

ECE 326 Lab #1 Report

1. If the system cannot print up until 100!, what is the reason?

The reason that a system would not be able to print up until the factorial of 100 would be due to the computer's architecture. For a computer with a 32-bit processor, it would only be able to hold values of integer type till 2^{32} (unsigned in the case of factorials). And for a computer with a 64-bit processor, it would only be able to hold values of integer type till 2^{64} unsigned. 100! Well exceeds the maximum value a 64-bit processor can handle. Thus, the main reason a system would not be able to print up until 100! would be due to the computer's architecture and processor size.

2. Find out the maximum number possible on your system. Provide reasons.

My computer is a MacBook Air 2020, and has an Intel i3 processor, which is a 64-bit processor. Thus, the maximum value that it can handle accurately would be under the value of 2^{64} . 20! is the largest value that is contained within the value of 2^{64} . Anything above the factorial of 20 exceeds 2^{64} .

3. Check all boundary conditions.

Ensured that negative numbers would not be accounted for and the numbers would only be between 1 to 100 by using a range between 1 and 101.

Used time library in order to calculate CPU time.

4. Calculate the total CPU time and report it. Did the time reduce with the optimized computation? Provide reasons.

CPU Time (Unoptimized): 0.0013099999999999987 seconds

CPU Time (Optimized): 0.0004520000000000001 seconds

Yes, the time taken to compute 1! through 100! did reduce with the optimized computation. The reason being is that in the unoptimized version, there is a double for loop which increases the number of loops and multiplication, and is $O(n^2)$. On the other hand, the optimized version uses only one for loop and is $O(n)$.