Submission guidelines:

1. Please submit your write-up as a single PDF file. That file should be named in the following manner: Lastname_3.pdf   (For example: Vadrevu_3.pdf)
2. Any additional scripts that you write as part of your analysis should also be included. These can be given any suitable names that you like.
3. Finally put all these files into a directory that should be named as follows: Lastname_4. Please compress this directory and upload it to Moodle.
4. The deadline for this assignment is 11:55 PM on April 18th, 2022 (Monday)

Question 1. [20 points]

Using the extended Euclidean algorithm, compute the greatest common divisor
and the parameters s,t of
1. 198 and 243
2. 1819 and 3587

For every problem check if $s.r_0 + t.r_1 = gcd(r_0, r_1)$ is actually fulfilled. Show what happens in every iteration step.

Question 2. [20 points]

1. Determine φ (m), for m = 10, 24, 30 by using this definition: φ (m) is the number of positive integers that are smaller than m and are co-prime with m. (You do not have to apply Euclid's algorithm for finding co-primes. Simply, list all the co-primes of m less than m and count them.)
2. Now, compute the φ (m) using the Euler's phi function formula (totient function) and verify that the result matches what was obtained above.

Question 3. [60 points]

Please submit code for both the tasks that you are required to do below along with answers for the questions asked.

- In this question, you will need to write code to find the modular inverse of any large number modulo any large number. Before that, you need to write a simple program that contains a loop construct to iterate a large number of times doing nothing. How many digits does your loop counter have in 5 minutes? Next, let the program run for 60 minutes. How many digits will the loop counter now have? This is just to demonstrate the limitation of O(n) implementations when the inputs are large numbers (of the order of hundreds of digits). This will clearly show that the naive trial and error approach will not work for large numbers.

- Next, you need to implement the Extended Euclidean Algorithm for finding an inverse. Use your code to find the following.

  Find inverse of x mod n where x and n are the following pairs (x, n):

  1. (13,
     5802166458563979118118402595044024839822613606951693823249368750582247183653682429882273371034225069773999682593823264194067085762451410312598613405099769716012730154799578846813788765182370710
     2007839)

  2. (250556952327646214427246777488032351712139094643988394726193347352092526616305469220133287929222242315761834129196430398011844978805263868522770723615504744438638381670321613949280530254014602887707960375752016807510602846590492724216092721283154099469988532068424757856392563537802339735359978831013,
     3373173703468584728999065980339992781006346742496369077930826947841348985090302550149421077377734508912005371590030093819999810288613548901222608296089578494563874807828486514870556035600292455467467109896792098065671846896181713928254906078154942647385027932581418962537100702721767973469229451341245778940484167761735597977769838688336125466941182558530861180855721395461768669312636029598922294022)