Summary of C Programming Basic Data Types

Integral Types - See limits.h for implementation-specific details and certain defined constants

Type	aka	Typical bytes	Typical min	Typical max	printf	scanf	Notes
int		4	-2,147,483,648	2,147,483,647	**%d** %i %o %x %X	**%d** %i %o %x %X	Most common integer type. Matches machine "word" size.
short int	short	2	-32768	32767	**%hd** %hi %ho %hx %hX	**%hd** %hi %ho %hx %hX	Size ≤ size of int
long int	long	4	-2,147,483,648	2,147,483,647	**%ld** %li %lo %lx %lX	**%ld** %li %lo %lx %lX	Size ≥ size of int
long long int	long long	8	~ -9.223×10^{18}	~ 9.223×10^{18}	**%lld** %lli %llo %llx %llX	**%lld** %lli %llo %llx %llX	Must be at least 64 bits
unsigned int	unsigned	4	0	4,294,967,295	**%ud** %ui %uo %ux %uX	**%ud** %ui %uo %ux %uX	
unsigned short		2	0	65,535	**%uhd** %uhi %uho %uhx %uhX	**%uhd** %uhi %uho %uhx %uhX	
unsigned long		4	0	4,294,967,295	**%uld** %uli %ulo %ulx %ulX	**%uld** %uli %ulo %ulx %ulX	
unsigned long long		8	0	~ 1.845×10^{19}	**%ulld** %ulli %ullo %ullx %ullX	**%ulld** %ulli %ullo %ullx %ullX	
char		1	-128	127	**%d** %i %o %x %X	**%d** %i %o %x %X	Normally characters (below)
unsigned char		1	0	255	**%ud** %ui %uo %ux %uX	**%ud** %ui %uo %ux %uX	

Integer Constant Formats - normally signed ints unless a trailing L or U indicates long int and/or unsigned respectively

Decimal	[+-]1-9[0-9...][lIUu]	Optional sign, followed by a digit from 1 to 9, followed by optional digits from 0 to 9, optionally followed by L or U
Octal	[+-]0[0-7...][lIUu]	Optional sign, followed by a leading 0, followed by optional digits from 0 to 7, optionally followed by L or U
Hexadecimal	[+-]0x[0-9a-fA-F...][lIUu]	Optional sign, followed by 0x, followed by optional digits 0 to F in upper or lower case, optionally followed by L or U

Floating Point Types - See float.h for implementation-specific details and certain defined constants.

Type	Typical bytes	Smallest positive	Largest Value	printf	scanf	Precision
float	4	1.17549×10^{-38}	3.40282×10^{38}	%f %e %E %g %G	**%f** %e %E %g %G	~ 6 decimal digits
double	8	2.22507×10^{-308}	1.79769×10^{308}	%f %e %E %g %G	**%lf** %le %lE %lg %lG	~ 15 decimal digits
long double	80 or 128 bits			%Lf %Le %LE %Lg %LG	**%Lf** %Le %LE %Lg %LG	

Floating Point Constant Formats -Normally doubles unless a trailing F or L is applied to indicate float or long double types respectively.

[+-]1-9[0-9…].[0-9…][[Ee[+-]0-9…][FfLl]	The presence of a decimal point normally indicates a double precision number.
[+-][0].[0-9…][[Ee[+-]0-9…][FfLl]	If the leading digit is a 0 or absent, no additional digits may preceed the decimal point.
[+-]1-9[0-9…]Ff[Ll]	If no decimal point is present, then either the F or E notation is required to indicate a floating point data type.
[+-]1-9[0-9…]Ee[+-]0-9…[FfLl]	Exponential notation indicates multiplication by a given power of 10. e.g. 6.02E23 = 6.02×10^{23}

Character Type

The **char** type occupies 1 bytes, uses % for both printf and scanf, and prints the ASCII character corresponding to its numerical value
Constant chars are enclosed in single quotes('A'), and may include escape sequences such as '\n', '\t', etc.

Character String

Arrays of characters use the %s format specifier for both printf and scanf. Scanning stops on the first white-space character read in.
Constant character strings are enclosed in double quotes, and may also include escape characters, e.g. "\n\nPlease enter X > '

Special Types: enum, struct, _Bool, _Complex, _Imaginary

Related Functions and Concepts: sizeof( ), typedef, type casting