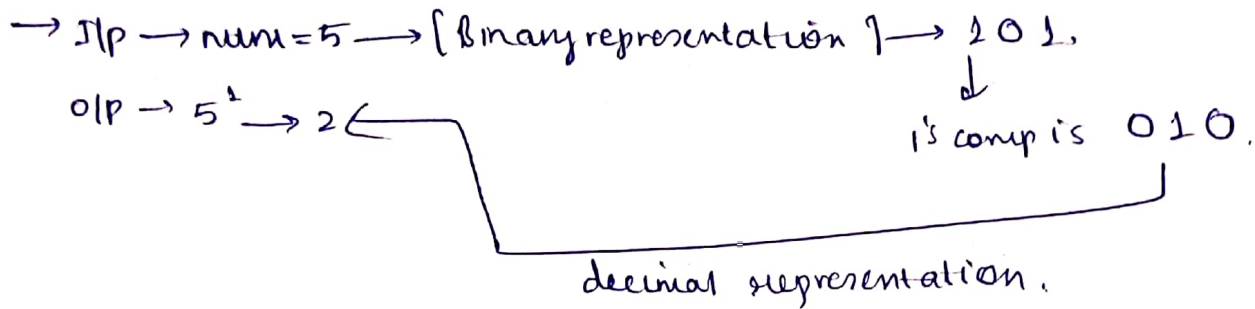


Day 6:- Programming Basics - III

21/4/2022

Problem solving:-

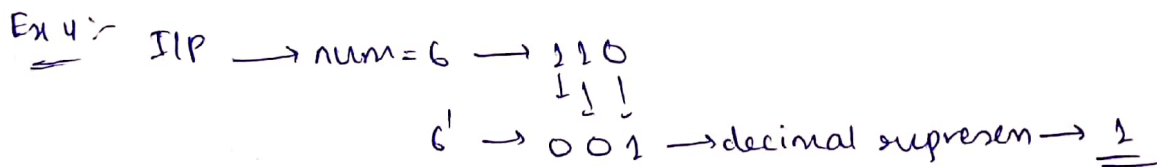
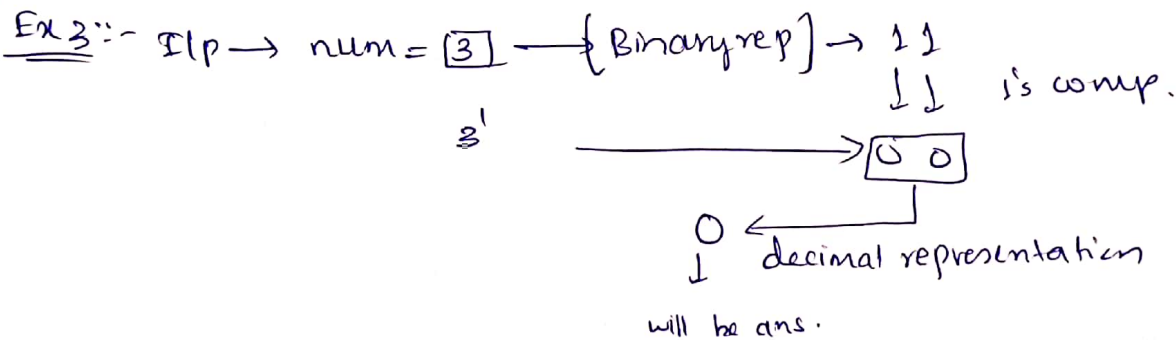
Leetcode prob 1:- Number complement



