Table 27.1 <stdint.h> Limit Macros for Specified-Width Integer Types

Name	*Value*	*Description*
INTN_MIN	$-(2^{N-1})$	Minimum intN_t value
INTN_MAX	$2^{N-1}-1$	Maximum intN_t value
UINTN_MAX	$2^{N}-1$	Maximum uintN_t value
INT_LEASTN_MIN	$\leq-(2^{N-1}-1)$	Minimum int_leastN_t value
INT_LEASTN_MAX	$\geq2^{N-1}-1$	Maximum int_leastN_t value
UINT_LEASTN_MAX	$\geq2^{N}-1$	Maximum uint_leastN_t value
INT_FASTN_MIN	$\leq-(2^{N-1}-1)$	Minimum int_fastN_t value
INT_FASTN_MAX	$\geq2^{N-1}-1$	Maximum int_fastN_t value
UINT_FASTN_MAX	$\geq2^{N}-1$	Maximum uint_fastN_t value
INTPTR_MIN	$\leq-(2^{15}-1)$	Minimum intptr_t value
INTPTR_MAX	$\geq2^{15}-1$	Maximum intptr_t value
UINTPTR_MAX	$\geq2^{16}-1$	Maximum uintptr_t value
INTMAX_MIN	$\leq-(2^{63}-1)$	Minimum intmax_t value
INTMAX_MAX	$\geq2^{63}-1$	Maximum intmax_t value
UINTMAX_MAX	$\geq2^{64}-1$	Maximum uintmax_t value

Limits of Other Integer Types

<stddef.h> header ➤21.4
<signal.h> header ➤24.3
<wchar.h> header ➤25.5

When the C99 committee created the <stdint.h> header, they decided that it would be a good place to put macros describing the limits of integer types besides the ones declared in <stdint.h> itself. These types are ptrdiff_t, size_t, and wchar_t (which belong to <stddef.h>), sig_atomic_t (declared in <signal.h>), and wint_t (declared in <wchar.h>). Table 27.2 lists these macros and shows the value of each (or any constraints on the value imposed by the C99 standard). In some cases, the constraints on the minimum and maximum values of a type depend on whether the type is signed or unsigned. The macros in Table 27.2, like the ones in Table 27.1, represent constant expressions.

Macros for Integer Constants

integer constants ➤7.1

The <stdint.h> header also provides function-like macros that are able to convert an integer constant (expressed in decimal, octal, or hexadecimal, but without a U and/or L suffix) into a constant expression belonging to a minimum-width integer type or greatest-width integer type.

Integer promotions ➤7.4

For each int_leastN_t type declared in <stdint.h>, the header defines a parameterized macro named INTN_C that converts an integer constant to this type (possibly using the integer promotions). For each uint_leastN_t type, there's a similar parameterized macro named UINTN_C. These macros are useful for initializing variables, among other things. For example, if i is a variable of type int_least32_t, writing