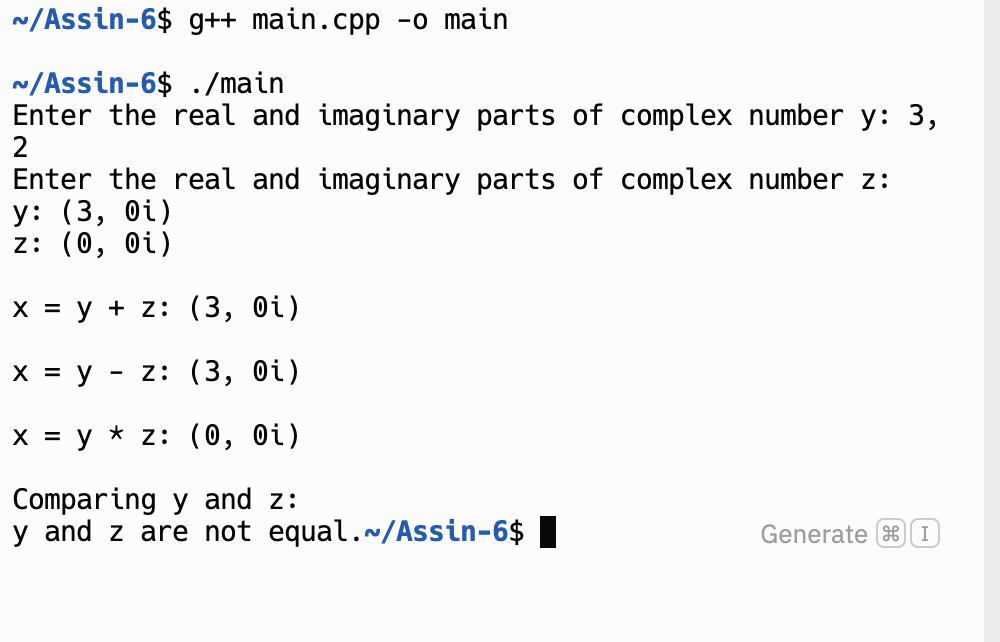
#Question-1:

**Output:**



#Question-2:

a.The HugeInt class is crafted to handle integers that are too large for standard C++ integer types. It captures each digit of these large numbers within an array, where each array slot holds a single digit.

Constructors are available to instantiate a HugeInt object from either a long or a string. This allows the class to accommodate various input formats. The class overloads the + operator to facilitate the addition of two HugeInt objects, or a combination of a HugeInt object with an int or a string, enhancing its arithmetic versatility. The << operator is overloaded to enable easy printing of HugeInt objects, ensuring they can be displayed cleanly and efficiently.

