Hands-On-Activiy 4.1 Tower of Hanoi

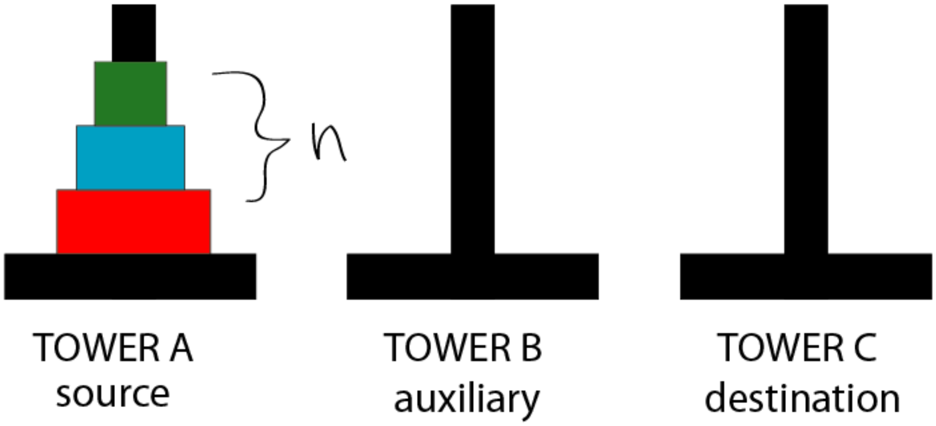
```
#Code

def Hanoi(n, tower1, tower2, tower3):
    if n == 1:
        print("Move Disk",n,"from Tower",tower1, "to", "Tower",tower3)

    else:
        Hanoi(n-1, tower1, tower3, tower2)
        print("Move Disk",n, "from Tower", tower1, "to Tower", tower3)
        Hanoi(n-1, tower2, tower1, tower3)

disk = int(input("Enter number of disks: "))
print( "--"*10, "SOLUTION", "--"*10)
Hanoi(disk,"A", "B","C")
```

```
Enter number of disks: 3
-------------------- SOLUTION --------------------
Move Disk 1 from Tower A to Tower C
Move Disk 2 from Tower A to Tower B
Move Disk 1 from Tower C to Tower B
Move Disk 3 from Tower A to Tower C
Move Disk 1 from Tower B to Tower A
Move Disk 2 from Tower B to Tower C
Move Disk 1 from Tower A to Tower C
```

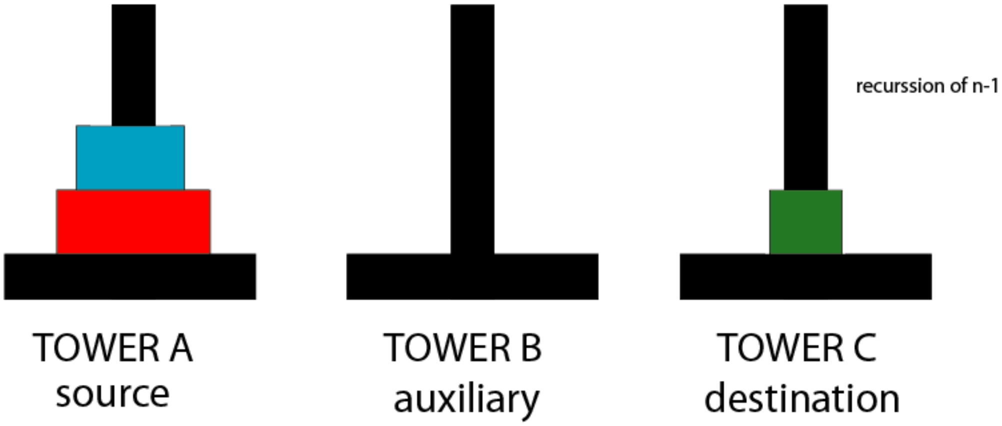
EXPLANATION

In the Hanoi Tower, we used recurssion to move the position of the disk to its destination. Also, while we are doing our recurssion, we are changing the source of the disk and its destination which means that not all the times the Tower A will be the source of the disk and also not all the times Tower C will be the destination.

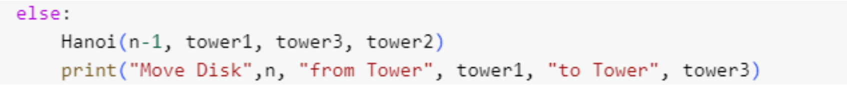
The code will ask firsts on how many disks would they want, in this case, we have 3 disks



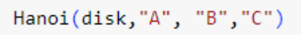
Since n is not equal to one, the recursion of the function will now occur until the our n = 1 so that we can remove the first disk.



After moving the disk 1 by recurssion, the postion of our source, destination, and the auxiliary of our tower will now change.



