Proposal for C2Y
WG14 N3324

Title:	Improved wording around “pole error”
Author, affiliation:	CFP group
Date:	2024-09-02
Proposal category:	Editorial
Reference:	N3301

This document suggests an update to clarify the definition of a pole error, and provides rationale for the changes.

Suggested change:

Changes for 7.12.2#3 first sentence

> Similarly, a *pole error* occurs in cases where the arguments are finite and where the mathematical function typically has an infinite limit (for example, a pole error occurs at **`log(0.0)`** because the mathematical log() function has a right-hand limit of $-\infty$ at 0).

Rationale:

The new wording focuses on the common case of an infinite limit at finite arguments, while being broad enough to embrace other conditions. Here are examples:

- The function **`logb()`** is synthetic, not purely mathematical. Its value at 0.0 is $-\infty$, but it is not a limit.
- Subtle distinctions can justify pole errors for **`tanpi(`**$n + 1/2$**`)`**, for integers n, though not for **`tgamma(`**x**`)`** for integral $x < 0$.
- The function **`rsqrt()`** is defined to be $-\infty$ at zero, though that value does not derive from a limit.

The current wording lacks the breadth to cover all of these cases of infinite results from finite arguments.

ISO/IEC 60559 follows this same approach for its divide-by-zero exceptions, which C pole errors are intended to encompass. It identifies the particular cases where divide-by-zero exceptions are signaled, but does not provide an actual definition of the condition.

The new wording removes the problematic wording “exact infinite result”. Exactness applies to library functions, not mathematical ones. The phrase “exact infinity” occurs once in C (F.10.10.1), where it refers to a library function rather than a mathematical one. The expression “exact infinite result” appears exactly once, without definition, in IEC-60559.

The new wording is more conventional mathematical usage, without attempting to capture all the various cases of pole errors in an elaborate definition. The definition is not given in terms of a mathematical pole as defined in complex analysis because one function, **`logb()`**, does not have an

underlying limit, and other functions, like `log()`, have underlying “essential singularities” not simple poles.

This suggested change requires no implementation changes.