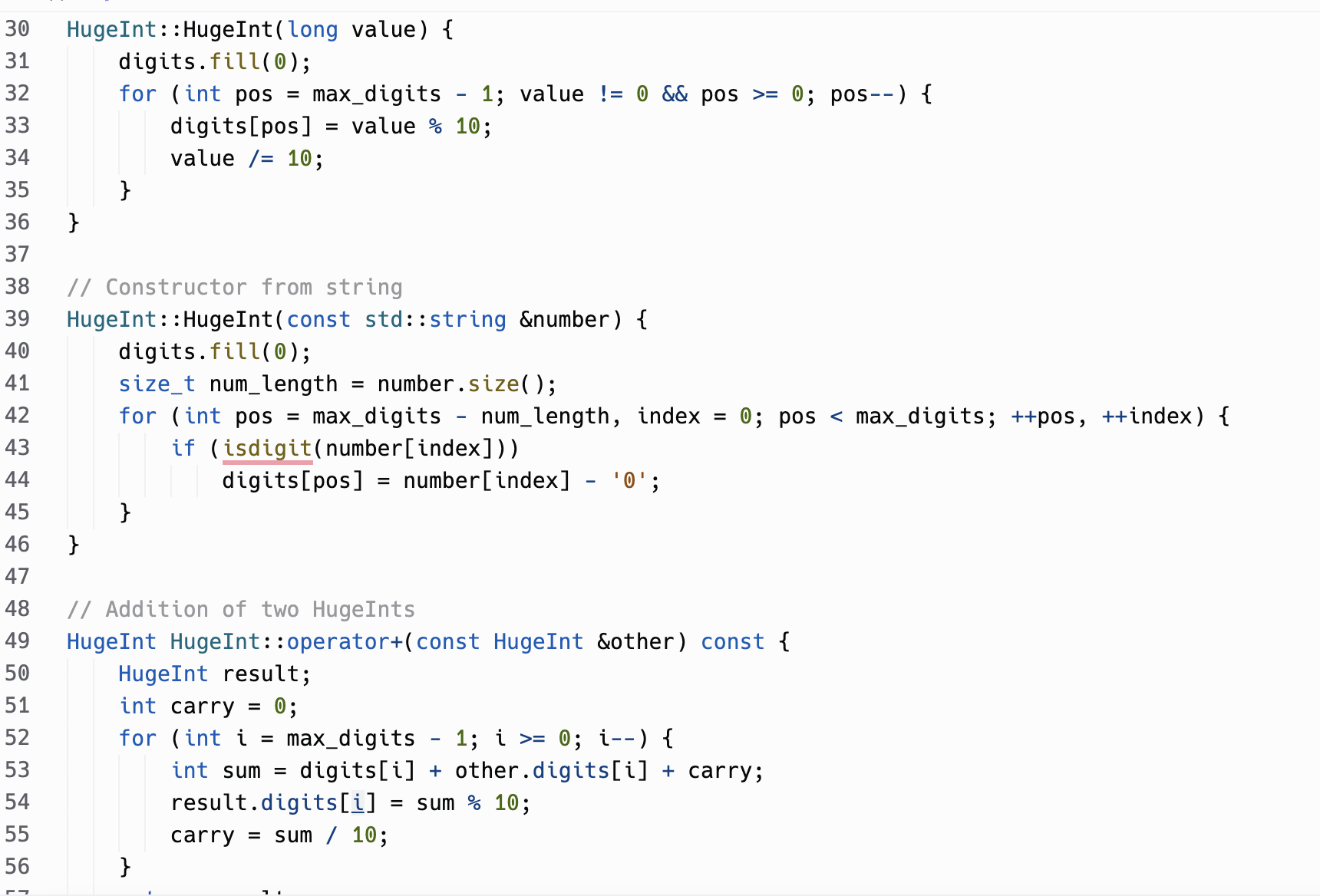
B. Positive Integers Only: The class is designed exclusively for positive integers, lacking the capability to represent or compute with negative values.

It is constrained by a set maximum number of digits it can manage, which is 30 digits. This limitation restricts the size of the integers it can accurately handle.

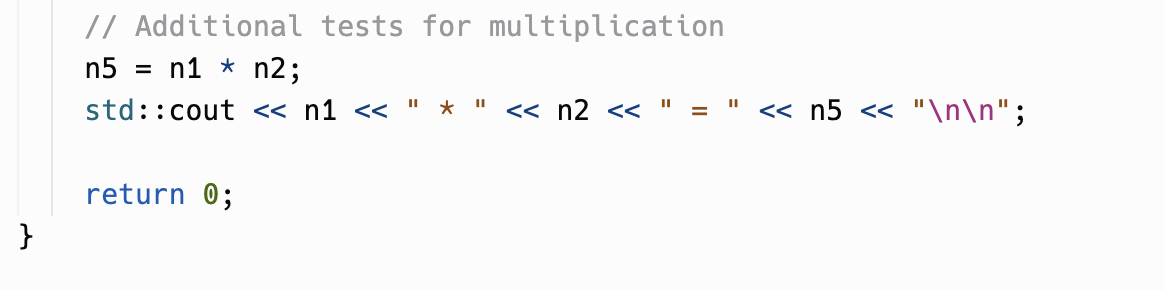
Limited Arithmetic Support: Currently, the class only supports the addition operation. It does not provide functionality for other arithmetic operations such as subtraction, multiplication, or division, which limits its applicability in broader numerical contexts.

No Comparison Operations: There is no built-in functionality to compare HugeInt objects, meaning relational operations like greater than, less than, or equality checks are not implemented. This restricts its use in conditions that require sorting or direct comparison of large numbers.









**Output:**

