# Recursive Function Assignment

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

A screen shot of a computer program

Description automatically generated

Output:

A black background with white letters and numbers

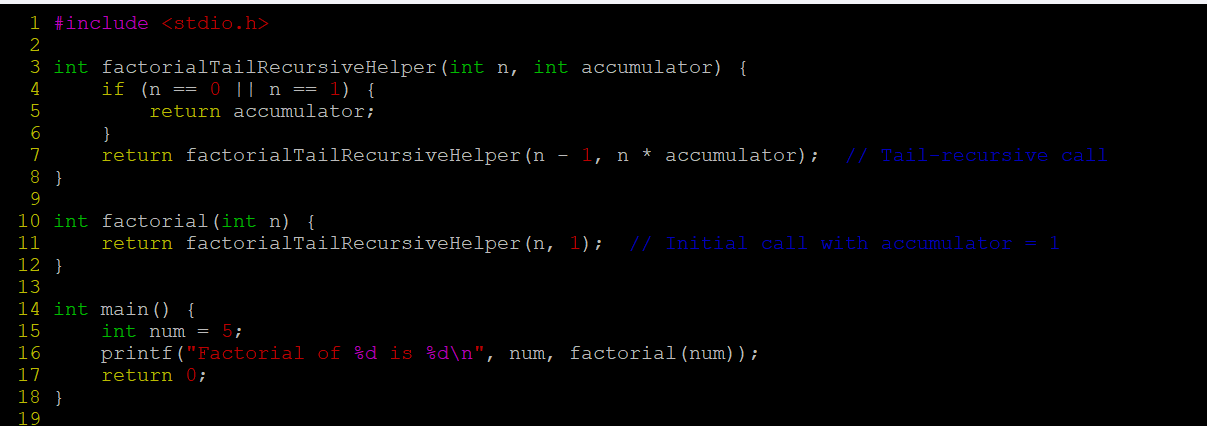
Description automatically generated

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

Ans) Tail recursion is a special case of recursion in which the recursive call is the **last operation** in the function. This means that there is **no further computation** to be done after the recursive call returns. In tail recursion, the current function call doesn't need to maintain its state (such as local variables) because the recursive call will return the final result.

**Importance of Tail Recursion**

* **Efficiency in Space**: Tail recursion allows the function to avoid using additional stack frames for each recursive call. This means it can execute more efficiently, especially for deep recursive calls.
* **Optimized Execution**: Many modern compilers or interpreters can optimize tail-recursive functions to use constant space and avoid the overhead of recursive calls.



Output:

A black background with white text

Description automatically generated