DSA SERIES

- Learn Coding

Topic to be Covered today

Recursion

LETS START TODAY'S LECTURE

231. Power of Two

```
class Solution {
public:

    bool isPowerOfTwo(int n) {

        if (n == 1) {
            return true;
        }

        if (n == 0 || n % 2 != 0)
            return false;
        return isPowerOfTwo(n / 2);
    }
};
```

326. Power of Three

```
class Solution {
public:
    bool isPowerOfThree(int n) {
        if(n == 1) return true;

        if(n<=0 || n%3 !=0){
            return false;
        }

        return isPowerOfThree(n/3);
    }
};
```

342. Power of Four

```
class Solution {
public:
    bool isPowerOfFour(int n) {
          if(n == 1) return true;

        if(n<=0 || n%4 !=0){
            return false;
        }

        return isPowerOfFour(n/4);
    }
};
```

509. Fibonacci Number

```
class Solution {
public:
    int fib(int n) {
        //  base case
        if(n==0) return 0;
        if(n==1) return 1;

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

    }
};
```

50. Pow(x, n)

```
class Solution {
public:
    double solve(double x , long n){
        if(n==0) return 1;

        if(n<0){
            return  solve(1/x,-n);
        }

        if(n%2 == 0){
            return solve(x*x ,n/2);
        }

        return x * solve(x*x,n/2);
    }
    double myPow(double x, int n) {
        return solve( x , (long)n);
    }
};
```

1922. Count Good Numbers

```
class Solution {
public:
int M = 1e9+7;

long long findPower(long long a ,long long b){
    if(b == 0){
        return 1;
    }

    long long half = findPower(a,b/2);

    long long result =( half * half) % M;

    if(b%2 == 1){
        result = (result * a) %M;
    }
    return result;
}
```

```
  int countGoodNumbers(long long n) {
        long long  even = (n+1)/2;
        long long  odd = n/2;

        return (long long)(findPower(5,even) * findPower(4,odd)) % M;
    }
};
```

1823. Find the Winner of the Circular Game

Using Array

```
class Solution {
public:
    int findTheWinner(int n, int k) {
        vector<int> arr;

        for(int i =1 ;i<=n;i++){
            arr.push_back(i);
        }

        int i =0;
        while(arr.size()>1){
            int idx = (i + k-1)% arr.size();

            arr.erase(arr.begin()+idx);

            i = idx;
        }

        return arr[0];
    }
};
```

1823. Find the Winner of the Circular Game

Using Queue

```
class Solution {
public:
    int findTheWinner(int n, int k) {
      queue<int> que;
      for(int i =1;i<=n;i++){
        que.push(i);
      }

    while(que.size()>1){
        for(int i = 1 ; i<=k-1;i++){
            que.push(que.front());
            que.pop();
        }
        que.pop();
    }

    return que.front();

    }
};
```

1823. Find the Winner of the Circular Game

Using Recursion

```
class Solution {
public:
    int solve(int n, int k) {

        if (n == 1)
            return 0;

        int idx = solve(n - 1, k);
        idx = (idx + k) % n;

        return idx;
    }

    int findTheWinner(int n, int k) {
        int result = solve(n, k);
        return result + 1;
    }
};
```

Learn coding

THANK YOU