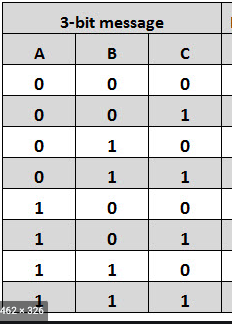
**BIT Manipulation**

1. A bit is a basic unit of computation . It can either be 1 or 0.
2. 1 Byte is equal to 8 BITS.
3. The range of value a 3 bit message can store is 2^3= 8 and n bit message can store value ranging from 2^n



1. Size of int is 4 bytes .

Int = 4 bytes = 4\*8 bits .

Therefore the range of values it can store is 2^32.

Signed int can store value ranging from -2,147,483,648 to 2,147,483,647

Unsigned int can store value ranging from 0 to 4,294,967,295

1. Any number that we see in real life is a decimal number.

Example 55 is a decimal number. It has a base 10.

1. Octal number has a base 8. Values ranging from (0,1,2,....7)
2. Binary number has a base 2 (0,1)
3. Hexadecimal number has a base 16 . Values ranging from (0,1,2,3,4,5...9,A,B,C,D,E,F)

Where A-10, B-11, C-12, D-13, E- 14, F- 15.

1. **Converting any number into its decimal format.**

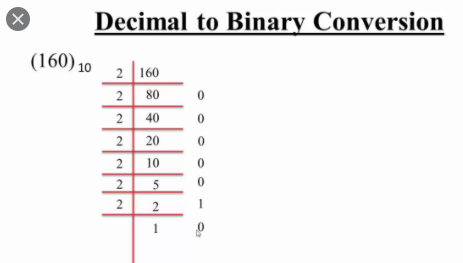
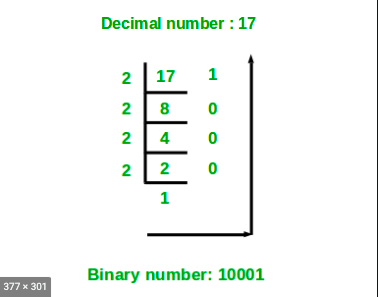
(3192)10 =2\*10^0 + 9\*10^1 + 1\*10^2 + 3\* 10^3 =2\*1 + 9\*10 + 1\*100+3\*1000= (3192)10

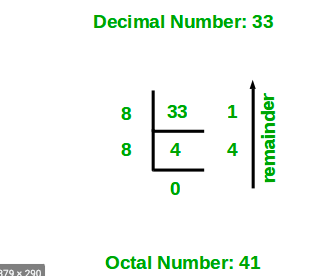
(3172)8 = 2\*8^0 + 7\*8^1 + 1\*8^2 + 3\*8^3 = 2\*1 + 7\*8 + 1\*64 + 3\*512 =(1658)10

(1010)2= 0\*2^0 + 1\*2^1 + 0\*2^2 + 1\*2^3 = 0 + 2 + 0 + 8 = (10)10

1. Converting a number from decimal to any other format.

Keep on dividing the number by the base of the other format and note the remainder of the number.



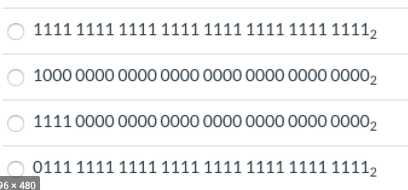


1. What will be length number if an integer number is converted into hexadecimal format

Int = 4 bytes = 4\*8 bits =32 bit.

In the binary representation it will be 32 bits long.

Example:-



Range of number we can have is 2^32 == (2^4)^8 == (16)^8.

Therefore the maximum Hexadecimal number will be size 8.

Ranging from

0000 0000

0000 0001

0000 0002

.

.

.

FFFF FFFE

FFFF FFFF

1. Watch this video to understand the various operations :-

<https://www.youtube.com/watch?v=NLKQEOgBAnw&ab_channel=HackerRank>

1. Understand about the right and left shift operations from this video

<https://www.youtube.com/watch?v=MiJdgxTWaFs&ab_channel=ApnaCollege>

1. Property of right shift operation :-

Those many bits are remove from the right

110101>>3 == 110

**Property**

Whenever we do a right shift we actually perform **integer** division the number by 2^n where n is the number specified in right operation

Example:-

15 >>0 = 15 /(2^0) =15.

15 >>1 = 15/ (2^1) =7

15 >>2 = 15(2^2) =3

100>>4 = 100/(2^4) = 6

(Note the division here is integer division)

1. Property of left shift is multiplying the number by 2^n.

Those many bits are added to the right

110101<<3 == 110101000

Example :-

6<<4 = 6\*(2^4) = 96

4<<7 = 4 \*(2^7) = 512

Watch these 2 videos :-

1. <https://www.youtube.com/watch?v=KE5Axm7uzok&ab_channel=NideeshTerapalli>
2. <https://www.youtube.com/watch?v=htX69j1jf5U&ab_channel=NideeshTerapalli>

**16. Or Operation**

101010

110110

111110

**Property**

Whenever we do an OR operation between 2 numbers A OR B = R

R is always greater than( greatest between A and B)

Try solving this question using the property :-

<https://leetcode.com/problems/bitwise-ors-of-subarrays/>

**17. AND Operation**

101010

110110

100010

**Property**

Whenever we do an AND operation between 2 numbers A AND B = R

R is always less than( smallest between A and B)

Try solving this question using the property :-

<https://leetcode.com/problems/bitwise-and-of-numbers-range/>

**18. XOR Operation**

XOR between same bits gives 0 (1 ^1 =0 , 0 ^0 =0

And XOR between different bits gives 1 (1^0 =1 , 0 ^1 =1 )

**Property**

Whenever we do an XOR operation between 2 same numbers output is 0 and XOR between a number and 0 we get the same number A .

A^A =0

A^0 =A

Try to Solve the following question using XOR property

1. Swap value in a variable without using 3 variable
2. If an array contains 2n+1 numbers with n numbers each have 1 duplicate .find the number which does not have any duplicate.