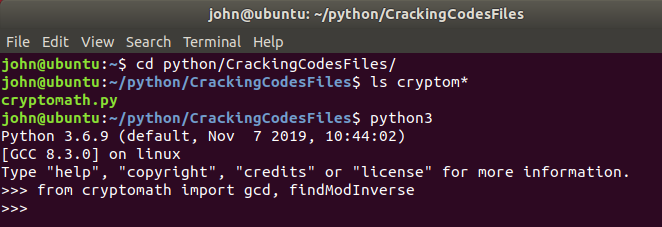
# Cryptography Homework 2b—Modular Arithmetic with Python—KEY

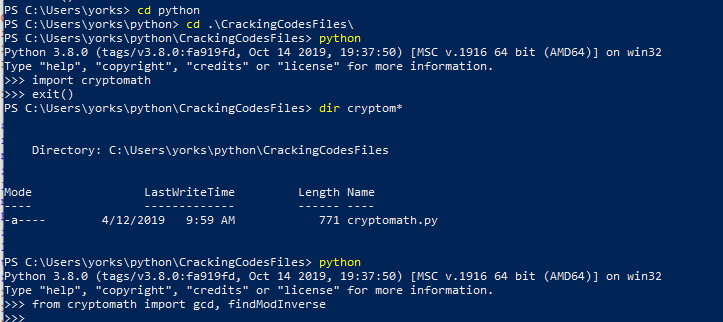
## Setting up in Idle



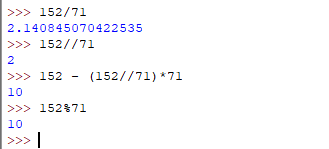
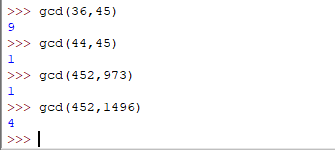
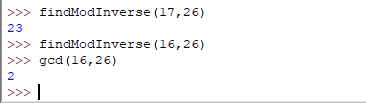
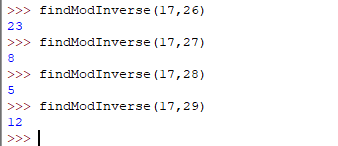
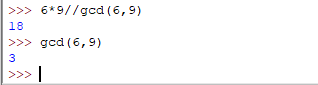
## Setting up in a Linux terminal



## Setting up in a Windows terminal



## For Turn In

1. Compute (you can do this easily at an interactive Python prompt. The mod operator in Python is “%”)
   1. 34 mod 18  
      
   2. (34 + 97) mod 12 Note: the mod operator has the same priority as multiplication. In a tie, Python executes operators from left to right.  
      
   3. 14 \* 71 mod 15  
      
   4. 152 / 71
   5. 152 // 71 (in Python, // is integer division)
   6. 152 – (152 // 71) \* 71 (this is the remainder after integer division)
   7. 152 mod 71 (you should see that the answer is the remainder when you divide 152//71 (integer division)  
      
2. Compute gcd(36, 45) and gcd(44, 45) using the cryptomath.py module you added above. You should get 9 and 1 as answers, as a check to make sure the code is correct. Now compute gcd(452, 973) and gcd(452,1496). Which one of the pairs of numbers relatively prime, and what is the GCD of the pair that is not relatively prime?  
     
   452 and 973 are relatively prime because their GCD is 1.  
   452 and 1496 have a GCD of 4, so they are not relatively prime
3. For the following numbers, compute the GCD of the number and the modulus. If the number and the modulus are relatively prime, compute the multiplicative inverse. You can use either brute force in Python, or the findModInverse() function from cryptomath.)
   1. 17 mod 26
   2. 16 mod 26
   3. Of the two numbers above, which could be used for the key of an affine cipher and which could not? Why?  
        
      The inverse of 17 mod 26 is 23. Since 16 and 26 share a common factor (GCD is 2), 16 does not have an inverse mod 26.
   4. Compute the multiplicative inverse of 17 in modulus 27, 28, and 29. Note that the inverse is completely different when the modulus changes.  
        
      The inverse is different in every modulus we calculated.
4. Least Common Multiple. The lcm(a, b) is the smallest number that can be divided by both   
   a and b. It is easily computed as a \* b // gcd(a, b)
   1. Compute the lcm of something simple, like 6 and 9. Check your answer by hand to verify that the equation is correct.  
        
      It is easy to see that 18 is the smallest number that has multiples 6 and 9.
   2. Compute lcm(252, 196)  
      