Recursion

- Recursion → Function calling itself.

knock, knok
Who's there ?
shailja
shailja Who ?
knock, knock

★ How to know, Rcursion is applied in question?
⇒ When question can break into sub-parts.

★ Important part of recursion question ?

1) Base Condition :- Condition where recursion will stop making calls. OR program will be terminate.

2) Body :-

★ Why I use Recursion ? (benifits mangta apun...)

1) It helps us in solving bigger/complex problems into ~~smaller~~ simple way.
2) We can convert recursion solution into interation & vice versa.
3) Space complexity is not constant because of recursive calls.
4) It helps in breaking down bigger problems into smaller problems.

* Visualising Recursion :- (Proof kidhar h bhai...)

Let's take example →

Print 1 to n numbers using Recursion.

Code ⇒

```
int main()
{
    int n;
    cin >> n;
    print(n);   // ~~5~~ 1
}

void print(int n)
{ ~~print(n-1);~~
  ~~cout << n;~~
  if(n==5)
        cout << n;
     return ;
         → cout << n;
  print(n+1);
}
```

1 to 5

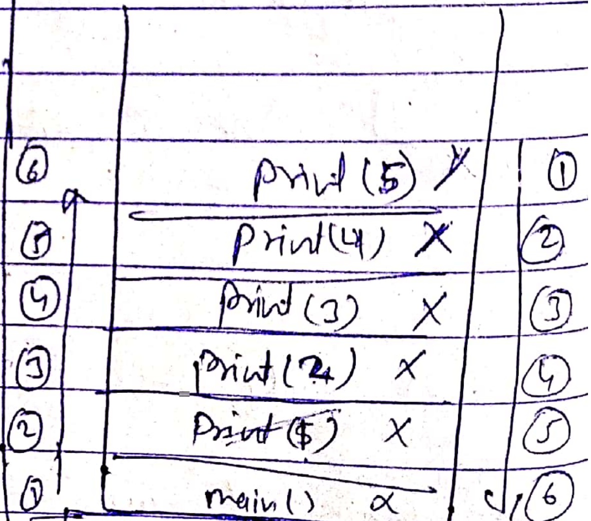
1 2 3 4 5

(n-4), (n-3), (n-2), (n-1), n

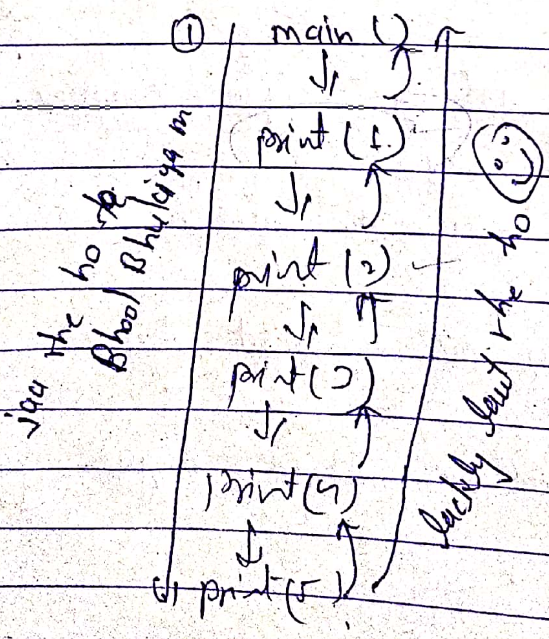
i) first print(5)

