BigInteger-Library

- Create a class called `BigInteger` that should represent large integer values with the following features
- Create a constructor that takes a `std::size_t` as input
- Create a function that outputs the content of the BigInteger in one of the following formats:
  - decimal number
  - hexadecimal number
- optional:
  - Create a constructor that takes a `std::string` as input
  - Create a constructor that takes a custom literal as input

First Stage

- Implement the following operations for your class:
  - (In)Equality-check
  - Comparison (less-than, etc)
  - Addition, Subtraction
  - Bit-shift

Second Stage

- Implement the following operations for you class:
  - Multiplication
  - Division

Third Stage

- Implement the following operations for you class:
  - Modulo (remainder of division)
  - Square-and-Multiply to calculate nth power in a residue class ring

Evaluation Criteria

- Functionality:
  - The code needs to compile and run without errors in C++14 or C++17
  - The code needs to provide the required functionality and follow the specifications
- Readability:
  - The implemented functions need to be comprehensible
  - The structure of the source code should be obvious and reasonable
- Best pratices:
  - Use known best practices for the placement and hierarchy of implemented functions
  - No functionality should be implemented twice or even more often
  - Avoid compiler warnings where possible, don't create quick workarounds for compiler warnings
- Performance:
  - Faster and more efficient code is preferred
  - This should not come with a tradeoff of readability
  - Real time performance as well as theoretical complexity is measured