Problem 2.86 Solution:

We have found that working through floating point representations for small word sizes is very instructive. Problems such as this one help make the description of IEEE floating point more concrete.

Description	Hex	M	E	V
-0	`8000`	0	-62	-0
Smallest value > 2	`4001`	$\frac{257}{256}$	1	$\frac{257}{128}$
512	`4800`	1	72	—
Largest denormalized	`00FF`	$\frac{255}{256}$	-62	255×2^{-70}
$-\infty$	`FF00`	—	—	—
Number with hex representation `3BB0`	—	$\frac{27}{16}$	-4	$\frac{27}{256}$

Problem 2.87 Solution:

This problem tests a lot of concepts about floating-point representations, including the encoding of normalized and denormalized values, as well as rounding.

Format A		Format B		Comments
Bits	Value	Bits	Value	
`1 01111 001`	$\frac{-9}{8}$	`1 0111 0010`	$\frac{-9}{8}$	
`0 10110 011`	176	`0 1110 0110`	176	
`1 00111 010`	$\frac{-5}{1024}$	`1 0000 0101`	$\frac{-5}{1024}$	Norm $\rightarrow$ denorm
`0 00000 111`	$\frac{7}{131072}$	`0 0000 0001`	$\frac{1}{1024}$	Smallest positive denorm
`1 11100 000`	-8192	`1 1110 1111`	-248	Smallest number $> -\infty$
`0 10111 100`	384	`0 1111 0000`	$+\infty$	Round to ∞.

Problem 2.88 Solution:

This problem requires students to think of the relationship between `int`, `float`, and `double`.

A. `(float) x == (float) dx`. Yes. Converting to `float` could cause rounding, but both `x` and `dx` will be rounded in the same way.

B. `dx - dy == (double) (x-y)`. No. Let `x` = 0 and `y` = $TMin_{32}$.

C. `(dx + dy) + dz == dx + (dy + dz)`. Yes. Since each value ranges between $TMin_{32}$ and $TMax_{32}$, their sum can be represented exactly.

D. `(dx * dy) * dz == dx * (dy * dz)`. No. Let `dx` = $TMax_{32}$, `dy` = $TMax_{32} - 1$, `dz` = $TMax_{32} - 2$. (Not detected with Linux/GCC)

E. `dx / dx == dz / dz`. No. Let `x = 0`, `z = 1`.