Tail Recursion

The tail recursion is basically using the recursive function as the last statement of the function. So when nothing is left to do after coming back from the recursive call, that is called **tail recursion**.Their is one example for understanding the tail recursion concept as follows:

```
#include<iostream>
using namespace std;
int fact(int n){
    if(n<=1)
        return 1;
    return n*fact(n-1);
}
int main()
{
        cout << fact(5);
        return 0;
}
```

In the above function to calculate factorial of n.The function fact is recursively called at the end of the function. Until the factorial of number is not calculated the function fact is recursively called.Above example is type of tail recursion. As there is no task left after the recursive call, it will be easier for the compiler to optimize the code. When one function is called, its address is stored inside the stack. So if it is tail recursion, then storing addresses into stack is not needed.