Problem 1: given a number, print its binary representation. [easy]

Input 1: number = 5

Output 1: 101

ANS. 2 | 5

—----

2 | 2 1

—----

2 | 1 0

Reading numbers is reverse order gives us 101 as the binary representation

Input 2: number = 10

Output 2: 1010

ANS . 2 | 10

—-----

2 | 5 0

—-----

2 | 2 1

—-----

2 | 1 0

Reading in reverse order from bottom to top gives us 1010 as the binary representation of 10.

Problem 2: given a number ‘n’, predict whether it is a power of two or not. [medium]

Input 1: n = 15

. Converting it in binary 1111, as its more than 1 set bits in the binary representation therefore it’s not power of 2.

Output 1: False

Input 2: n = 32

Converting it to binary 10000, as it has only 1 set bit, therefore its power of 2.

Output 2: True

Ans. https://github.com/PRAVALSHARMA/PW\_JAVA\_DSA\_ASSIGNMENT.git

Q3. Problem 1: Given a number. Using bit manipulation, check whether it is odd or even.

Input 8, Even

3, False

Ans. https://github.com/PRAVALSHARMA/PW\_JAVA\_DSA\_ASSIGNMENT.git

Q4. Given a number, count the number of set bits in that number without using an extra space.

Note : bit ‘1’ is also known as set bit.

Ans. https://github.com/PRAVALSHARMA/PW\_JAVA\_DSA\_ASSIGNMENT.git

Q5. Given an integer array, duplicates are present in it in a way that all duplicates appear an even number of times except one which appears an odd number of times. Find that odd appearing element in linear time and without using any extra memory.

For example,

Input : arr[] = [4, 3, 6, 2, 6, 4, 2, 3, 4, 3, 3]

Output : The odd occurring element is 4.

Ans. https://github.com/PRAVALSHARMA/PW\_JAVA\_DSA\_ASSIGNMENT.git