Proposal for C2X
WG14 N2586

Title:	footnote about sufficient formatting precision
Author, affiliation:	C FP group
Date:	2020-10-11
Proposal category:	Editorial
Reference:	N2573

The **`fprintf`** specification for **`a`**,**`A`** style formatting in 7.21.6.1 #8 in the current C2X draft (N2573) says:

> … if the precision is missing and **`FLT_RADIX`** is not a power of 2, then the precision is sufficient to distinguish300) values of type **`double`**
>
> …
>
> 300) The precision *p* is sufficient to distinguish values of the source type if $16^{p-1} > b^n$ where *b* is **`FLT_RADIX`** and *n* is the number of base-*b* digits in the significand of the source type. A smaller *p* might suffice depending on the implementation's scheme for determining the digit to the left of the decimal-point character.
>
> .

Problem 1: The notation in the footnote is not consistent with the rest of the standard. The precision *p* in the footnote refers to a formatting precision (in the footnote anchor) which the standard denotes with *P*. The footnote uses *n* to refer to what in the C model is the type precision *p*, and *p* is used further down in the paragraph containing the footnote anchor. (The difference between the characters *P* and *p* is clear enough in N2478.)

Problem 2: The sufficiency inequality can be relaxed. It is based on the property:

> $B^{P-1} > b^p$ implies base-*B* numbers of precision *P* distinguish base-*b* numbers of precision *p*.

Ref.: D. W. Matula, "The base conversion theorem", Proc. Amer. Math. Soc., vol. 19, no. 3, pp. 716-723, June 1968. C prints floating-point numbers in hexadecimal form:

[**-**]*h* **.** *h…h***p**±*d*

where each *h* represents a hexadecimal digit, *d* is a decimal integer power of 2, and the *h* to the left of the decimal point character is nonzero (but otherwise unspecified). Where *P* is the formatting precision, i.e., the number of hexadecimal

digits to the right of the decimal point character, the hexadecimal form can represent at least all binary numbers with precision $4P+1$.

Using the property above,

$2^{(4P+1-1)} > b^p$

or

$16^P > b^p$

is sufficient for the hexadecimal output to distinguish base-b numbers of type precision p. (The footnote anchor is referring to b not a power of 2, so we needn't improve the inequality further for that case.)

The following suggested change addresses both of these problems.

Suggested change:

1. Replace footnote 300 in 7.21.6.1 #8 with:

> 300) The formatting precision P is sufficient to distinguish values of the source type if $16^P > b^p$ where b (not a power of 2) and p are the base and precision of the source type (5.2.4.2.2). A smaller P might suffice depending on the implementation's scheme for determining the digit to the left of the decimal-point character.