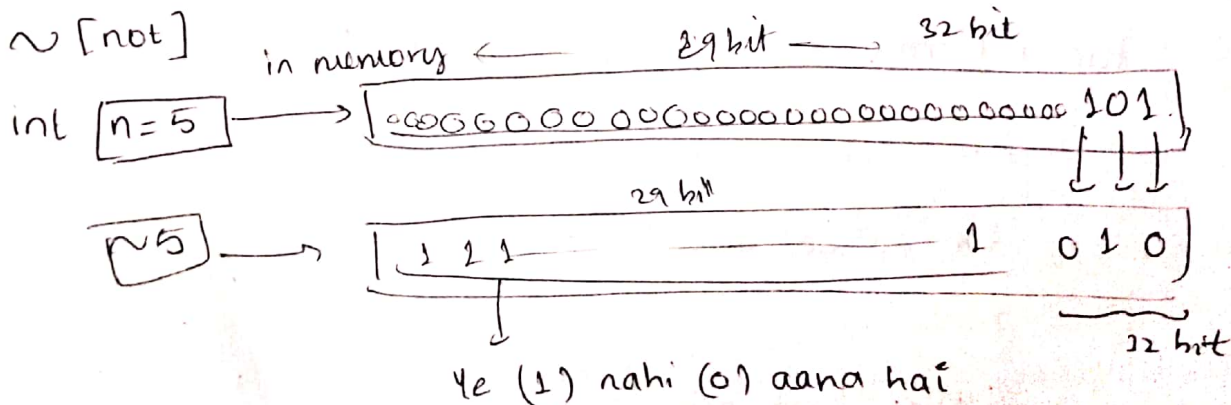
Ex:-

② num → 1 → [1]

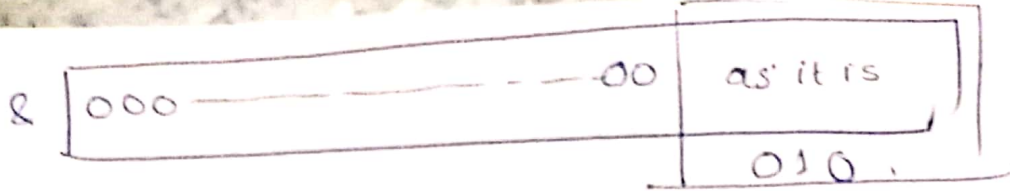
(1's comp) 1' → [0]



