

The range of values represented by each of the six integer types varies from one machine to another. However, there are a couple of rules that all compilers must obey. First, the C standard requires that `short int`, `int`, and `long int` each cover a certain minimum range of values (see Section 23.2 for details). Second, the standard requires that `int` not be shorter than `short int`, and `long int` not be shorter than `int`. However, it's possible that `short int` represents the same range of values as `int`; also, `int` may have the same range as `long int`.

Table 7.1 shows the usual range of values for the integer types on a 16-bit machine; note that `short int` and `int` have identical ranges.

**Table 7.1**  
Integer Types on a  
16-bit Machine

<i>Type</i>	<i>Smallest Value</i>	<i>Largest Value</i>
<code>short int</code>	-32,768	32,767
<code>unsigned short int</code>	0	65,535
<code>int</code>	-32,768	32,767
<code>unsigned int</code>	0	65,535
<code>long int</code>	-2,147,483,648	2,147,483,647
<code>unsigned long int</code>	0	4,294,967,295

Table 7.2 shows the usual ranges on a 32-bit machine; here `int` and `long int` have identical ranges.

**Table 7.2**  
Integer Types on a  
32-bit Machine

<i>Type</i>	<i>Smallest Value</i>	<i>Largest Value</i>
<code>short int</code>	-32,768	32,767
<code>unsigned short int</code>	0	65,535
<code>int</code>	-2,147,483,648	2,147,483,647
<code>unsigned int</code>	0	4,294,967,295
<code>long int</code>	-2,147,483,648	2,147,483,647
<code>unsigned long int</code>	0	4,294,967,295

In recent years, 64-bit CPUs have become more common. Table 7.3 shows typical ranges for the integer types on a 64-bit machine (especially under UNIX).

**Table 7.3**  
Integer Types on a  
64-bit Machine

<i>Type</i>	<i>Smallest Value</i>	<i>Largest Value</i>
<code>short int</code>	-32,768	32,767
<code>unsigned short int</code>	0	65,535
<code>int</code>	-2,147,483,648	2,147,483,647
<code>unsigned int</code>	0	4,294,967,295
<code>long int</code>	-9,223,372,036,854,775,808	9,223,372,036,854,775,807
<code>unsigned long int</code>	0	18,446,744,073,709,551,615

Once more, let me emphasize that the ranges shown in Tables 7.1, 7.2, and 7.3 aren't mandated by the C standard and may vary from one compiler to another. One way to determine the ranges of the integer types for a particular implementation is to check the `<limits.h>` header, which is part of the standard library. This header defines macros that represent the smallest and largest values of each integer type.