CS222: Assignment 6 - Modular division

1. Submission deadline: Monday, 13 Feb at 3:00 pm.
2. Follow good coding practices to gain more marks.
3. No copying among the students or from the Internet or any other source.
4. The assignment can be submitted in groups of size ≤ 2.
5. Submit a `.cpp` file and a `.pdf` file.
6. Write the names and roll numbers of the students at the top of each file.
7. The files should be called

   `noModNDiv_firstRollNumber_secondRollNumber.cpp`,

   `noModNDiv_firstRollNumber_secondRollNumber.pdf`,
8. In case you do not know about C++ templates, check `https://www.learncpp.com/cpp-tutorial/template-non-type-parameters/`.
9. For more information about the assignment: `https://stackoverflow.com/questions/66546257/in-c-can-we-create-a-class-for-each-integer`
10. You need to extend the functionality of the mod N class that you defined in assignment 5. Please make a copy of that C++ file and add the lines of code for this assignment. Use comments to clearly demarcate the added lines.
11. For more about exception/runtime error handling in C++, i.e. `try-throw-catch`, check out `https://www.w3schools.com/cpp/cpp_exceptions.asp` and the top answer in `https://stackoverflow.com/questions/6121623/catching-exception-divide-by-zero`

---

1. (10 points) In this assignment, you will overload the division operator for numbers modulo N (`noModN`) class.

   1. Extended Euclid's algorithm is implemented in a `private` function `extendedEuclid`.
   2. Define the `public` function `findInverse(noModN)` that returns the inverse of a number modulo N if it exists. Else it throws a *runtime error for dividing by zero*. To find the inverse, use the function `extendedEuclid`.
   3. Overload the `/` operator that on inputs x, y returns xy^{-1}. It throws the *runtime error for dividing by zero* if y does not have an inverse modulo N.
   4. Take $N = 60$ for this assignment.
   5. In the pdf file, output the run for 6/10. And for 6/31.
   6. Also give an analysis of the Extended Euclid's algorithm. Remember the size of integers also play a part.