we know,

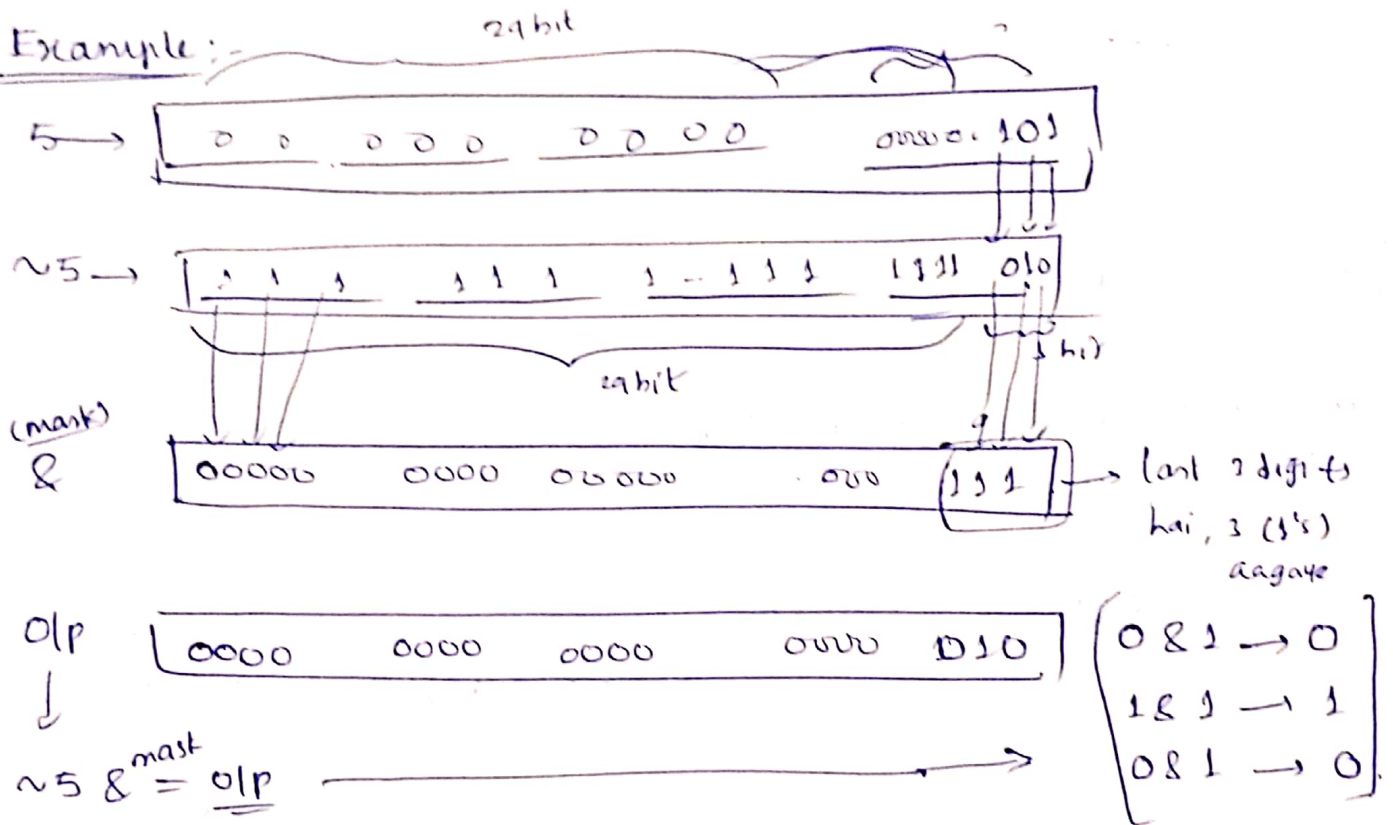


mask

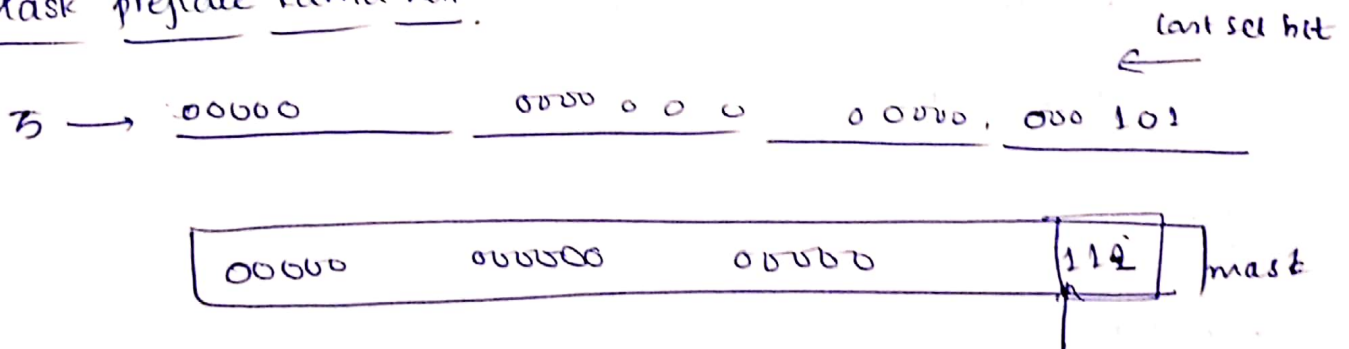


000 ----- 00 010 → ye answer kaana hai

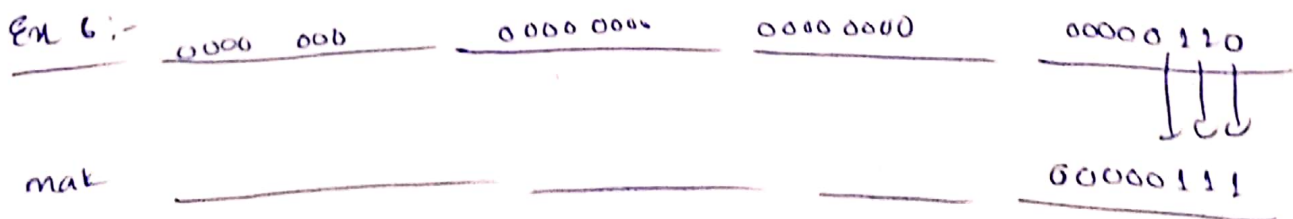
Example:-



Mask prepare karna hai.



