29 Sptember,2023

Bitwise Operators:

In Python, “/” gives float where as “//” gives int.

Eg: (10\*4/6+3-1%2)

(40/6+3-1%2)

(6.66+3-1%2)

(6.66+3-1)

(9.66-1)

8.66

Priority is \*, / ,%, +,-

|  |  |  |  |
| --- | --- | --- | --- |
|  | C | Python | Java |
| Bitwise and | & | and | & |
| Bitwise or | | | or | | |
| Bitwise Negotiation | ~ (tilt) | ~ | ~ |

\*Priority of Bitwise and Arithemetic Operators is Arithemetic takes first.

\*Priority of Bitwise is ~, &, ^, |.

\*~3= -(3+1) = -4

\*~7 = -(7+1) =-8

\*XOR concept is important bcz same numbers give zero and different numbers give one.

\*XOR- 2, inclusive XOR (OR), exclusive XOR(XOR).

\*Right shift(left to right) and Left shift(right to left) Operators.

\*right shift of 7 is 14.

Bit Manipulation Tricks:

\*Rule1: xor of number itself is 0(number with number xor is 0).

\*xor of number with 0 is number itself.

\*Rule2: even 1’s:0

\*odd 1’s:1

\*Rule3:5>>1 🡪5/2=2 (1 time)

5>>2 🡪5/2=2 🡪2/2=1(2 time)

14>>4 🡪14/2=7 🡪7/2=3 🡪3/2=1 🡪1/2=0(4 time)

6>>3 🡪6/2=3 🡪3/2=1 🡪1/2=0(3 time)

Applicable for only right shift

\*Rule4:5<<2 🡪5\*(2^2)=20

10<<3 🡪10\*(2^3)=80

9<<4 🡪9\*(2^4)=144

Applicable for only left shift

Q: After creating an array, find out the smallest missing positive integer.

Q: Array [1,5,1,2,3,2,3] every integer occurs twice, one integer occurs once. Print that num.

Q: For given num n, check the kth bit is set or not. set means 1, otherwise 0.

Eg: n=10, k=3

1 0 1 0

K= 4 3 2 1 🡪3rd bit is 0 so not set

\*Formula is n&(1<<k-1), if answer is 0 then it is not set otherwise set.

\*10 & (1<<2) 🡪10&4=0 (not set).

Q: For the given num n find out xor of all n numbers.

|  |  |
| --- | --- |
| n | Output |
| 1 | 1 |
| 2 | 3 |
| 3 | 0 |
| 4 | 4 |
| 5 | 1 |
| 6 | 7 |
| 7 | 0 |
| 8 | 8 |
| 9 | 1 |
| 10 | 11 |
| 11 | 0 |
| 12 | 12 |
| 13 | 1 |

Q: For the given num n find out xor of all numbers for the given range.

Eg: range is 2 to 4 ans:5

xor (1 to 4) ^ xor (1)

(1^2^3^4) ^ (1)= 4 ^ 1 =5

Q:checking num even or odd using bitwise.

\*bitwise and of any even number is 0 and odd number is 1.

\*last bit of even is 0, odd is 1.

Q:Finding n th term in Fibonacci series