits	Rango usual de valores	Especificador de formato (lectura numérica)
bool	bool	8	[0, 1]	
char	char	8	[-128, 127] o [0, 255]	
signed char	signed char	8	[-128, 127]	%hhd
unsigned char	unsigned char	8	[0, 255]	%hhu
signed short int	short	16	[-32768, 32767]	%hd
unsigned short int	unsigned short	16	[0, 65535]	%hu
signed int	int	32	[-2147483648, 2147483647]	%d
unsigned int	unsigned	32	[0, 4294967295]	%u
signed long int	long	32	[-2147483648, 2147483647]	%ld
unsigned long int	unsigned long	32	[0, 4294967295]	%lu
signed long long int	long long	64	[-9223372036854775808, 9223372036854775807]	%lld
unsigned long long int	unsigned long long	64	[0, 18446744073709551615]	%llu

En C/C++, **sizeof(T)** es la cantidad de **chars** que ocupan la misma memoria que una variable de tipo **T**. Por definición, **sizeof(char)** es 1. Como en la mayoría de los computadores un **char** tiene 8 bits, **sizeof(T)** es la cantidad de bytes que ocupa una variable de tipo **T**.

Todos los tipos enteros son **signed** por omisión (excepto **char**, que depende de la plataforma: en algunas actúa como **signed** y en otras como **unsigned**; si no investigan qué ocurre en su plataforma, sólo pueden usar el rango [0, 127] de un **char** de manera segura). En Linux de 64 bits, **signed long int** y **unsigned long int** tienen 64 bits. El tipo **bool** de C++ se llama **_Bool** en C, pero incluyendo **<stdbool.h>** puede usarse con el nombre **bool**.

Tipos de flotantes en C++

Tipo de dato	Número de bits para la mantisa	Número de bits para el exponente	Rango usual de valores	Especificador de formato
float	24	8	[-3.40282e+038, 3.40282e+038]	%f
double	53	11	[-1.79769e+308, 1.79769e+308]	%lf
long double	64 o 113	15	[-1.18973e+4932, 1.18973e+4932]	%Lf