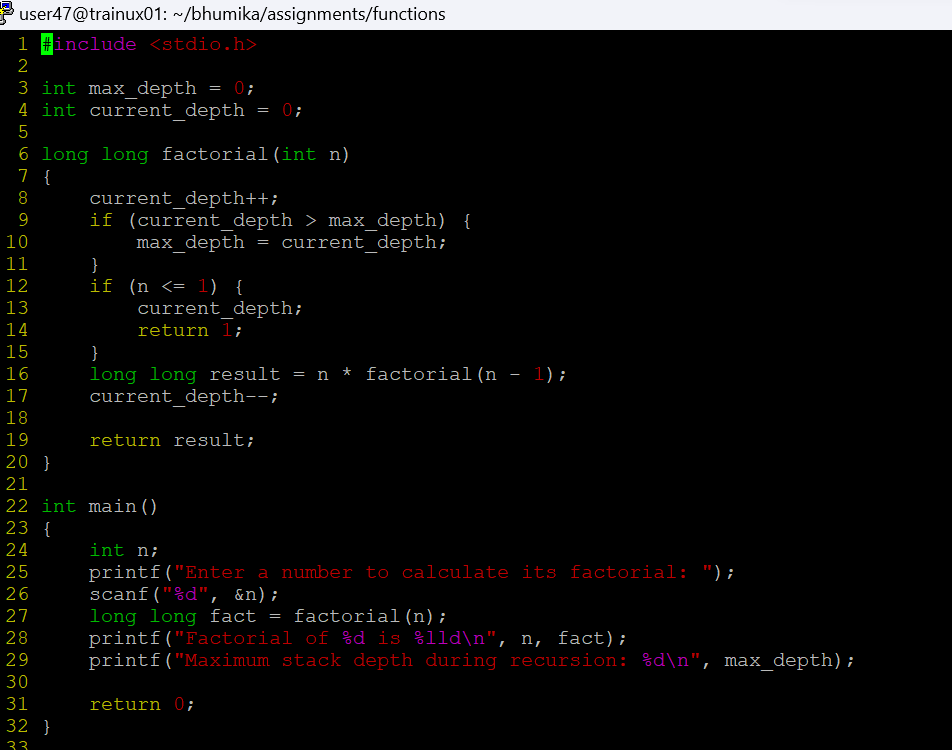
Recursive Function Assignment

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

Description automatically generated

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

**Tail Recursion** is a type of recursion where the recursive call is the last operation performed in the function. This means that the function returns the result of the recursive call directly, without any further computation after that call. Tail recursion is important because it can be optimized by the compiler or interpreter to avoid increasing the call stack depth, which can lead to more efficient memory usage and prevent stack overflow errors.

**Importance of Tail Recursion:**

1. **Memory Efficiency**: Tail recursion can reduce the memory footprint because it allows the reuse of the current function's stack frame for the next function call.
2. **Performance**: In languages that support tail call optimization (TCO), tail-recursive functions can run in constant stack space, which enhances performance.
3. **Preventing Stack Overflow**: Tail recursion helps avoid stack overflow errors in cases of deep recursion.

**Example of Tail Recursion in C:**

Here's an example of a tail-recursive function in C that calculates the factorial of a number.