// last 3 bit ke basis pe mask banata hai



Ex 3:-

mask

$n=0 \rightarrow$

<u>0000 0000</u>	<u>0000 0000</u>	<u>0000 0000</u>	<u>0000 0000</u>
$\ll 1$			
<u>0000 0000</u>	<u>0000 0000</u>	<u>0000 0000</u>	<u>0000 0000</u> 1
$\ll 1$			
<u>0000 0000</u>	<u>0000 0000</u>	<u>0000 0000</u>	<u>0000 0000</u> 11
$\ll 1$			
<u>0000 0000</u>	<u>0000 0000</u>	<u>0000 0000</u>	<u>0000 0000</u> 111
$\ll 1$			
<u>0000 0000</u>	<u>0000 0000</u>	<u>0000 0000</u>	<u>0000 0000</u> 1111

// LeetCode 1 Number complement

```
int findComplement(int num) {
```

```
    int mask = 0;
```

```
    while (mask < num) {
```

```
        mask = (mask << 1) | 1; [left shift karne ke
```

baad, 1 add karna hai]

```
    }
```

```
    int ans = (~num) & mask;
```

```
    return ans;
```

```
}
```

```
}
```

approach:-

$5 \rightarrow 5^c$ [complement]

5

$5 \rightarrow 00 \dots 00101$
 $\sim 5 \rightarrow 11 \dots 11010$

Zero hoga hai

as it is hoga hai

1's comp
 $101 \rightarrow 010$

mask \rightarrow 00 - - - - 00 (saari honi chahiye)

I know, $0 \& 1 \rightarrow 0$
 $1 \& 1 \rightarrow 1$.

111

I need to create ye waala mask

mask,

while(mask < num),

mask \rightarrow 000 - - - - 0000

\ll I [left shift]

000 - - - - 0000

I want 1 here

so,

0000 - - - - 0001

[or 1] II [uske baad or with 1]

1 \ll III [left shift]

000 - - - - 00010

[or 1] IV [uske baad or 1]

000 - - - - 0011

\ll V [left shift]
 [or 1] VI [or 1]

mask \leftarrow 000 - - - - 00111

\rightarrow mask which we wanted

ab 5 \rightarrow 000 - - - - 00101

$\sim 5 \rightarrow$ 111 - - - - 11010

mask 0000 - - - - 00010

&

mask banane ka approach

step I \rightarrow left shift

step II \rightarrow or 1. karna hai

\rightarrow ye o/p me aana hai

~~while(<=>)~~

while(mask <= num) → why can't we do

ex: - 7 → 0000 - - - - 111

→ ~7 → 11111 - - - - 000

mask <= num
7 7

0000 - - - 111

00000 1111 → Jo wrong hai, [left shift ek extra baar hoga, cozot "="]

→ Why Only Leftshift, but not right shift

→ 0000 - - - - 111 ← left side hona hai

~~0111~~

mask = (mask << 1) | 1

↓

mask = mask << 1 → 0000 - - - - 010

mask = mask | 1 → [left most bit hat, jaati hai]
right most bit → '0' add hojata]

or

ka truth table

x	y	o/p
0	0	0
0	1	1
1	0	1
1	1	1

0000 - - 010
⊕ (or) 1

0000 11

Q/p:- Leetcode (number comp)

I/p \rightarrow 10

O/p \rightarrow 5

Dry run:-

int mask = 0

while(mask < num)

{ mask = (mask << 1) | 1;

}

int ans = (~num) & mask;

return ans;

ans = (~5) & 7;
 ↓ ↓
 mask

iske baad ans
me execute hojata,

ans = (~5) & 7;

5 \rightarrow 000 ——— 101

~5 \rightarrow 11111 ——— 010

& 00000 ——— 111

0000000010

ans = 2;

\rightarrow return ans

fn num = 5

mask = 0

1) $0 < 5 \rightarrow T$

2) 000000 $\leftarrow \leftarrow$
left shift

3) 000000 [or]
1 1
—————
0001 $\rightarrow 1 \rightarrow$ mask

$1 < 5 \rightarrow T$

000010 [left shift]

000010 \rightarrow mask
1 [or 1]

00011 \rightarrow mask

$3 < 5 \rightarrow T$

000110 \leftarrow (left shift)

1 \leftarrow [or]

000111 $\rightarrow 7$

$7 < 5 \rightarrow F$

loop se bahar

left shift

or

karna hai
sirf.

another approach for mask:- [instead of left shift]

Step 1: 00000 ———— 0101 → no of bits = count karlo.
= 3
← last set bit [before getting zero]

Step 2:- add 3 one's last me

0000 ———— 0111 → mask obtained.

