CS 153 PROJECT
(spfestin)

INSTRUCTIONS:

1. This is an individual effort work.
2. Submission deadline is on 25-May-2017 (Thursday). Follow the submission guidelines indicated below. Incomplete submissions will not be accepted.
3. Late submissions will be docked 5 points for each day late.

Using any of the following programming languages (Python, C, or Java), write a Galois Field Calculator GF(2^m). You are not to use any special-purpose (crypto) libraries.

There are three expected inputs:

1. A(x)
2. B(x)
3. P(x) (irreducible polynomial)

You can assume that P(x) is indeed an irreducible polynomial.

The input polynomials would be entered as decimal coefficients separated by spaces, for example the polynomial P(x) = $x^3 + x + 1$, would be entered as 1 0 1 1. As another example, the polynomial A(x) = $x^3 + 7x + 6$, would be entered as 1 0 7 6 (which means **1** x^3 + **0** x^2 + **7** + **6).**

After entering the three inputs, the user could pick to perform one of the following operations:

1. A(x) + B(x)
2. A(x) - B(x)
3. A(x) x B(x)
4. A(x) / B(x)

The output should display:

1. A(x)
2. B(x)
3. The result given the operation chosen by the user (+, -, x, /).

Additional points will be given if details of the computation are displayed, step-by-step.
Reference:
You may wish to look at http://www.ee.unb.ca/cgi-bin/tervo/calc2.pl as a guide to a similar GF calculator.

FOR SUBMISSION ON 25-MAY-2017.
PLEASE READ THE DETAILS OF THE SUBMISSION BELOW AND USE THE CHECKLIST TO SEE IF YOU HAVE COMPLETED ALL REQUIREMENTS:

CHECKLIST OF REQUIREMENTS FOR CS 153 PROJECT SUBMISSION	
1	Source code link. Source code must be hosted on a git repository (use github.com or bitbucket.com).
2	Project writeup (PDF file). In at most 5 pages, describe the following: a) Your details: Full Name, Student Number b) Programming Language Used: c) Operating System Used in development: d) Git repository link: e) Reflection on the development process, answer the following questions: [1] Which part(s) of the project, if any, did you find easy to do? Why do you think did you find these easy to do? [2] Which part(s) of the project, if any, did you find challenging to do? Describe how you solved these challenges. f) Reference used.
3	Compilation instructions (if applicable) and/or installation instructions. This should be sufficient for someone to re-compile or build from the git repository.
4	Executable program.

POINT SYSTEM FOR PROJECT GRADING:

FEATURE/FUNCTION	POINTS
Input Validation. Check that there is properly-formed input for A(x), B(x), and P(x), as described. Assume P(x) is indeed an irreducible polynomial.	20
Correct computation of A(x) + B(x)	20
Correct computation of A(x) - B(x)	20
Correct computation of A(x) x B(x)	20
Correct computation of A(x) / B(x)	20
TOTAL	100
(Bonus) Detailed computations.	40

If there are problems with compiling or building from source code, a demo may be scheduled.