**Peter Julius M. Estacio**

**ENGG 151.01-A**

**February 11, 2026**

# Validation of Integers and Floating Point Numbers

For this homework, functions `is_int()` and `is_floating_pt()` were programmed to identify and validate integers and floating-point numbers from passed input strings. Should they be valid, they are able to modify the corresponding integer and double pointers to store the detected value.

The passed strings are first initialized in a `stringstream`. A temporary variable `n` is then initialized and set to `0`, serving as a buffer to store tokens from the `stringstream` using the insertion operator (>>). Should the inserted string be stored in `n` successfully and no strings follow it in the inputstream, the function returns true and the processed string is valid. From here, the passed pointer `*value` is modified and assigned to `n`. Otherwise, the function returns false, the processed string is invalid, and `0` is stored in `*value`. Headers were used to implement these functions for use in future projects.

To test these functions, the five strings `"abc", "123", "4.", "5.6a", "7.8e-2"` were stored in an array and used as inputs for both functions. Variables `testI` and `testD` were initialized as empty integer and empty double variables respectively, to be used as modifiable variables to be passed. For each string, the test program would output to the console the result of the integer and floating-point validation test. From there, the test program would output to console the data stored within `testI` and `testD`. The processed number would print to console should the string be valid, otherwise `0` is printed. All test strings returned accurate results as described in the homework specifications, with the passed variables modified to precisely store the processed values.