Lecture 2 qn:-

Power of two:- approach:-

Ip:- $n = 16$

$$\rightarrow 2^4 = 2 \times 2 \times 2 \times 2 = 16$$

→ true

Op:- bataana hai 2^n hai ya nai

ex:- $n = 15 \rightarrow 2^n \rightarrow F$

ex:- $n = 6 \rightarrow 2^n \rightarrow F$

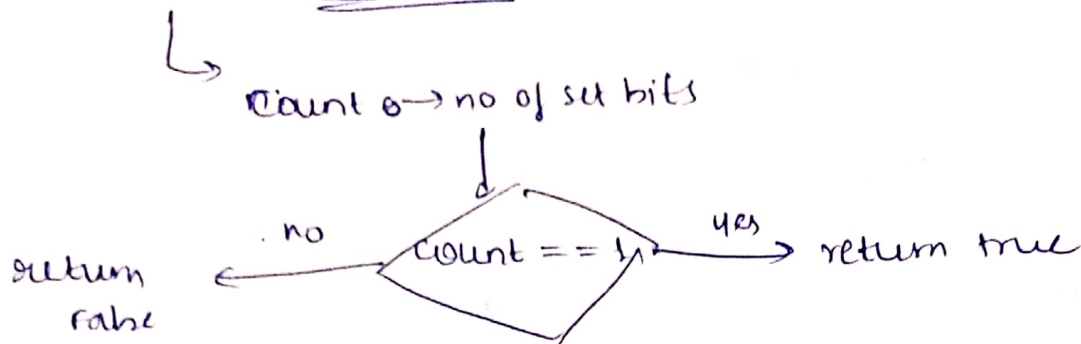
we know:-

$2^0 = 1$	→	0000
$2^1 = 2$	→ True aayega	0001
$2^2 = 4$	→	0011
$2^3 = 8$	→	0111
$2^4 = 16$	→	1000
$2^5 = 32$	→	10000

If $n \rightarrow$ [set bit count = 1] → 2 ki power rahega → T
[no of ones] otherwise → F

If IP = num

Flowchart:



2^n → -ve number keliye bhi false

if ($n \leq 0$)

↓
return false

⇒ How to count set bit

5 → 0000 000 ——— 00101
81

ke baad
same
81 → 1
aaya
↓
set bit

>> [left shift]
000 ——— 001 0 & 1
↓

81 → 0 → aaya
↳ not
set bit

>>
000 ——— 000 1 & 1
↓

>>
000 ——— 00

→ [iska 1 aaya
and remaining
poora iske
baad zero hua
tho, count of
set bits aagaya]

Count \rightarrow no of set bits

$n \& 1 \rightarrow 1 \rightarrow \text{count}++$
 $\rightarrow 0 \rightarrow \text{nth}$

left shift

\times
0000 — 01010
[0 add hojata]
000 — 10100

Right shift

\rightarrow 0000 — 0101 $\boxed{1} \rightarrow \times$
(0 add hojata)
000 — 010

approach 2:-

$n \rightarrow 2^0$
 2^1
 2^2
 $2^3 \rightarrow 1000$

Formula:- $(n \& (n-1)) \rightarrow$ if ans = 0 then $n \rightarrow 2^n \rightarrow T$
else $\rightarrow F$.

ex:- 8 & 7

1000
0111
——
0000

ex:- 16 \rightarrow 10000

15 \rightarrow 01111
——
00000

dry run:- Pseudo code for power of 2

If ($n \leq 0$)

return false

int ans = $n \& (n-1)$

return (ans == 0)

$O(1) \rightarrow$ Time Complexity

approach 3:- for power of 2

Input \rightarrow num

int num, i = 0

while ($n < \text{num}$)

2 $n = \text{pow}(2, i)$

if ($n == \text{ans}$)

return true

i++

}

Code for power of 2

bool isPowerofTwo (int n) {

if ($n \leq 0$)

return false;

int ans = ($n \& (n-1)$); \rightarrow return (ans == 0);

if (ans == 0)

return true;

else

return false;

}

```
if (n <= 0)
```

code optimization

```
return false;
```

```
return ((n & (n-1)) == 0);
```

```
}
```

```
};
```

