* Problem statement :-

→ Calculate the n^{th} fibonacci number.

→ $F(n) = F(n-1) + F(n-2)$ for $n>1$ where $F(0)=0$ & $F(1)=1$

* Test Cases :-

Input :- n = 2

Output :- 1

* Solution :-

* Approach 1 :- Basic recursive approach

→ Base case is that if n==0 then return 0 & for n==1, return 1.

→ $F(n) = F(n-1) + F(n-2)$ which will be our final return.

```
func fibo (n int) {
    if n==0 {
        return 0
    }
    if n==1 {
        return 1
    }
    return fibo (n-1) + fibo (n-2)
}
```