1. NVIDIA has a "half" format, which is similar to IEEE 754 except that it is only 16 bits wide. The leftmost bit is still the sign bit, the exponent is 5 bits wide and the Fraction field is 10 bits long. A hidden 1 is assumed.

Representation range of the NVIDIA 'half format'

For each of the following, write the *binary value* and *the **corresponding** decimal value* of the 16-bit floating point number that is the closest available representation of the requested number. If rounding is necess2ary use round-to-nearest. Give the decimal values either as whole numbers or fractions.

(i) The number '0', Charge of an electron ($-1.6\text{x}10^{-19}$), smallest positive *normalized* number, smallest positive *denormalized* number, largest positive *normalized* number, largest positive *denormalized* number, distance b/w Earth and Neptune in inches: `171,072,000,000,000`

Number	Binary	Decimal
0	00000 0000000000	0
Charge of an electron: $-1.6 \text{ x } 10^{-19}$ (C)	1 00000 0000000000	0
Smallest positive normalized number	0 00001 0000000000	$6.103515625 \text{ x } 10^{-5}$
Smallest positive **denormalized** number > 0	0 00000 0000000001	$5.960464477 \text{ x } 10^{-8}$
Largest positive **denormalized** number > 0	0 00000 1111111111	$\sim 6.103515625 \text{ x } 10^{-5}$
Largest positive number < infinity	0 11110 1111111111	= 65536
Average distance b/w proton and neutron in Hydrogen atom = $0.8751 \text{ x } 10^{-15}$ m	0 00000 0000000000	0
Distance between Earth and Neptune in inches = 171,072,000,000,000	0 11111 0000000000	overflow

<u>**Smallest normalized number > 0**</u>: 0 00001 0000000000; bias = 15; val of exponent = 1-bias = -14, Significand = $1.000\ldots0_2$ = 1, Value in decimal ~ 1.00 x 2^{-14} = $6.103515625 \text{ x } 10^{-5}$

<u>**Smallest positive denormalized number > 0**</u>: 0 00000 0000000001; value of exponent = -14; significand = 0.0000000001 = 1 x 2^{-10}; so smallest denormalized number = 1 x 2^{-10} x 2^{-14} = 2^{-24} = $5.960464477 \text{ x } 10^{-8}$

<u>**Largest possible denormalized number > 0:**</u> 0 00000 1111111111; value of exponent = 1-15 = -14; significand = 0.1111…1 ~ 1.0; so largest denormalized number = 1 x 2^{-14} ~ $6.103515625 \text{ x } 10^{-5}$

Largest positive number < inf: 0: `0 11110 1111111111; value of exponent = 30 – 15 = 15; Significand =` $1.11…1_2$ `= ~ 2; value in decimal ~ 2 x` $2^{15} = 2^{16}$ `= 65536`

Overflow: `represented by +infinity: 0 11111 0000000000`

2. Implement in RISC V these line of code in C:

(i) `f = g - A[B[C[27]]]`

(ii) `f = g - A[ C[10] + B[11] ]`

(iii) `A[i] = 4B[4i-44] + 3C[32i+32]`

3. When parallelizing an application, the ideal speedup is speeding up by the number of processors. This is limited by two things: percentage of the application that can be parallelized and the cost of communication. Amdahl's law takes into account the former but not the latter.

(i) What is the speedup with N processors if 80% of the application is parallelizable, ignoring the cost of communication?

`Speed-up = 1/(0.2 + 0.8/N)`

What is the speedup with 8 processors if, *for every processor added, the communication overhead is 0.5% of the original execution time*?

`Speed-up = 1/(0.2 + 8 x 0.005 + 0.8/8) = 2.94`

What is the speedup with 8 processors if, for every time the *number of processors is doubled*, the communication overhead is *increased by 0.5% of the original execution time*?

`Speed-up = 1/(0.2 + 3 x 0.005 + 0.8/8) = 3.17`

(ii) Write the general equation that solves this question: What is the number of processors with the highest speedup in an application in which P% of the original execution time is parallelizable, and, for every time the number of processors is doubled, the communication is increased by 0.5% of the original execution time?

$$\frac{d}{dN}\left(\frac{1}{(1-P) + logN \times 0.005 + P/N}\right) = 0$$