With approach 1 [counting no of set bits]

pseudo code

```
if (n <= 0)
```

```
return 0;
```

```
int cnt = 0;
```

```
while (n != 0)
```

```
{ if (n & 1) → set bit  
  cnt++;
```

```
n = n >> 1;
```

```
}
```

```
if (cnt == 1)
```

```
return true
```

```
else
```

```
return false.
```

code

```
bool isPowerOfTwo(int n) {
```

```
if (n <= 0)
```

```
return false;
```

```
int count = 0;  
// to count no of set bits
```

```
while (n != 0) {
```

```
  // check last bit
```

```
if (n & 1) {
```

```
  count++;
```

```
}
```

After checking bit, just remove

it from n

```
n = n >> 1;
```

```
} // 2 ki power vale no, has
```

```
if (count == 1)
```

only 1

```
return true;
```

set
bit.

```
else
```

```
return false;
```

```
}
```

```
}
```

Dry run 2

$n = 5$
 $5 \leq 0 \rightarrow F$ | $\text{if } (n \leq 0)$
| $\text{return } 0;$

$\rightarrow 5 \neq 0 \rightarrow T$

$5 \& 1 \rightarrow T$

$\hookrightarrow \text{count} = 0 + 1$
 $= 1$

$n = n >> 1 \rightarrow n = \frac{5}{2} = 2$

$\boxed{n = 2}$

$\rightarrow 2 \neq 0 \rightarrow T$

$2 \& 1 \rightarrow F$

\downarrow
 10
 $\underline{01}$
 0

next

$n = \frac{2}{2} \rightarrow [n = n >> 1]$
 $= 1$

$\rightarrow 1 \neq 0 \rightarrow F$

$1 \& 1 \rightarrow T$

$\hookrightarrow \text{count}++ = \text{count} = 1 + 1$

$\boxed{\text{count} = 2}$

$\rightarrow 0 \neq 0$

$\hookrightarrow F$

\textcircled{T}

\rightarrow So in case of 5 \rightarrow no of set bits = 2

so, it's ~~not a~~ 2^n \times .

Leetcode 3:- Reverse Integer

Ip \rightarrow num = n \rightarrow 1234

Op \rightarrow 4321.

Condⁿ: $\boxed{n} \rightarrow \text{rev_n}$

$\hookrightarrow [-2^{31}, 2^{31} - 1]$

\hookrightarrow If it exceeds then return 0;
and is not in this range.

#include <limits.h>

INT_MIN

INT_MAX

Given \rightarrow n \rightarrow reverse dedo

range se bahar chalagaya tho \rightarrow false
return 0;

n = 1243 \rightarrow ans 3421

approach:-

ans = (10 * ans) + digit

digits use karke \rightarrow number
no banane

ex:- 157 = n

olp \rightarrow 751 banana hai

se 7 nikalna hai

$$157 \% 10 = 7$$

$$\frac{157}{10} = 15$$

$$15 \% 10 \rightarrow 5$$

$$\frac{15}{10} = 1$$

$$1 \% 10 \rightarrow 1$$

$\frac{1}{10} \rightarrow 0$ stop

3 individual digits
obtained

Ex2:- 546
↑↑↑

$\boxed{\% 10} \rightarrow \text{last digit}$

$$546 \% 10 \rightarrow \boxed{6}$$

$$\frac{546}{10} \rightarrow 54$$

$$54 \% 10 \rightarrow \boxed{4}$$

$$\frac{54}{10} \rightarrow 5$$

$$5 \% 10 \rightarrow \boxed{5}$$

$$\frac{5}{10} \rightarrow 0 \rightarrow \text{stop}$$

Pseudocode

```
loop (n) == 0  
{  
  n = n % 10  
  n = n / 10  
}
```

to obtained individual digits.

Pseudocode (pending)

int ans = 0

while (n != 0) {

int digit = n % 10;

if ((ans > INT_MAX / 10) ||

code for reverse of a number [Integer]

```
int reverse(int x) {
```

```
    int ans = 0;
```

```
    while(x != 0) {
```

```
        int digit = x % 10;
```

```
        if ((ans > INT_MAX/10) || (ans > INT_MIN/10)) {
```

```
            return 0;
```

```
        }
```

```
        ans = (ans * 10) + digit;
```

```
        x = x / 10;
```

```
    }
```

```
    return ans;
```

```
}
```

```
};
```