Lab3: Primality Testing

Part 1

Implement the **modexp** function in Figure 1.4 (pg 19 of the text book) for modular exponentiation. It should take as input three integers x, y, and N and return $x^y \bmod N$

Part 2

Implement the randomized primality testing algorithm in Fig 1.8 (page 27 of the text book) using the **modexp** from part 1 as a subroutine. Given inputs N and k, you should determine whether N is prime or not using k trails. Print N "is prime" or N "is not prime".

What to turn in

You can test your code on arbitrary integers N to test if they are prime or not. In general, with k trials the probability of returning a wrong answer is $1/2^k$. However, Carmichael numbers are composite numbers that pass Fermat's little theorem for all integers aa relatively prime to N.

For each of the following Carmichael numbers determine the probability of failure in k trials, using k=1000. That is, the probability that the number is determined to be a prime when in fact it is not.

The first 32 Carmichael numbers are: 561, 1105, 1729, 2465, 2821, 6601, 8911, 10585, 15841, 29341, 41041, 46657, 52633, 62745, 63973, 75361, 101101, 115921, 126217, 162401, 172081, 188461, 252601, 278545, 294409, 314821, 334153, 340561, 399001, 410041, 449065, 488881.

Verify that these probabilities are much higher than $1/2^k$.