This paper summarizes the changes that are expected to make it into the next version of the C standard ("C23") from a number of TS's relating to the binding of recent versions of IEC 60559 and IEEE 754 into C. In particular:

TS 18661-1: Binary floating point arithmetic (required by IEC 60559)
TS 18661-2: Decimal floating point arithmetic (IEC 60559 requirements, supersedes ISO/IEC TR 24732)
TS 18661-3: Interchange and extended types (optional by IEC 60559)
TS 18661-4a: Supplementary mathematical functions (optional by IEC 60559, reduction functions under 4b not added to C)

A summary of each TS is given below (part numbers correspond to the TS name after the "-").

Part 1: Binary floating point
Macros added to give integer type widths.
Macros and functions added to query and set floating point environment flags and modes.
Macros and functions added (ex. fromfpx, roundeven, fmaxmag, llogb, nextup, fadd, ffma, totalorder, canonicalize, setpayload, strfromd including tgmath versions).
Add constant rounding modes: #pragma STDC FENV_ROUND direction - Some standard functions are affected by this (Ex. cos, exp, log, scalbn, cbrt, lgamma, rint, fadd, wcstod, wprintf).
Add macros for signaling NaNs.
Add macros for queries on the classification of floating-point values (Ex. iscanonical, issignaling, iszero).

Part 2: Decimal floating point
Distinct types (from float, double and long double) conditionally added for decimal floating-point types: _Decimal{32,64,128}.
Literal suffixes added for decimal floating-point types: df/DF, dd/DD, dl/DL.
Macros conditionally added to provide information about decimal floating-point values (Ex. Min, max values, DEC_EVAL_METHOD).
Macros and functions conditionally added to provide decimal floating-point functions and environment modes corresponding to binary floating point (Ex. fe_dec_setround, DEC_INFINITY, cosd32, expd128, fabsd64, lroundd64, nextafterd32, strtod64, dMadddN, dMmuldN).
Functions conditionally added to get decimal floating-point type specific information (Ex. samequantumd32, llquantexpd64).
Functions conditionally added to convert between different decimal floating-point encodings (Ex. encodedecd128, decodebind64).
Format specifiers added to the printf/scanf family of functions to handle decimal floating-point types.

Part 3: Interchange (fN, dN) and extended (fNx, dNx) types - conditionally normative annex
Distinct types added for binary and decimal floating-point interchange and extended

types (Ex. _Float32, _DecimalN, _FloatNx)
Literal suffixes added for binary floating-point and decimal floating-point types: fN/FN, fNx/FNx, dN/DN, dNx/DNx.
Binary and Decimal floating-point information macros generalized to interchange and extended types (Ex. FLTN_MAX, DECNX_TRUE_MIN).
Binary and Decimal floating-point functions, type generic macros, and other macros generalized to interchange and extended types (Ex. coshfN, ceilfNx, sinhdNx, dMadddNx, strtofN, FP_FAST_FMADDFN, FLTN_SNAN, and tgmath versions).
Decimal floating-point specific functions generalized to interchange and extended types (Ex. dMencbindN, strfromencdecdN, quantizedNx).
Binary complex and imaginary types generalized to interchange and extended types (Ex. _FloatN _Imaginary, _FloatNx _Complex)
Binary complex floating-point functions generalized to interchange and extended types (Ex. cexpfN, crealfNx).
Evaluation method macro values updated to include interchange and extended types (DEC_EVAL_METHOD N for _DecimalN, FLT_EVAL_METHOD N+1 for _FloatNx).
Encoding and decoding functions to allow conversions between non-arithmetic interchange formats (Ex. decodefN, dMecndecdN).

Part 4a: Supplementary math functions
New functions added for standard types and conditionally added for binary and decimal, interchange and extended floating-point types (Ex. pown, acospifN, exp2m1ldN, compoundndNx).