**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*BITWISE OPERATOR\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**&-------------🡪2**

**|--------------🡪4**

**~-------------🡪1**

**^-------------🡪3**

Based on priority

Eg:- ~9+4&6----------🡪2

**XOR OPERATOR:**

In XOR calculation same number gives value 0 and different number gives value 1.

**Eg:- 10^5** **10 =1010**

**5 = 0101**

**1111**-----🡪15

**LEFT SHIFT OPERATION:**

The left shift operation (<<) is a bitwise operation used in many programming languages, including Python, to shift the bits of a binary number to the left by a specified number of positions. Here's how it works:

**Eg:- x = 7**

**result = x << 1 where 7 = 00000111**

**shifting 1 bit = 00001110 --------🡪14**

Here I have taken binary value of 7 as 8bit

**RIGHT SHIFT:**

**Eg:- x = 7**

**result = x >> 1 where 7 = 00000111**

**shifting 1 bit = 0000011 --------🡪3**

**BIT MANIPULATION TRICKS:**

XOR ^

Even 1’s : 0

Odd 1’s : 1

XOR of number itself is 0

XOR of number with 0 is number itself

**Eg:- 4^5^6**

**Ans:- 0100**

**0110**

**0101**

**0111---------🡪7**

\*\*How right shift works:

Eg:- 5>>1 5/2 ans 2

"How many times should those numbers be right-shifted to divide them by 2"

\*\*How left shift works:

Eg:- 5<<2 5\*(22) ans 20

"How many shifts are required for those numbers to become powers of 2"

Program question: After creating an array find out the smallest missing positive integer?

def missing\_number(arr):

num\_set = set(arr)

for i in range(0, max(arr) + 2):

if i not in num\_set:

return i

input\_str = input("Enter the list of array: ").split()

arr = [int(x) for x in input\_str]

result = missing\_number(arr)

print(result)

TASK:

1. Give an array every number occurs twice only one number occurs once find the number which is occurs once.

Ans:-

def find\_single\_number(nums):

result = 0

for num in nums:

result ^= num

return result

arr = [2,3,5,4,5,3,2,4,88]

new\_arr = find\_single\_number(arr)

print(new\_arr)

Q. Swap 2 number using xor?

a= 100

b= 200

a=a^b

b=a^b

a=a^b

print("a =", a,"b =", b)

Q. Find the kth bit is set or not?

n = 10

k = 3

result = n&(1<<(k-1))

if result != 0:

print(f"The {k}th bit is set.")

else:

print(f"The {k}th bit is not set.")

Q. Give n print the xor of all numbers

n = int(input("Enter your value: "))

xor = 0

for i in range(1,n+1):

xor = xor^i

print(xor)

0(n)-------🡪 time complexity

**#optimised**

n=int(input("Enter your value: "))

xor=0

if n%4==0:

print(n)

elif n%4==1:

print(1)

elif n%4==2:

print(n+1)

elif n%4==3:

print(0)

O(1)---------🡪 time complexity

Q.Find out Even or Odd using bitwise operator

n =int(input("Enter a value: "))

if (n&1==0):

print("Even")

else:

print("Odd")

Q. Find out the Fibonacci value of given number

def fib(n):

if n <= 0:

return 0

elif n == 1:

return 1

else:

return fib(n-1) + fib(n-2)

n = int(input("Enter the value you want fibonacci value : "))

print(fib(n))