2824 Cryptography

I have neither given nor received unauthorized assistance_____.

1. (10pts) Finding Modular inverses.

    A. Find 5 numbers x such that: $x * 3 \textbf{ mod } 10 = 1$

Notice that just as we have inverses in multiplication, for example, $3 * \frac{1}{3} = 1$, we can have inverses of 3 **mod** n.

B. Are modular inverses unique the same way multiplication inverses are unique? Answer why or why not ?

C. What is the same about each of the x's you found in Question 1?

2. (10pts) In ordinary multiplication, inverses always exist.
Can you always find a modular inverse?

Try $x * 5 \textbf{ mod } 10 = 1$ and $x * 2 \textbf{ mod } 10 = 1$

For which of the following do modular inverses exist?

(Find an inverse x if you can. Use a Guess and Check method)

$x * 7 \bmod 10 = 1$

$x * 26 \bmod 13 = 1$

$x * 7 \bmod 5 = 1$

$x * 5 \bmod 21 = 1$

$x * 6 \bmod 35 = 1$

$x * 7 \bmod 13 = 1$

What is the rule for when Modular inverses exist?

3. (10pts) Work through #1 - 4 page 284. Summarize important ideas here that show you have studied this topic.

4. (10pts) You are stranded on an island. Each day 2 jugs wash up on the beach, which are marked with exact measures of a and b cups. You also have an empty (unmeasured tub), and a jug maker. Jugs last one day and then disappear. An evil Volcano Demon requires you to measure out exactly 1 cup each day or someone will be tossed into the Volcano (then, Volcano Demon can make a single cup measure, and make big cake recipes measured in cups - his dream is being on a cooking show one day).

Below are the jug measures in cups that show up each day for 9 days.
For each day:

- Using the provided cups and the jug maker, **find the smallest cup measure** you can create each day; use the jug making videos or Euclidean Algorithm for intuition. You may put down your answers: you do not need to show your work.
- On which days will someone be tossed into a Volcano (you are unable to measure out 1 cup)?

1. 12, 9

2. 27, 5

3. 8, 4

4. 66, 11

5. 25, 5

6. 12, 3

7. 9, 4

8. 141, 19

9. 89, 55

5. (10pts) The Volcano God again. Same set up, HOWEVER, the jug maker is broken. This time you have the same need to make the smaller measure with jugs but you have **no jug maker.** Notice the Euclidean Algorithm requires that we use these new jugs (the remainders) again and again.

For day 8, a day no one was thrown into the Volcano, show the recipe for measuring out a single cup measure. Show your work. You may use either a formal or informal process to find the answers (the recipes numbers are the Bezout Coefficients).

Remember:

- What tells me the smallest amount I can measure? (GCD)
- What tells me the recipe for measuring? (Bezouts)

6. (10pts) Notice there are 3 fundamental situations with the pairs of jugs.

- Some pairs of jugs you can use to measure any amount asked for.
- Some pairs of jugs you can only measure multiples of the given jugs.
- Some pairs of jugs which can measure some additional amounts smaller than each of the given jugs, but not all possible amounts.

Explain mathematically what each case means in **terms of Primes, GCD, etc..**
Give an example of each.

From #4 notice that we can use the EA to find if an inverse of two integers exists (easy-ish).
From #5 notice that actually finding the inverse (finding Bezout Coefficient) is harder .

Things We Know

Q = Bezout THM - for all a, b there exist s and t such that gcd(a,b) = s * a + t * b
I = an inverse of a mod b exists
R = a and b are relatively prime.
G = gcd(a,b) =1

More Things We Know (Theorems)

I => R
R => G
Q => 1 = s * a + t *b

Consider $1 = (s * a + t * b)$

$1 \bmod b = (s * a + t * b) \bmod b$

$= (s * a + 0) \bmod b$ (because $t*b \bmod b = 0$)

$= s* a \bmod b$

Thus 1 mod b = s* a **mod** b, and hence 1 = s* a **mod** b.

