Count Total Setbits

```
int countBits(int n) {
        int count=0;
        for(int i=1;i<=n;i++){
            for(int j=0;j<=31;j++){
                int mask=(1<<j);
                if((i&mask)!=0){
                    count++;
                }
            }
        }
        return count;
    }
```

Time Complexity:31*n
Space:Constant

Missing number in array

```
class Solution {
  public:

    // Note that the size of the array is n-1
    int missingNumber(int n, vector<int>& arr) {

        int xor1=0;
        for(int i=0;i<n-1;i++){
            xor1=xor1^arr[i];
            xor1=xor1^i+1;
        }

    //    for(int i=0;i<arr.size();i++){
    //        xor1=xor1^arr[i];
    //    }
        return xor1^n;
    }
};
```

Time Complexity:O(n)
Space:Constant

Check whether K-th bit is set or not

```
class Solution
{
    public:
    // Function to check if Kth bit is set or not.
```

```
    bool checkKthBit(int n, int k)
    {
        int mask=(1<<k);
        if((n&mask)!=0){
            return true;
        }
        else{
            return false;
        }
    }
};
```

Time Complexity:Constant
Space:Constant

Number of 1 Bits

```
class Solution {
  public:
    int setBits(int N) {
        // Write Your Code here
        int count=0;
        while(N!=0){
            if(N&1==1){
                count++;
            }
            N=N>>1;
        }
        return count;
    }
};
```

Time Complexity:O(MSB)
Space:Constant

Single Number II

```
class Solution {
public:
    int singleNumber(vector<int>& nums) {
         long long int singleNumber=0;
        for(int i=0;i<=31;i++){
            int count=0;
            for(int j=0;j<nums.size();j++){
                if((nums[j]&(1<<i))){
                    count++;
```

```
                }
            }
            if(count%3==0){
                //do nothing
            }
            else{
                singleNumber+=pow(2,i);
            }
        }
        return (int)singleNumber;
    }
};
```

Time Complexity:O(31*n)
Space:Constant