Homework

1. Given a (very) tiny computer that has a word size of 5 bits, what are the smallest negative numbers and the largest positive numbers that this computer can represent in each of the following representations?
Hint: show your answers in binary and decimal forms.

a. One's complement

Ans: The smallest negative number that can be represented in 5 bits of one's complement is 10000 in binary forms, which is represented as -15 in decimal forms. The largest positive number that can be represented is 01111, which is represented as 15 in decimal forms.

note: Range of 1's complement for n bit number is from $-2^{n-1} - 1$ to $2^{n-1} - 1$

b. Two's complement

Ans: The smallest negative number that can be represented in 5 bits of one's complement is 10000 in binary forms, which is represented as -16 in decimal forms. The largest positive number that can be represented is 01111, which is represented as 15 in decimal forms.

note: Range of 1's complement for n bit number is from -2^{n-1} to $2^{n-1} - 1$

2. Show how the floating point value of 26.625 would be stored using IEEE-754 double precision with excess-1023 exponent (be sure to indicate the sign bit, the exponent, and the significand fields):

Soln: **First, convert 26.625 to binary**

$(26)_{10} = (11010)_2$

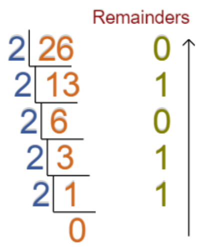
$(0.625)_{10} = (0.101)_2$

$(26.625)_{10} = 11010 + 0.101$

$= 11010.101$

$= 11010.101 \times 2^0$

$= 1.1010101 \times 2^4$



$0.625 \times 2 = 1.250$

$0.250 \times 2 = 0.500$

$0.500 \times 2 = 1.000$

Second, finding the exponent by adding 1023 to the exponent of our binary(4)

4 + 1023 = 1027 convert 1027 to binary

$= (10000000011)_2$

Third, we need mantissa of 52 bits because of IEEE-754 double precision (64 bits)

1010101 add more 45 zero's

= 101010100000000000000000000000000000000000000000

Sign bit of this number is 0 because it is a positive numbers

Answer:

0 sign(1)	10000000011 Exponent(11)	101010100000000000000000000000000000000000000000 Mantissa (52)

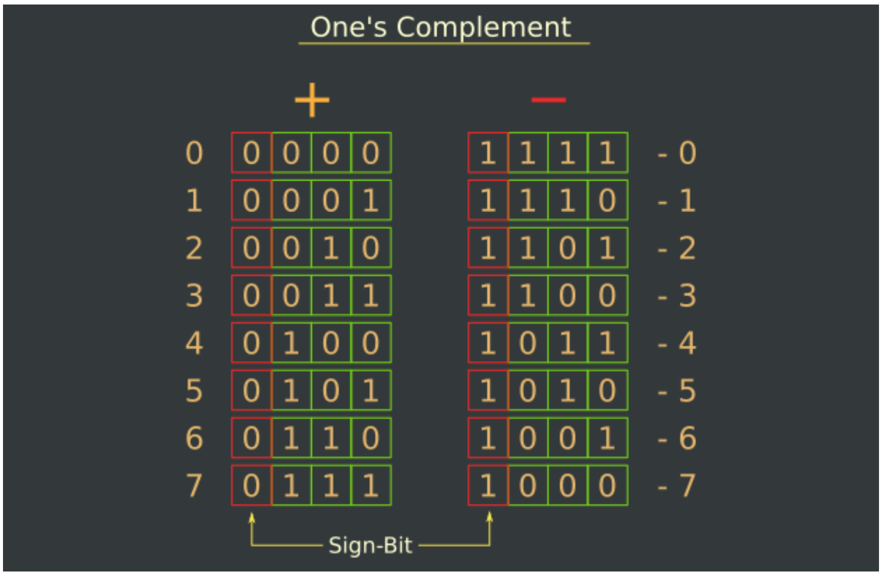
3. Suppose a computer uses 4-bit one’s complement representation. Ignoring overflows, what value will be stored in the variable *j* after the following pseudocode routine terminates?

-2 j // Store -2 in j.

6 k // Store 6 in k.

```
while k ≠ -6
    j=j-1
    k=k+1
end while
```

Answer:



K	J
0110 (6)	-2-1 = -3
0111 (7)	-3-1 = -4
1000 (-7)	-4-1 = -5
1001 (-6)	-5-1 = -6

The value that will be stored in the variable j after the loop of pseudocode routine terminated is -6

4. Convert 9.5 and 1.25 to unsigned binary, then compute the multiplication of the two values. Answer in 14-bit floating point model with bias-16 exponent.

Soln: **Convert 9.5 to unsigned binary**

$(9)_{10} = (1001)_2$

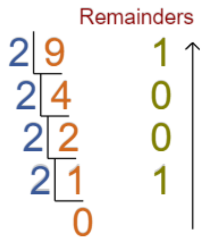
$(0.5)_{10} = (0.1)_2$

$(9 + 0.5)_{10} = 1001 + 0.1$

$= 1001.1$

$= 1001.1 \times 2^0$

$= 1.0011 \times 2^3$



$0.5 \times 2 = 1.0$

Convert 1.25 to unsigned binary

$(1)_{10} = (0001)_2$

$(0.25)_{10} = (0.01)_2$

$(1 + 0.25)_{10} = 0001 + 0.01$

$= 0001.01$

$= 0001.01 \times 2^0$

$0.25 \times 2 = 0.50$

$0.50 \times 2 = 1.00$

Multiply the binary numbers

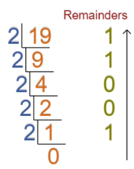
$(1.0011 \times 2^3) \times (0001.01 \times 2^0) = (1.0011 \times 1.01 \times 2^{(3+0)})$

$= (1.011111 \times 2^3)$

Biased exponent for 3

16 + 3 = 19 Convert 19 to binary

$(19)_{10} =$ 10011



Sign bit of this number is 0 because it is a positive numbers

Answer:

0 sign(1)	10011 Exponent(5)	01111100 Mantissa (8)