So s must be an inverse of a **mod** b. So the extended Euclidean Algorithm gives us **s** and **t**. And **s** is the inverse we want.

(ungraded) IN YOUR OWN WORDS - how are Modular Inverses, and Bezout's theorem related?

In order to create an efficient algorithm for EEA, we will use some tricks in the code.
In particular, instead of working through the process by hand - with jugs, or the long "by hand" examples in the book, we'll use a formula for updating the "recipe" for jugs.

The EEA algorithm will be a key component of your RSA Project

7. (10pts) Now let's formalize the process for this new method without a jug maker, which is the **Extended Euclidean Algorithm** by working through the algorithm using the example in the video.

1. Using 77 and 14, what variables represent 77 and 14 in the algorithm? (be specific)

2. Which values will update through the program, m and n, or m_0 and n_0 ?

s1, t1 and s2, t2 are the initial "recipes" for m and n.

77 = 1 * 77 + 0 * 14
14 = 0 * 77 + 1 * 14

3. Rewrite above using s1, t1 and s2, t2 and m and n, and m_0 and n_0

   m =

   n =

Before the loop we have now initialized our settings.

4. What is the condition on the loop (what will cause it to stop?)

5. What are k and q and why are we finding them?

6. Which lines update m and n? (write them here)

The code in blue is calculating the new values (the new "recipe") of s1, t1 and s2, t2. Notice the use of $\widehat{s1}$, $\widehat{t1s2}$,$\widehat{t2}$

7. How are these DIFFERENT variables related to s1, t1 and s2, t2?

8. How many distinct variables are being updated in the blue code?

9. How many are being updated in the pink code? (look carefully)

(Optional/Recommended) Work through the example chart from the EEA video and reproduce it here.

(Optional worksheet for deriving the trick that makes this algorithm work)

Perhaps the most important/clever/trickiest step in code is using the formulas (s1 - q * s2) and (t1 - q * t2) to update the coefficients. If you would like to know where/how these essential formulas are derived, fill in the worksheet below.

Initial recipes for b and a
b = Jug 1
a = Jug 2

Recipe for b = __ * Jug1 + __* Jug2 s1 = ___ and t1 = __ **(scoops to get b)**

Recipe for a = __ * Jug1 + __ * Jug2 s2 = __ and t2 = __ **(scoops to get a)**

Since we no longer have a jug maker to measure out our remainder, we need a recipe for the new measure using the original jugs.

Our new measure, k, comes from putting a's (Jug2) into b (Jug1), and seeing what is left.
k = ___ mod __ Our new measure
q = _____ The number of scoops that fit into b

Now we need a recipe for our new k (using b, a, and q) k = _____

Great, but how do we measure this with our original jugs, Jug1 and Jug2 ?
*k = b - q * a*
*k = (**s1** Jug1 + **t1** Jug2) - q (**s2** Jug1 + **t2** Jug2) (from above - a and b from jugs)*
*k = (**s1** Jug1 + **t1** Jug2) - _____ (multiply thru the second part)*
*k = **s1** Jug1 + **t1** Jug2 - q **s2** Jug1 - q **t2** Jug2 (remove parenthesis - assoc. property)*
k = _____ (group like terms - commutativity)
*k = (**s1** - q **s2**) Jug1 + Jug2 (**t1** - q **t2**) (factor)*

We do not need all the above algebra in our code - we just have the following magically appear:

I need (_____) of Jug 1 and (_____) of Jug2 to measure k.

Project Check Points

8. (10pts) Write the code for the Extended Euclidean Algorithm using Sriram's algorithm. Show the code here in plain text with **comments** - *use feedback on code style from last week. This can be a screenshot.*

9. (10pts) Test your code with #41 - 44 page 273. Show the results as linear combinations as instructed (either by hand or you can include as code).

10. (10pts) Finally, for a project check in, work through the example on 8 and 9 page 300 - 301 on your own.

Create a similar SIMPLE 4 letter word as an example following this format and using the keys from the example.