# Recursive Function Assignment

1. WAP to calculate the maximum stack depth of a recursive call to a function. (For eg a factorial function).

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

1. What is tail recursion? Why is it important? Give an example

Ans: **Tail recursion** is a special form of recursion where the recursive call is the last operation performed in the function. In other words, the function returns the result of the recursive call directly, without any additional computation after the call returns.

**Importance:**

 **Space Efficiency**: Tail recursion avoids the overhead of maintaining multiple stack frames, as the function call doesn't need to retain any information once the recursive call is made.

 **Prevention of Stack Overflow**: Deep recursion doesn't lead to stack overflow errors in the case of tail recursion because the recursion depth is reduced, and stack frames are reused.

 **Improved Performance**: If the compiler supports tail call optimization, the performance of tail-recursive functions can be better than traditional recursion due to lower memory usage.

Example:

