Reflection Report

Overview:

I carried out the unbounded integers assignment using a dynamic linked list in Fortran. A digit is represented as a linked list of digits (in reverse) so that arithmetic operations can be easily carried out.

Design Decisions:

Implementation of Linked List: Each digit is coded in a node, starting with the least significant digit first. Addition, subtraction, multiplication, and division are made simple to achieve.
Modularization: The program is divided into two files:
dynllist.f03 to implement data types and arithmetic operations (including factorial)
unbounded.f03 for the user interface.
Input Validation: The input is cleansed to remove spaces and control characters and only valid digits (and a possible sign) are calculated.
Zero Normalization: Zero is converted to positive in sign to avoid output like "-0".

Challenges:

Long division on a linked list was hard to do and to give correct results (e.g., 1/3 giving 0).
Dynamic memory handling and support for negative numbers were properly taken care of by prudent design.

How It Works:

The operation to be done is selected by the user and operands are taken as input.
The input is cleaned and transformed to a linked list representation (BigInt).
The proper arithmetic function is invoked. During division, if the result is 0, the program expressly returns a BigInt of 0.
The output is written to the terminal.

Compilation and Execution:

Compile with:
gfortran -Wall dynllist.f03 unbounded.f03 -o unbounded
Run with:
./unbounded