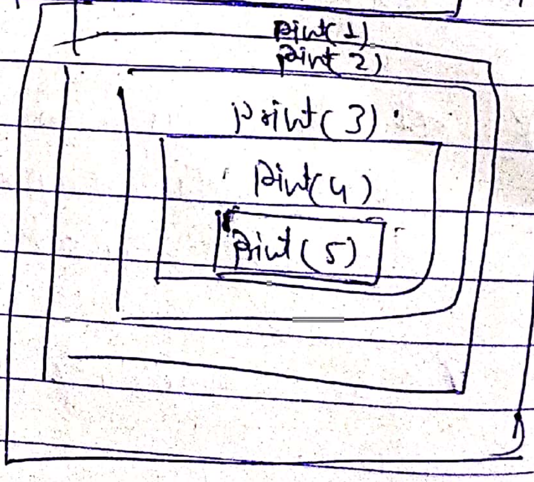
O/P:- 1

Stack!-



Tree :-

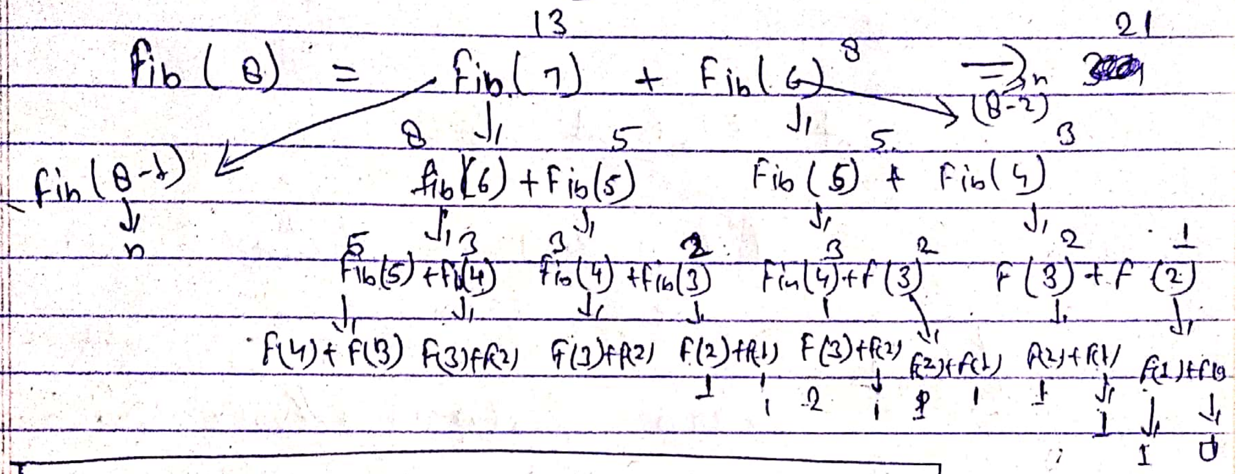




Hey, congratulations ☺ !
you completed 1st module
let's go further →

Example 2 ⇒ Find nth fibonacie number.

0th	1th	2nd	3rd	4th	5th	6th	7th	8th
0	1	1	2	3	5	8	13	21 .....



