

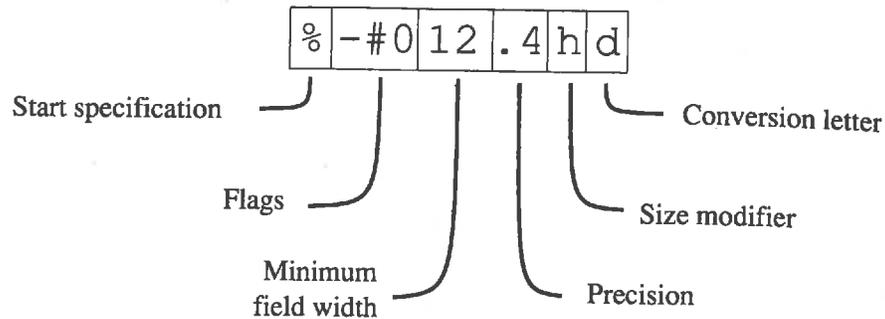
Table 15-2: Output conversion specifications

| Conversion | Defined flags - + # 0 space | Size modifier | Argument type | Def. prec ¹ | Output |
|-------------------|--------------------------------|------------------|---|---------------------------|---|
| d, i ² | - + 0 space | none h l | int short long | 1 | dd...d -dd...d +dd...d |
| u | - + 0 space | none h l | unsigned int unsigned short unsigned long | 1 | dd...d |
| o | - + # 0 space | none h l | unsigned int unsigned short unsigned long | 1 | oo...o 0oo...o |
| x, X | - + # 0 space | none h l | unsigned int unsigned short unsigned long | 1 | hh...h 0xhh...h 0Xhh...h |
| f | - + # 0 space | none L | double long double | 6 | d...d.d...d -d...d.d...d +d...d.d...d |
| e, E | - + # 0 space | none L | double long double | 6 | d.d...de+dd -d.d...dE-dd |
| g, G | - + # 0 space | none L | double long double | 6 | like e, E, or f |
| c | - | none | int | 1 | c |
| s | - | none | char * | ∞ | cc...c |
| p ² | <i>impl. defined</i> | none | void * | 1 | <i>impl. defined</i> |
| n ² | | none h l | int * short * long * | n/a | none |
| % | | none | none | n/a | % |

¹ Default precision, if none is specified.

² Available in ANSI C; may be rare elsewhere. i and d are equivalent on output.

The prefix is computed as follows. If the argument is negative, the prefix is a minus sign. If the argument is nonnegative and the + flag is specified, then the prefix is a plus sign. If the argument is nonnegative, the space flag is specified, and the + flag is not



15.11.3 Conversion Flags

The optional flag characters modify the meaning of the main conversion operation:

- Left-justify the value within the field width .
- 0 Use 0 for the pad character rather than space.
- + Always produce a sign, either + or -.
- space* Always produce either the sign - or a space.
- # Use a variant of the main conversion operation.

The effects of the flag characters are described in more detail below.

The - flag If a minus-sign flag is present, then the converted value will be left-justified within the field; that is, any padding will be placed to the right of the converted value. If no minus sign is present, then the converted value will be right-justified within the field. This flag is relevant only when an explicit minimum field width is specified and the converted value is smaller than that minimum width; otherwise, the value will fill the field without padding.

The 0 flag If a 0 (zero) flag is present, then 0 will be used as the pad character if padding is to be placed to the left of the converted value. The 0 flag is relevant only when an explicit minimum field width is specified and the converted value is smaller than that minimum width. In integer conversions, this flag is superseded by the precision specification.

If no zero-digit flag is present, then a space will be used as the pad character. Space is always used as the pad character if padding is to be placed to the right of the converted value; that is, if the - flag character is present.

The + flag If a + flag is present, then the result of a signed conversion will always begin with a sign; that is, an explicit + will precede a converted positive value. (Negative values are always preceded by - regardless of whether a plus-sign flag is specified.) This flag is relevant only for the conversion operations `d`, `e`, `E`, `f`, `g`, `G`, and `i`.

