Recursion

(1) - End?
→ ↳ Condition (stop)?
* Base Condition
↗ → Solution Available (known).

```
void func( )
{
    Cout<<" A function"<<endl;
    func();
}

main
func();
```

```
int factorial (int num)
{
    if (num ==0 || num ==1) // Base Case
        return 1;
    else
        return num*factorial(num-1);
}
```

factorial(4)

```
factorial(4)   num=4   24
return 4*factorial(3)
                   2 =6
 num=3
return 3* factorial(2) num=2
                           1    = 2
                return 2* factorial(1)
```

Table of n? Solution? n X 1 = n

2 X 1	= 2
2 X 2	= 4
2 X 3	= 6
2 X 4	= 8
2 X 5	= 10

```
void tableOf(int val, int start, int till)
{
    if (start>till) // Base Condition
                       stoping
        return;
    else
tableOf(2,1,5) {
        Cout<< val << " * " <<start << " = " << val * start;
        Cout<<endl;
        tableOf(val, start+1, till);
    }
}
```

Golden Spiral
& fibonacci sequence.

Fibanacci Sequences

0 1 1 2 3 ----

fib(0) fib(1) fib(2) fib(3) fib(4) fib(15)

fib(n) = fib(n-1) + fib(n-2) ✓

fib(3)
fib(2)+fib(1)
fib(1)+fib(0)

```
int fib(int n)
{
  if (n==0 || n==1) // Base case
       return n;
  else
       return fib(n-1) + fib(n-2);
}
```

$$\overset{3}{\text{fib}(4)} \rightarrow \text{return } \overset{2}{\text{fib}(3)} \overset{+}{+} \overset{1}{\text{fib}(2)} = 3 \rightarrow \text{return } \overset{1}{\text{fib}(1)} \overset{+}{+} \overset{0}{\text{fib}(0)} = 1$$

$$\text{return } \overset{1}{\text{fib}(2)} \overset{+}{+} \overset{1}{\text{fib}(1)} = 2$$

$$\text{return } \overset{1}{\text{fib}(1)} \overset{+}{+} \overset{0}{\text{fib}(0)} = 1$$

Slow ← Recursive methods