

Pre-Practical activity

1. Define recursion and how it works
2. Explain the concept of tail recursion and how it differs from regular recursion.
3. Why is tail recursion generally considered more efficient in terms of memory usage compared to non-tail recursion?
4. Rewrite this code to be a recursive function:

```
int factorialIterative(int n) {  
    int result = 1;  
    for (int i = 2; i <= n; i++) {  
        result *= i;  
    }  
    return result;  
}
```

5. Rewrite this code to use tail recursion:

```
int factorial(int n) {  
    if (n <= 1)  
        return 1;  
    else  
        return n * factorial(n - 1);  
}
```

6. Write a tail recursive function to compute the factorial of a number.

7. Write a recursive function that reverses a string.