11 Feb

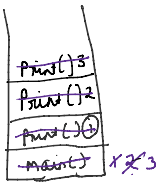
11 February 2022 20:53

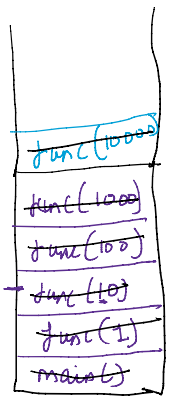
## Recursion:-

↳ func. calling itself.

```
main() {
    1 - Print("H");
    2 - Print("I");
    3 - Print("Z");
}
Print(s) {
    Sout(s);
}
```

H
I
Z




```
func(n) {
    if(n==10000) return;
    func(n*10)
}
```

```
main() {
    func(1);
}
```

10

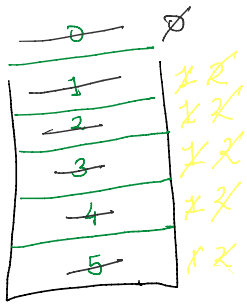
n x 1 = n

n x 2 = (n x 1) + n

n x 3 = (n x 2) + n

n = 5

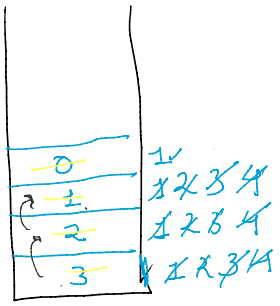
5
4
3
2
1




```
Void Print(n) {
    0- if(n==0) return;
    1 - Sout(n)
    2 - Print(n-1);
}
```

n = 3

3
2
1
1
2
3




```
Print(n) {
    1 if(n==0) return;
    2 Sout(n)        Pre-ORDER
    3 Print(n-1);
    4 Sout(n)        Post-ORDER
}
```

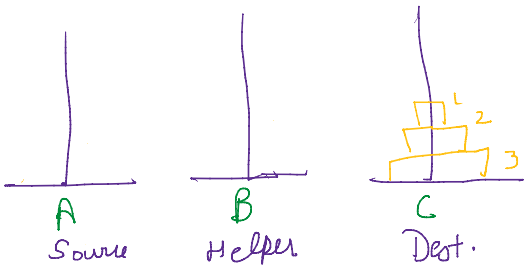
n=3 n=2 n=1

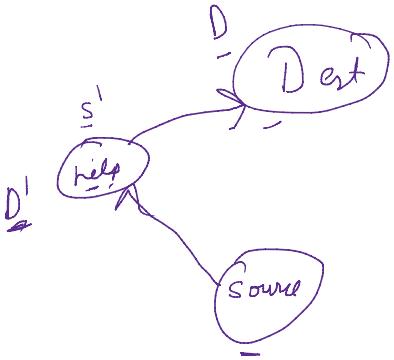

```
pattern(n) {
1   if(n==0) return;
2       Sout(n);
3       pattern(n-1);
4       Sout(n);
5       pattern(n-1);
6       Sout(n);
}
```

2
1
1
2
3

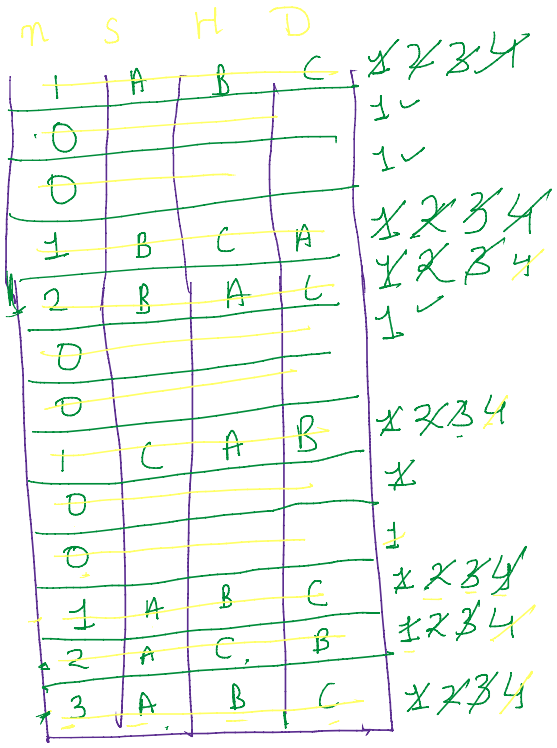
# Tower of Hanoi:-

n = no. of disks = 2




(1) Only one disk can be moved at a time.
(2) A disk is slid off the top of one tower onto another tower.
(3) A disk cannot be placed on top of a smaller disk. Write a program to move the disks from the first tower to the last using Stacks.

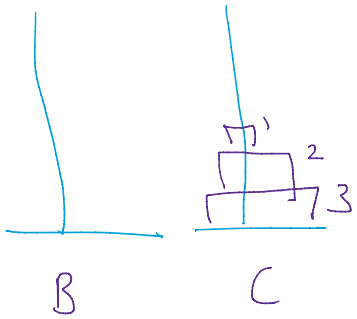



n = 4

3, A B C

```
1  public static void toh(int n, char source, char helper, char dest) {
2      if (n == 0) return;
3      toh( n: n - 1, source, dest, helper);
4      System.out.println("move " + n + " from " + source + " to " + dest);
       toh( n: n - 1, helper, source, dest);
   }
```

move 1 from A to C
move 2 from A to B
move 1 fro C to B
Move 3 from A to C
move 1 from B to A
Move 2 from B to C
move 1 from A to C


1
2
3
B
C