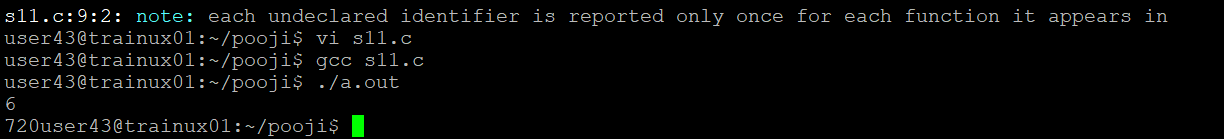
Recursive Function Assignment

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



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

Tail recursion is important because it allows a recursive function to be optimized by the compiler or runtime system to **avoid adding extra frames to the call stack**. This is particularly useful when dealing with deep recursion, where traditional recursion (non-tail recursion) could lead to a stack overflow or excessive memory usage.

**Example :**

#include <stdio.h>

int factorial\_helper(int n, int accumulator) {

if (n == 0) {

return accumulator; // Base case

}

return factorial\_helper(n - 1, n \* accumulator); // Tail call

}

int factorial(int n) {

return factorial\_helper(n, 1);

}

int main() {

int number = 5;

printf("Factorial of %d is: %d\n", number, factorial(number));

return 0;

}