Proposal for C2Y
WG14 N3461

Title:	**range error definition followup (updates N3383)**
Author, affiliation:	C FP group
Date:	2025-01-28
Proposal category:	Technical
Reference:	N 2454, N3301, N3383

This update to N3383 addresses the issue raised by Joseph Myers in [SC22WG14.28228]. See the last suggested change below. This update includes no other changes to N3383 and is still based on C2Y draft N3301.

C18 (and N2454) 7.12.1 #4 says

> … a *range error* occurs if and only if the mathematical result of the function cannot be represented in an object of the specified type, due to extreme magnitude.

C23 (and N3301) 7.12.2 #4 says

> A *range error* occurs if and only if the result overflows or underflows, as defined below.

and #5 says

> A floating result overflows if a finite result value with ordinary accuracy265) would have magnitude (absolute value) too large for the representation with full precision in the specified type. A result that is exactly an infinity does not overflow. If a floating result overflows and default rounding is in effect, then the function returns the value of the macro **`HUGE_VAL`**, **`HUGE_VALF`**, or **`HUGE_VALL`** according to the return type, …

Thus C23 defines overflow, and hence range errors for overflow, only for floating results.

The C18 definition appears to allow range errors for functions that return results in integer type. The C23 change excludes them.

The C23 definition of range errors aligns with ISO/IEC 60559 which regards out-of-range cases for integer formats as "invalid" floating-point exceptions, i.e. as domain errors. However, the C23 change did not account for the specification of certain functions with integer return types (e.g. **`ilogb`**, **`lrint`**, **`lround`**) which explicitly allow range errors. For example, for **`ilogb`**, 7.12.7.8 #2 says

> … If the correct value is outside the range of the return type, the numeric result is unspecified and a domain error or range error may occur.

The occurrence of a range error in these cases is now disallowed by the definition of range errors.

The following suggestions are intended to make the function specifications consistent with the range error definition (essentially the C23 one), but avoid invalidating existing implementations or user code that provide or depend on a range error.

Suggested changes:

Insert after the first sentence in 7.12.2 #4:

> Likewise, a range error occurs if and only if the result overflows or underflows, as defined below. Also, a range error may occur if the function result has integer type and the result is outside the range of the type; however, the occurrence of a range error in such cases is an obsolescent feature. The description of each function lists …

In 7.12.7.8 #2, delete words:

> … If the correct value is outside the range of the return type, the numeric result is unspecified and a domain error ~~or range error~~ may occur.

In 7.12.10.5 #2, delete words:

> … If the rounded value is outside the range of the return type, the numeric result is unspecified and a domain error ~~or range error~~ may occur.

In 7.12.10.7 #2, delete words:

> … If the rounded value is outside the range of the return type, the numeric result is unspecified and a domain error ~~or range error~~ may occur.

In 7.33.9, append the paragraph:

> The optional occurrence of an overflow range error when the function result has integer type and the result is outside the range of the type is an obsolescent feature.