

1. First, I read the Wikipedia page for the method. We have learned a bit about RSA in math before, so I went back and looked at my old notes. Then I discussed with someone else who has this course about how we thought it was logical to solve it. First, I used the prime generator to find possible primes to use as p and q. Then I checked if p\*q = the given n. This must be true for p and q to be possible candidates. Then it runs through the possible p and q combinations by using the method for multiplicative inverse to find a possible d. To test if this d is the right one, we decrypt the message and check if it fits with “m.startswith("h")”. If it fits, then we are done and print our data and stop the loop.
2. This brute force of the private key is not a good thing for security. That is why it is normal to use large prime numbers. In this task, the prime numbers used are very small and that makes it easy and fact to run through all the alternatives and find the private key.
3. I personally didn’t code any other methods to the one that is working, but I have discussed the task with other students and they have used other methods about number theory and more straight forward brute forcing, so it is possible to do it in a different way.