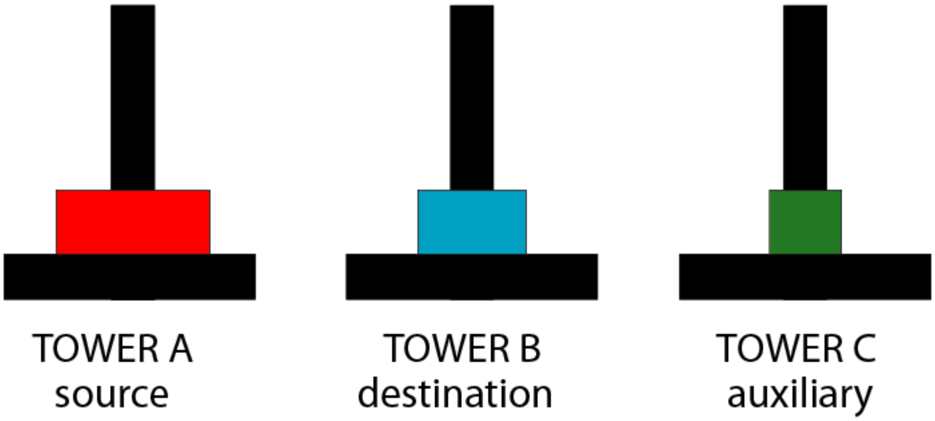
```
else:
    Hanoi(n-1, tower1, tower3, tower2)
    print("Move Disk",n, "from Tower", tower1, "to Tower", tower3)
```



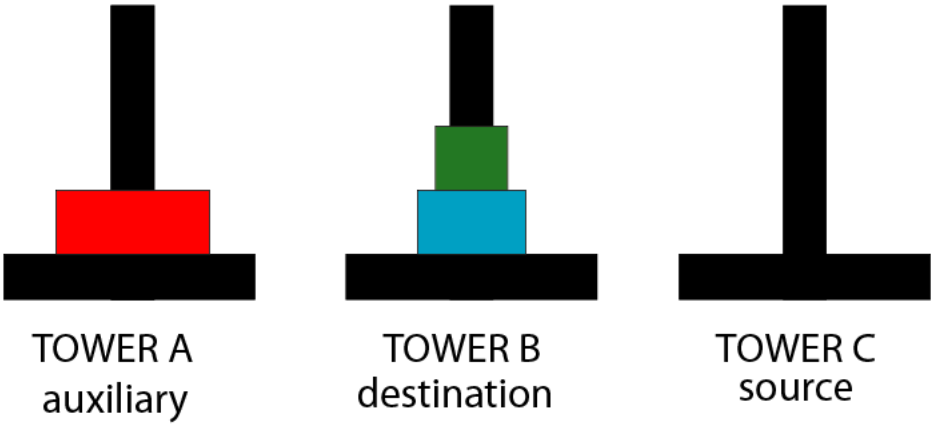
```
Hanoi(disk,"A", "B","C")
```

We are now going to move disk 2, the source is still Tower A but our destination will be Tower B, which is our auxiliary

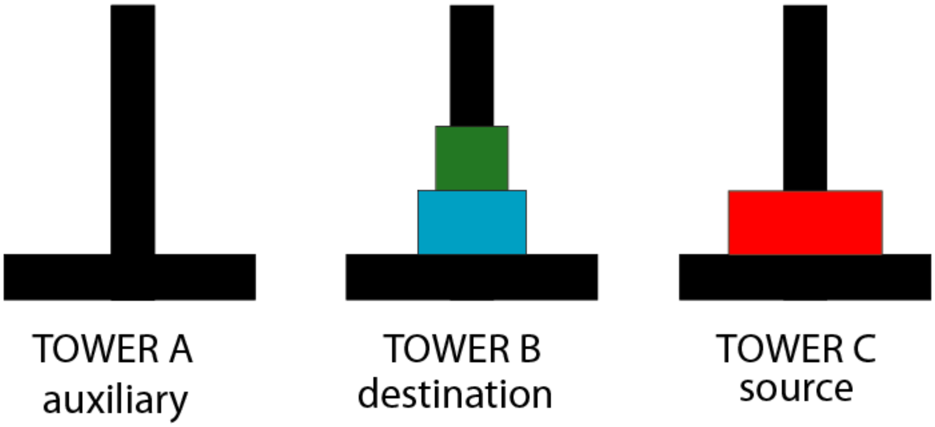
```
else:
    Hanoi(n-1, tower1, tower3, tower2)
    print("Move Disk",n, "from Tower", tower1, "to Tower", tower3)
```



After moving the disk 2, the recurssion will occur so that we can move the disk 1, but our source will be Tower C and our destination will be Tower B



The code will now move the disk3 from Towe A to Tower B. The movement of disk3 is not by recurssion because disk 3 is not an answer to our parameter of n-1.



After moving the disk 3, the recurssion will now occur. However, by changing our parameters inside our function, we can change the source, auxiliary and the destination of our code so that we can properly move them to their designated places.

```
def Hanoi(n, tower1, tower2, tower3):
    if n == 1:
        print("Move Disk",n,"from Tower",tower1, "to", "Tower",tower3)

    else:
        Hanoi(n-1, tower1, tower3, tower2)
        print("Move Disk",n, "from Tower", tower1, "to Tower", tower3)
        Hanoi(n-1, tower2, tower1, tower3)
```

The recurssion will continue until the first conditional statement happens again, which is disk 1 (disk n) is moved from source to destination.

```
if n == 1:
    print("Move Disk",n,"from Tower",tower1, "to", "Tower",tower3)
```

If you observe the output, the movement of the disks (n-1, and n-1 recurssions) is the same after moving the last disk (n), the only difference is where they are coming from and where they are going to.

BEFORE MOVING DISK 3

```
Move Disk 1 from Tower A to Tower C
Move Disk 2 from Tower A to Tower B
Move Disk 1 from Tower C to Tower B
```

MOVING DISK 3 (disk n)

```
Move Disk 3 from Tower A to Tower C
```