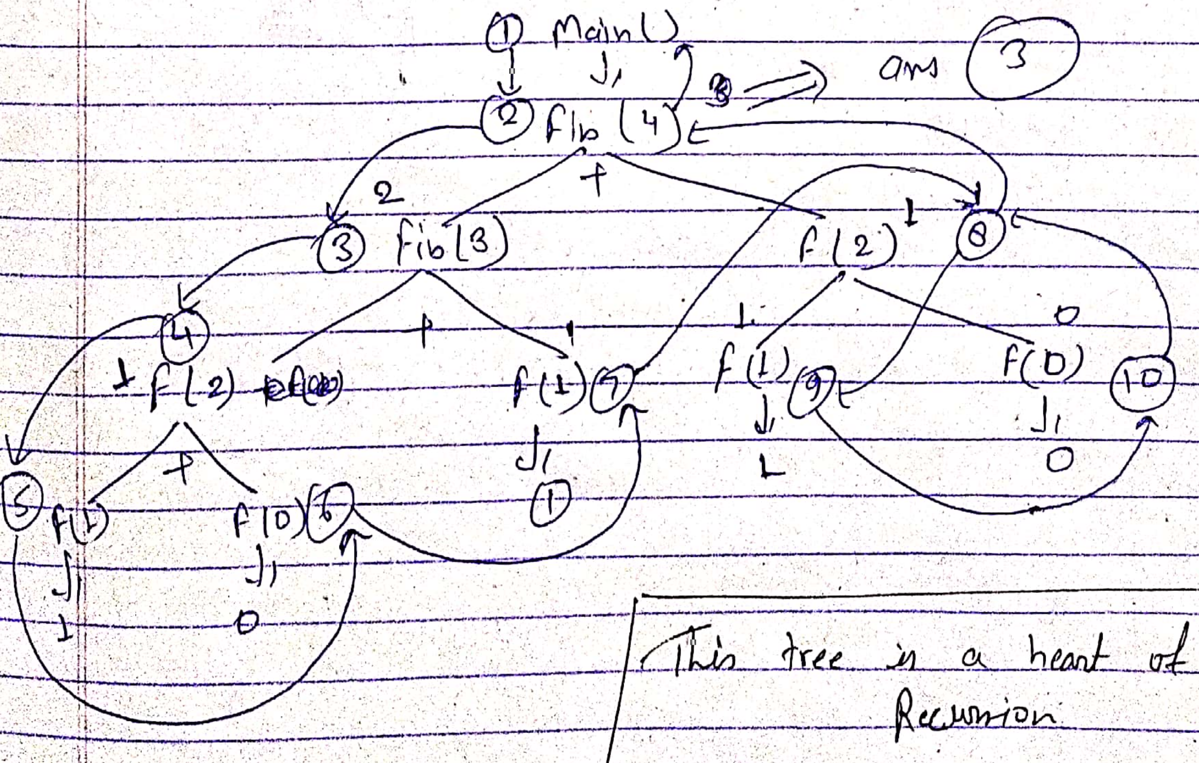
13
21
fib(8) = fib(7) + fib(6) 8
⇒ n
(8-2)
fib(8-1)
n
8 5
fib(6) + fib(5)
5 3
Fib(5) + Fib(4)
5 3 3 2
Fib(5) + Fib(4) fib(4) + fib(3)
3 2 2 1
Fib(4) + f(3) F(3) + F(2)
F(4) + F(3) F(3) + F(2) F(3) + F(2) F(2) + F(1) F(3) + F(2) F(2) + F(1) F(2) + F(1) F(1) + F(0)
1 1 2 1 1 1 1 1 1 0

$$\boxed{fib(n) = fib(n-1) + fib(n-2)}$$

dhyan do ↗ This is called as a ~~Recursion~~ Recurence Relation

Visualization

4th Fibonacie Number



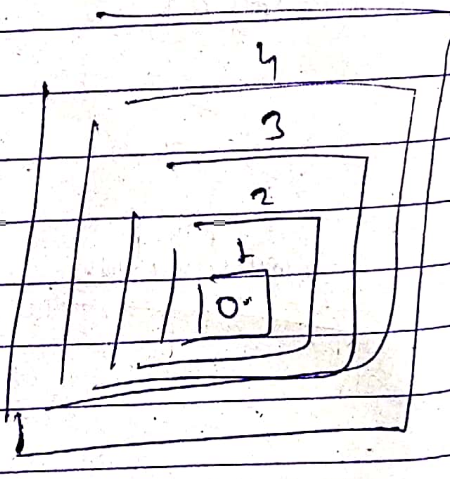
This tree is a heart of Recursion

(n = = 0 || n = = 1)

Code:-

```
int  main ()
{
     int  n = 4  ;
     fib (n4) ;
 }

 int  fib ( int  n)
{
     if ( n <2)          // Basic Condition
          return  n ;
// body
     return  fib (n-1) + fib (n-2) ;
                  1            0
 }
```



☆☆ Most Important Part, if u want to masters in Recursion

☆ How to approach Recursion Problem ?

Step1 ⇒ Identify if you can break down problem into smaller problems.

Step2 ⇒ Write the Recurrance Relation, if needed.

Step 3 ⇒ Draw the recursive tree.

Step4 ⇒ About the tree →

(a) See the flow of function, how they are getting solve.

(b) Identify flows of left and right tree calls.

(c) Draw the tree and pointer again and again using Pen & Paper.

(d) Use a debugger to see the flow.

Step5 ⇒ See how the values and what type of values (int, string etc) are retrived at each step. See where two function call will come out. In the end, you will come out the main function.

> Do Practice and have Patience. ☺
> you will definitly achieve target.