**Mark Eatough**

**CSIS 2430 9:00 Class**

**Programming Project 3**

**Modular Exponentiation Program**

**Assignment objective:**

Implement Modular Exponentiation.  You will need Algorithms 1 & 5 from Chapter 4. Your algorithm will need to solve problems 25-28 on page 255.

**What Worked?:**

Once I figured out what the algorithms were doing implementing them was pretty simple. I only needed about ten to twelve lines of code for each method.

**What did not work?:**

Python parses out strings reverse from what I was expecting. I had assumed that the first character in the string would be at position 0, but that was actually the last character in the string.

**Comments:**

For this assignment figuring out the algorithms took a lot more work than actually implementing them. This is probably more indicative of how programming will be in the real world as a lot of programming jobs involve code maintenance or developing small parts of code for a large program rather than developing an entire program by yourself.

1 '''  
 2 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
 3 \* Discrete Structures  
 4 \* Modular Exponentiation Program  
 5 \* Programmer: Mark Eatough  
 6 \* Course: CSIS 2430   
 7 \* Created September 15, 2013  
 8   
 9 \*This program finds the modulus of a number raised to a   
10 \*large power given the number, the exponent, and the modulus  
11 \*The algorithm 1 method converts the exponent to a binary number  
12 \*and the algorithm 5 method calls the algorithm one method, and  
13 \*then uses that to find the modulus of our very large number.  
14 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
15 '''  
16 #algorithm 1 out of book, used to find binary representation of my exponents  
17 def algorithm1(n,b):  
18 q = n  
19 binary = ""  
20 while(q > 0):  
21 a = q%b  
22 q = q/b  
23 binary = str(a) + binary  
24 return binary  
25   
26 #algorithm 5 out of book, used to find modulus of very large numbers  
27 def algorithm5(n, b, m):  
28 x = 1  
29 binary = algorithm1(b, 2)  
30 i = len(binary)-1  
31 while (i > 0):  
32 n\*=n  
33 n = n % m  
34 if(binary[i-1] == '1'):  
35 x\*=n  
36 x = x % m  
37 i-=1  
38 return x  
39   
40 #print out the exponent in binary code  
41 print "Find the binary translation of my exponents using algorithm 1:\n"  
42 print "644 in binary is:\t", algorithm1(644, 2)  
43 print "644 in binary is:\t", algorithm1(644, 2)  
44 print "2003 in binary is:\t", algorithm1(2003, 2)  
45 print "1001 in binary is:\t", algorithm1(1001, 2)  
46 #print out my modulus  
47 print "\n\nModulus using algorithm 5:\n"  
48 print "7^644 mod 645 = \t", algorithm5(7,644, 645)  
49 print "11^644 mod 645 = \t", algorithm5(11,644, 645)  
50 print "3^2003 mod 645 = \t", algorithm5(3,2003, 99)  
51 print "123^1001 mod 645 = \t", algorithm5(123,1001, 101)

