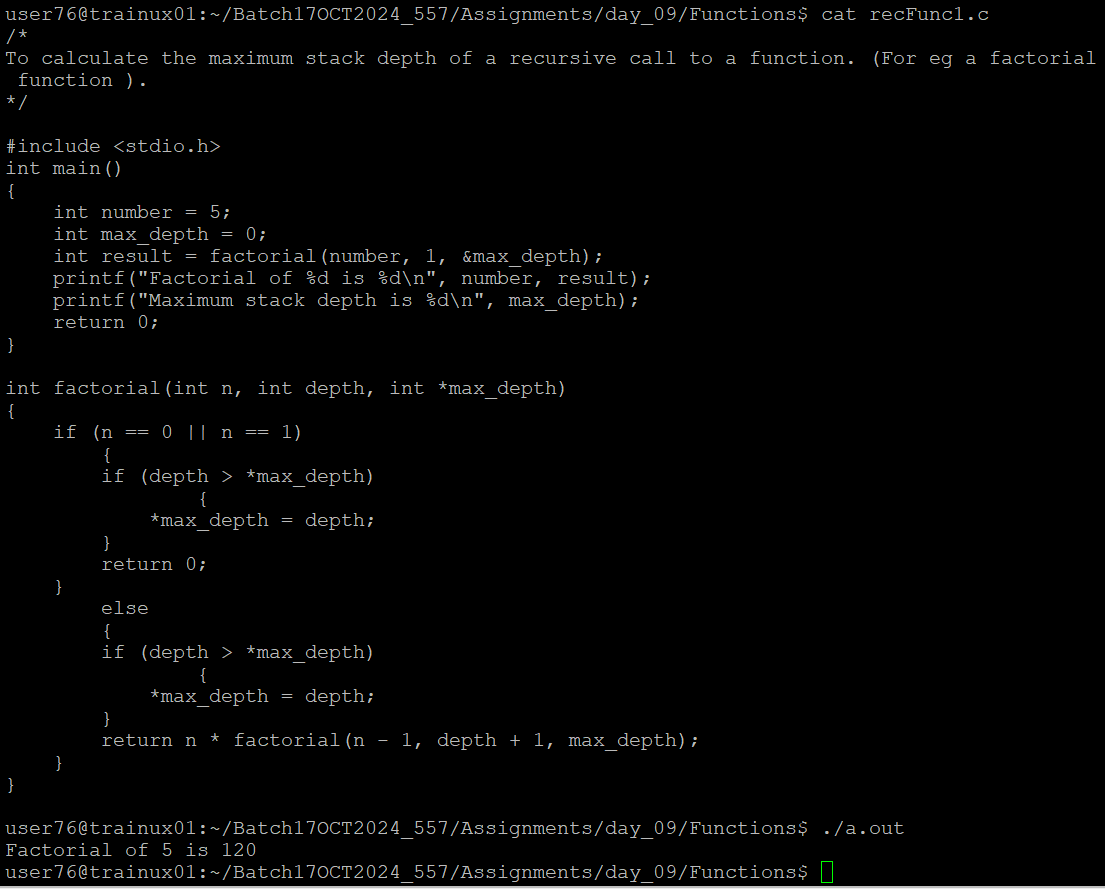
**Recursive Function Assignment**

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

****

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

Tail recursion occurs when a function calls itself as its last operation, meaning there are no further computations after the recursive call.

**Importance of Tail Recursion**

1. **Optimization**: Tail recursion allows compilers and interpreters to optimize recursive functions. When a function is tail-recursive, the compiler can transform the recursion into a loop, which reduces the overhead of function calls and improves performance.
2. **Memory Efficiency**: In traditional recursion, each function call adds a new frame to the call stack, which can lead to stack overflow if the recursion is too deep. Tail recursion mitigates this issue by reusing the same stack frame for each recursive call, thus conserving memory and preventing stack overflow errors.
3. **Improved Performance**: By converting tail-recursive calls into iterative loops, the execution time can be reduced. This is particularly beneficial in functional programming languages where recursion is a common pattern.

A computer screen with white text

Description automatically generated