Tail Recursion

- If a recursive function is calling itself and that recursive call is the last statement in a function then it is called as tail recursion.
- After that call it will nor perform any thing.
- All the function will be performing on the calling time itself
- If there is some function that need to be performed after its returning time then it is not a tail function

Example :

```
void fun(int n)
{
    if(n>0)
    {
        printf("%d", n);

        fun(n-1);
    }
}
fun(3);
```

Tail Recursion v/s loops:

- Tail recursion can easily converted into loops as its structure and syntax is almost same
- In term of time taken by both is same 0(n)
- Space taken by tail is 0(n) where as the space for loops is 0(1)
- To conclude , if you are using tail recursions its better to convert it into loop as the space used is less