The conversion operations are very complicated. A brief summary is presented in table 15-1; detailed explanations then follows.

Table 15-1: Input conversions (scanf, fscanf, sscanf)

| Conversion letter | Size specifier | Argument type | Input format |
|-------------------|----------------|------------------|--|
| d | <i>none</i> | int * | [-+]dd...d |
| | h | short * | |
| | l | long * | |
| i | <i>none</i> | int * | [-+][0x]dd...d ¹ |
| | h | short * | |
| | l | long * | |
| u | <i>none</i> | unsigned * | [-+]dd...d |
| | h | unsigned short * | |
| | l | unsigned long * | |
| o | <i>none</i> | unsigned * | [-+]dd...d ² |
| | h | unsigned short * | |
| | l | unsigned long * | |
| x | <i>none</i> | unsigned * | [-+][0x]dd...d ³ |
| | h | unsigned short * | |
| | l | unsigned long * | |
| c | <i>none</i> | char * | a fixed-width sequence of characters |
| s | <i>none</i> | char * | a sequence of non-whitespace characters |
| p | <i>none</i> | void ** | a sequence of characters such as output with %p in fprintf. |
| n | <i>none</i> | int * | none; the number of characters read is stored in the argument |
| | h | short * | |
| | l | long * | |
| f, e, g | <i>none</i> | | any floating-point constant or decimal integer constant, optionally preceded by - or + |
| | l | | |
| | L | | |
| [| <i>none</i> | char * | a sequence of characters from a scanning set |

¹ The base of the number is determined by the first digits in the same way as for C constants.

² The number is assumed to be octal.

³ The number is assumed to be hexadecimal regardless of the presence of 0x.

specified, then the prefix is a space. Otherwise, the prefix is empty. The # flag is not relevant to the d and i conversions.

Examples of the d conversion

| Sample format | Sample output Value = 45 | Sample output Value = -45 |
|---------------|-----------------------------|------------------------------|
| %12d | 45 | -45 |
| %012d | 000000000045 | -000000000045 |
| % 012d | 000000000045 | -000000000045 |
| %+12d | +45 | -45 |
| %+012d | +000000000045 | -000000000045 |
| %-12d | 45 | -45 |
| %- 12d | 45 | -45 |
| %-+12d | +45 | -45 |
| %12.4d | 0045 | -0045 |
| %-12.4d | 0045 | -0045 |

The u conversion Unsigned decimal conversion is performed. The argument should be of type unsigned if no size modifier is used, type unsigned short if h is used, or type unsigned long if l is used.

The converted value consists of a sequence of decimal digits that represents the value of the argument. This sequence is as short as possible but not shorter than the specified precision. The converted value will have leading zeros if necessary to satisfy the precision specification; these leading zeros are independent of any padding, which might also introduce leading zeros (see below). If the precision is 1 (the default), then the converted value will not have a leading 0 unless the argument is 0, in which case a single 0 is output. If the precision is 0 and the argument is 0, then the converted value is empty (the null string). The prefix is always empty. The +, space, and # flags are not relevant to the u conversion operation.

Examples of the u conversion

| Sample format | Sample output Value = 45 | Sample output Value = -45 |
|---------------|-----------------------------|------------------------------|
| %14u | 45 | 4294967251 |
| %014u | 00000000000045 | 00004294967251 |
| %#14u | 45 | 4294967251 |
| %#014u | 00000000000045 | 00004294967251 |
| %-14u | 45 | 4294967251 |
| %-#14u | 45 | 4294967251 |
| %14.4u | 0045 | 4294967251 |
| %-14.4u | 0045 | 4294967251 |

The o Conversion Unsigned octal conversion is performed. The argument should be of type unsigned if no size modifier is used, type unsigned short if h is used, or type unsigned long if l is used.