**Okobi Kosisochukwu Jeremy  
Computer Engineering**

**20CJ027472**

1. Algorithm: Sum of Taylor Series
2. Inputs: x, a, n
3. Initialize: Let f[] be an array containing f(a), f'(a), f''(a), ..., f^n(a), sum = 0
4. For i = 0 to n
5. term = (f[i] / i!) \* (x - a)^i

v. sum = sum + term

vi. End For

vii. return sum

1. Algorithm for Calculating Factorial of a Given Integer n
2. Input n
3. Initialize: factorial = 1
4. If n == 0, then factorial = 1
5. For i = 1 to n
6. factorial = factorial \* i
7. End For
8. Return factorial
9. Algorithm for Pearson’s Correlation Coefficient
10. Input: arrays X[] and Y[], and their size N
11. Initialize: sumXY = 0, sumX2 = 0, sumY2 = 0
12. For i = 1 to N
13. sumXY = sumXY + (X[i] - X̄) \* (Y[i] - Ȳ)
14. sumX2 = sumX2 + (X[i] - X̄)^2
15. sumY2 = sumY2 + (Y[i] - Ȳ)^2
16. End For
17. r = sumXY / sqrt(sumX2 \* sumY2)
18. Return r
19. Algorithm for Newton’s Iterative Formula to Find the Reciprocal of N
20. Input: N (given N = 5)
21. Initialize: x0 = 0.1
22. For i = 1 to max\_iterations do
23. xi+1 = xi \* (2 - xi \* N)
24. Print xi+1
25. End For
26. Return final reciprocal value