Mathematics 3159A Assignment 1 Fall 2020

Instructions

- This assignment is due on Tuesday, September 22, 2020 at 2:00 PM EDT. Late submissions will **not** be accepted.
- This assignment consists of two problems. You should choose one for submission.
- Your solution needs to be formatted using the LaTeXtemplate available on OWL.
- All solutions must be written in full sentences.
- You are not allowed to work with others or use any online resources.
- This assignment is worth 5 points.

Problem 1.

In this exercise, we will study solution of linear equations in modular arithmetic. To this end, consider the following congruence

$$ax \equiv c \mod m,$$

where a, c, and m are fixed integers.

1. Show that this congruence has a solution, i.e., there exists an integer x satisfying it if and only if $\gcd(a, m)$ divides c.
2. Show that if there is a solution, then there are exactly $\gcd(a, m)$ distinct solutions in $\mathbb{Z}/m$.

Note that this is in contrast to ordinary algebra, where an equation $ax = c$ can have only one solution.

Problem 2.

In this exercise, we will be computing inverses in modular arithmetic.

Statement

The assignment has two parts.

1. Write a function in Python3 called `solve` that, given integers $N > a > 0$, returns $a^{-1} \mod N$ if such a number exists, or a warning saying that a is not invertible mod N if it does not.

2. Download the file `generate_input.py` from OWL, use it to obtain three pairs (N, a) by running

   ```
   python generate_input.py [last three digits of your student number]
   ```

   and run your program on these three inputs.

Your submission must consist of a single PDF file containing:

1. the *Python code* implementing your solution;
2. and the three *inputs you generated*, and the *output of your program* run on these three inputs.

Examples

Here are some examples of what your function `solve` should do:

```
>>> solve(10,7)
the inverse of 7 mod 10 is 3
>>> solve(10,8)
8 is not invertible mod 10
>>> solve(11,8)
the inverse of 8 mod 11 is 7
```

Notes

- The numbers generated by `generate_input.py` are quite big, so a brute-force solution will not work.
- The file `generate_input.py` is written in Python3, and so should be your solution. Make sure you are using a 64bit version of Python3
- Your submission must use the LaTeX template available on OWL.
- Your code should not make use of any external libraries such as `numpy` or `math`. All the auxiliary functions should be implemented by you, and should be included in your submission. You should only use the most basic arithmetic operations such as `+`, `-`, `*`, `//`, `%`.
- Comments in the code are not mandatory. However in the case of an incorrect solution, the comments can provide grounds for partial credit.