Ahmed Saaif 661925946 ahmeds7@rpi.edu

Monday, February 17, 2020 5:08 PM

1. What decimal number does the bit pattern 0X1EF00F00 represent if it is a floating point number? Use the IEEE 754 standard.

   $1EF00F00_{16} \rightarrow 00011110111100000000111100000000_2$

   Sign bit: 0 (positive)
   Exponent: $00111101 - 127_{10} = -66_{10}$
   significand: $1.11100000000111100000000_2 \rightarrow 1.8754578$

   Answer: $1.8754578 * 2^{-66}$ or $2.5417 * 10^{-20}$

2. Write down the binary representation of the decimal number 31.7 assuming the IEEE 754 single precision format.

   $31_{10} \rightarrow 11111_2$
   Sign bit: 0
   $.7_{10}$ to binary. Use python script to convert.

   Fraction: 10110011001100110011001

   Whole number: 11111.10110011001100110011001
   Cut off: $1.11111011001100110011001 * 2^5$

   Exponent: $5 \rightarrow 101 + 127_{10} \rightarrow 10000100_2$

   Answer: $01000010011111011001100110011001_2$

3. Write down the binary representation of the decimal number 31.5 assuming the IEEE 754 double precision format (note that the bias in double precision format is 1023).

   $31_{10} \rightarrow 11111_2$
   Sign bit: 0
   $.5_{10}$ to binary. Use python script to convert

   Fraction: 1000000000000000000000000000000000000000000000000000

   Whole Number: 11111.1000000000000000000000000000000000000000000000000000
   Cut off: $1.1111100000000000000000000000000000000000000000000000 * 2^5$

   Exponent: $5 \rightarrow 101 + 1023_{10} \rightarrow 10000000100_2$

   Answer: $0100000001001111100000000000000000000000000000000000000000000000_2$

4. For a 16-bit half precision format, the left most bit is the sign bit, the exponent is 5 bits wide, and the fraction is 10 bits long. A hidden 1 is assumed. Write down the bit pattern to represent 4.8125 x 10-2 assuming a version of this format, which uses an excess-15 format to store the exponent.

   Number: 0.048125

   Sign: 0 (positive)

   Exponent: $0+15=15 \rightarrow 01111_2$

   Fraction: 0000110001_2

   Cut off: 0000110001

   Answer: $0011110000110001_2 \rightarrow 0x3C31_{16}$

Python Script used for conversion:

main.py saved

```
x = 0.048125  #stores the decimal number
y= []

for i in range(10):
  x=2*x
  if x>=1:
    y.append(1)
    x-=1
  else:
    y.append(0)

for i in y:
  print(i,sep = " " ,end="")

```