AERE 361: Lab 9

Sean Hellar

Due 5 APR 2022

1 Report Questions

1.)

Table 1: Bit Integer Ranges

Size	Unsigned		Signed	
	Min. Value	Max. Value	Min. Value	Max. Value
8-bit	0	255	-128	127
16-bit	0	65535	-32678	32767
32-bit	0	$2^{32} - 1$	-2^{31}	$2^{31} - 1$
64-bit	0	$2^{64} - 1$	-2^{63}	$2^{63} - 1$

2.) **88:** `0101100` **0:** `00000000` **1:** `00000001` **127:** `01111111` **255:** `11111111`

3.) **88:** `0101100` **-44:** `11010100` **-1:** `11111111` **0:** `00000000` **1:** `00000001` **-128:** `10000000` **127:** `01111111`

4-7.) **Normalized:** $\pm 2^{-126}$ to $(2-2^{-23})*2^{127}$
Denormalized: $\pm 2^{-149}$ to $(1-2^{-23})*2^{127}$

8-11.) **Normalized** $\pm 2^{-1022}$ to $(2-2^{-52})*2^{1023}$
Denormalized $\pm 2^{-1074}$ to $(1-2^{-52})*2^{1023}$

2 Exercise 4

When using single precision, and `n = 10` then the solution converges to six, however when using double precision, and `n = 20` the solution converges to 100. The reason they converge on these numbers is becuase the decimal becomes to large for the magic box to tell the difference so it converges on its respective value.

I'm not entirely sure why it breaks through 6 on iteration 16 in double precision, and I doubt I'll ever find out why. It makes a huge jump from 14 to 64 then then to 99, which makes little sense in either mathematics or computing.

3 Sources

https://tex.stackexchange.com/questions/297564/why-is-my-table-before-the-section-title

https://latex-tutorial.com/subscript-superscript-latex/

https://en.wikipedia.org/wiki/Single-precision_floating-point_format

https://en.wikipedia.org/wiki/IEEE_754-1985

http://steve.hollasch.net/cgindex/coding/ieeefloat.html