Recursive Function Assignment

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

user60@trainux01: ~/Batch17OCT2024_175/Assignments/Day07/Recursive_Function_Assignment

```c
#include <stdio.h>

int max_depth = 0;
int current_depth = 0;
int factorial(int n) {
    int result;
    current_depth++;
    if (current_depth > max_depth) {
        max_depth = current_depth;
    }
    if (n <= 1) {
        current_depth--;
        return 1;
    }
    result = n * factorial(n - 1);
    current_depth--;
    return result;
}
int main() {
    int number,result;
    printf("Enter a number to calculate its factorial: ");
    scanf("%d", &number);
    result = factorial(number);
    printf("Factorial of %d is %d\n", number, result);
    printf("Maximum stack depth: %d\n", max_depth);
    return 0;
}

~
```

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

Tail Recursion is a special kind of recursion where the recursive call is the last operation in the function. In other words, there is no additional computation or operation after the recursive call returns. This makes it easier for the compiler or the runtime system to optimize the recursion, often converting it into an iterative loop.

Why is Tail Recursion Important?

- **Optimization**: Tail recursion can be optimized by the compiler to avoid consuming additional stack frames, thereby preventing stack overflow and reducing the overhead of recursion. This optimization is called tail call optimization (TCO).
- **Efficiency**: With tail recursion, there is no need to keep track of the previous state of the function, which makes it more memory efficient compared to non-tail recursion.

user60@trainux01: ~/Batch17OCT2024_175/Assignments/Day07/Recursive_Function_Assignment

```c
#include <stdio.h>

int factorial_tail_recursive(int n, int accumulator) {
    if (n <= 1) {
        return accumulator;
    }
    return factorial_tail_recursive(n - 1, n * accumulator);
}
int main() {
    int number,result;
    printf("Enter a number to calculate its factorial: ");
    scanf("%d", &number);
    result = factorial_tail_recursive(number, 1);
    printf("Factorial of %d is %d\n", number, result);
    return 0;
}

```