Please write your hackerrank username somewhere on your problem set.
Written problems are due at the beginning of class on Friday, October 16. Programming problems must be electronically submitted by 10:50am according to the instructions on problem set 1. The submission page for the programming questions is the following.

`https://www.hackerrank.com/math158-problem-set-4/`

Written problems

1. Solve each system of congruences. Your answer should take the form of a single congruence of the form $x \equiv c \pmod{m}$ describing all solutions to the system.

   (a) $x \equiv 1 \pmod{3}$
   $x \equiv 2 \pmod{5}$

   (b) $x \equiv 6 \pmod{11}$
   $x \equiv 2 \pmod{10}$

   (c) $x \equiv 2 \pmod{3}$
   $x \equiv 1 \pmod{10}$
   $x \equiv 3 \pmod{7}$

   (d) $x \equiv 6 \pmod{8}$
   $x \equiv 3 \pmod{9}$
   $x \equiv 17 \pmod{17}$

2. The element $288 \in (\mathbf{Z}/919)^\times$ has order 17. Use the babystep-giantstep algorithm, making using of the fact that the order of 288 is known to be 17, to evaluate the discrete logarithm $\log_{288} 162$ for the prime $p = 919$. You may use a computer to do the arithmetic, but show explicitly the two lists from which you find the collision.

3. Evaluate the discrete logarithm $\log_{40} 33$ for the prime $p = 73$ using the Pohlig-Hellman algorithm, according to the following steps (see the statement of Theorem 2.31 in the textbook for details on the notation). You may use, without proof, the fact that 40 is a primitive root modulo 73.

   (a) Let $N = \operatorname{ord}_{73}(40)$. Factor N into prime powers as $N = q_1^{e_1} \cdots q_t^{e_t}$.

   (b) Determine the numbers g_i and h_i for each i from 1 to t inclusive. For each i, what is the order of g_i modulo 73?

   (c) For each i, evaluate the discrete logarithm $y_i = \log_{g_i} h_i$, using a method of your choice.

   (d) Solve the system of congruences $x \equiv y_i \pmod{q_i^{e_i}}$ to obtain the discrete logarithm $x = \log_{40} 33$.

4. Consider the set $\mathbf{N}$ of positive integers, equipped with the following operation.

$$x \star y = \max(x, y)$$

   Show that $(\mathbf{N}, \star)$ satisfies all of the conditions in the definition of a group (as on page 74 of the textbook) except one. Which condition does not hold?

5. (a) Consider the set M consisting of all 2×2 matrices with integer entries, with the operation $\cdot$ being ordinary matrix multiplication. Show that $(M, \cdot)$ is *not* a group.

(b) Let S denote the subset of M consisting of those matrices with determinant equal to 1. Show that $(S, \cdot)$ *is* a group. (This group is usually denotes $\mathrm{SL}_2(\mathrm{Z})$ and is called the *special linear group of degree* 2 *over* $\boldsymbol{Z}$).

(c) Show that $(S, \cdot)$ is not a *commutative* group.

Programming problems

6. Write a program which solves the a discrete logarithm problem, where the base of the exponentiation has a *known* order considerably smaller than the prime number p. Specifically, your program will read four integers p, g, a, N, where p is a 1024 bit prime, g, a are elements of $\mathbf{Z}/p$, and N is a 32-bit integer guaranteed to be equal to the order of g modulo p. It is further guaranteed that some power of g is congruent to $a \pmod{p}$. Your program should print an element e of $\mathbf{Z}/N$ such that $g^e \equiv a \pmod{p}$.

7. Write a program which takes as input an integer n and n pairs of integers y_i, m_i, and prints a pair of integers x, m, where $x \pmod{m}$ is the solution to the system of congruences $x \equiv y_i \pmod{m_i}$. The n integers m_i are guaranteed to be pairwise relatively prime.

   Note. After solving this problem, you will have written most of the necessary ingredients to implement the Pohlig-Hellman algorithm for discrete logarithms, which will be assigned on problem set 5. If you wish to try implementing Pohlig-Hellman now, this problem is already open for submissions; change the "4" to a "5" in the hackerrank url.

8. Let m be any positive integer, and let G denote the set of all 2×2 matrices A with entries chosen from $\mathbf{Z}/m$ such that the determinant of A (which you can regard as an element of $\mathbf{Z}/m$) is a unit modulo m. Then G, with usual matrix multiplication (where you should reduce each entry modulo m after computing it in the usual way) forms a group, usually denoted $\mathrm{GL}_2(\mathbf{Z}/m)$. Write a program which takes an integer m and the four entries of an element of G and prints the inverse element in G, represented as the four entries of the matrix, all reduced modulo m.

   Note. The group $G = \mathrm{GL}_2(\mathbf{Z}/m)$ is usually called the *general linear group of degree* 2 *over* $\boldsymbol{Z}/m$. In contrast to "special linear" groups, "general linear" groups are defined by the weaker condition that the determinant of the matrix is invertible, rather than the stronger condition that this determinant is exactly equal to 1.

9. Let G be as in the previous problem. Write a function which takes as input the integer m, the four entries of an element A of G, and an integer n (which may be positive or negative) and returns the element A^n of G. Note that n may be quite large; you should use the fast-powering algorithm to ensure that your program will finish in time.