DSA SERIES

- Learn Coding

Topic to be Covered today

Bit Manipulation

LETS START TODAY'S LECTURE

XOR OPERATION

1720. Decode XORed Array

```
class Solution {
public:
    vector<int> decode(vector<int>& encoded, int first) {
        int n = encoded.size();
        vector<int> arr(n+1);

        arr[0]=first;

        for(int i = 1;i<n+1;i++){
            arr[i]=arr[i-1] ^ encoded[i-1];
        }

        return arr;
    }
};
```

136. Single Number

```
class Solution {
public:
    int singleNumber(vector<int>& nums) {
        int result = 0;

        for(int num :nums){
            result = result ^ num;
        }

        return result;
    }
};
```

260. Single Number III

```
class Solution {
public:
    vector<int> singleNumber(vector<int>& nums) {
        long long XOR = 0;
        for(int num : nums){
            XOR ^= num;
        }
        int mask = XOR & (-XOR);
        int groupA = 0;
        int groupB = 0;

        for(int num :nums){
            if(num&mask){
                groupA ^= num;
            } else{
                groupB ^= num;
            }
        }
        return {groupA ,groupB};
    }
};
```

371. Sum of Two Integers

```
class Solution {
public:
    int getSum(int a, int b) {
       while(b !=0){
        int carry = (a&b)<<1;

        a = a^b;
        b = carry;
       }
       return a;
    }
};
```

268. Missing Number

```
class Solution {
public:
    int missingNumber(vector<int>& nums) {
        int n = nums.size();

        int result = n;
        for(int i =0;i<n;i++){
            result = result ^ i ^ nums[i];
        }

        return result;
    }
};
```

1310. XOR Queries of a Subarray

```
class Solution {
public:
    vector<int> xorQueries(vector<int>& arr, vector<vector<int>>& queries) {

        int n = arr.size();
        vector<int> cumXor(n);

        cumXor[0] = arr[0];

        for (int i = 1; i < n; i++) {
            cumXor[i] = cumXor[i - 1] ^ arr[i];
        }

        vector<int> result;

        for (vector<int> query : queries) {
            int L = query[0];
            int R = query[1];
```

```
            int temp = cumXor[R] ^ (L == 0 ? 0 : cumXor[L - 1]);

            result.push_back(temp);
        }
        return result;
    }
};
```

Learn coding

THANK YOU