CS 310-0
Homework Assignment No. 7
Due Tue 3/6/2001

1. Students that are weak in algebra tend to think that the operation of "raising to a power" is distributive respect to addition, i.e., $(x+y)^p = x^p + y^p$. This is obviously false (except in some trivial cases such as $p = 1$), but it turns out to be surprisingly true in arithmetic "modulo p" when p is prime. In other words, if p is a prime number then:

$$(x+y)^p = x^p + y^p \pmod{p}.$$

Prove it. [Hint: prove that $\binom{p}{k} \equiv 0 \pmod{p}$ for every k such that $0 < k < p$.][1]

2. Find all elements x in $\mathbb{Z}_{15}$ such that $x^2 = 3x - 2 \pmod{15}$.

3. Prove that there are no positive integers x, y, z such that $3 \nmid xyz$ and $x^3 + y^3 = z^3$. (Hint: think modulo 9.)

4. Let N be the number $N = 13^{3^{17^{9^{7^{12}}}}}$. Note that exponentiation associates right to left, i.e., a^{b^c} means $a^{(b^c)}$, not $(a^b)^c$. So, in order to compute N first we must compute $7^{12} = 13841287201$, then $9^{7^{12}} = 9^{13841287201} =$ (a number of 13207944633 digits), and so on. Of course a complete computation of N is impossible, but it is fairly simple to find the two rightmost digits of N in base 10. Do it. (Hint: find N mod 100. All necessary computations can be made with pencil and paper, or with a simple pocket calculator with no special features.)

5. Right before closing you enter a pet food store and see that all cans of dog food are arranged on a large table as a rectangle of 16 long rows of cans. You buy 11 cans and see the owner rearrange the remaining ones quickly into a rectangle with 9 rows. Then you realize that you actually need 25 cans, so before leaving you buy another 14 cans. While checking out you see the owner rearranging the remaining cans into a rectangle of 25 rows. After going out you see a sign inviting clients to guess the number of cans of dog food sold during that day and offering to the first one to come up with the right answer a reward of 10¢ per can sold. The only clue is that at opening time that morning there were 5000 cans of dog food in the store. It turns out that you just attended a CS 310 class in which the instructor taught the Chinese Reminder Theorem, and the new knowledge that you acquired is still fresh in your memory, so you scratch a few numbers on a paper and compute the number of cans sold during the day. Next morning you go back to the store with the right answer and receive your prize consisting of how many dollars?

(next page →)

[1] Note that the result is not true in general if p is not a prime; e.g.: $\binom{4}{2} = 6 = 2 \pmod{4} \neq 0$.

6. We have intercepted a secret message from the enemy encrypted with the RSA algorithm. The encoding is $A = 01, B = 02, C = 03, \ldots, Z = 26$, and the public encryption key used by the enemy is $(n, e) = (31732031, 29553281)$. For instance, the word DOG would be encrypted as follows. First it is encoded: $DOG = 041507$. Then it is encrypted: $041507^{29553281} \bmod 31732031 = 5470487$. The encrypted message that we have intercepted is $m' = m^e \bmod 31732031 = 18091168$. Your mission (should you choose to accept it) is to decrypt (and decode) the message. In order to do that, use the following steps (you will need some computer algebra system such as Maple to do the computations):[2]
   (1) Find the prime factors p and q of $n = 31732031$.
   (2) Find $\phi(n)$.
   (3) Find $d = e^{-1} \bmod \phi(n)$. The decryption key is (n, d).
   (4) Decrypt the secret message by computing $m = {m'}^d \bmod n$.[3]

[2] In Maple you may use "`ifactor`" to find the prime factors of n, and "`mod`" for computations modulo n. Remember to use the ampersand in expressions of the form "`a&^b mod n`" in Maple (so that Maple uses the efficient algorithm to compute powers modulo n.) Basically, the computations to be performed are: find p and q with "`ifactor(n)`", find $\phi(n) =$ "`(p-1)*(q-1)`", find $d =$ "`e&^(-1) mod` $\phi(n)$", find $m =$ "`m'&^d mod` $\phi(n)$". Finally, decode m.

[3] Note that you managed to decode the message because you were able to factor n—so if you want to develop a secure implementation of the RSA algorithm you must choose an n that is hard to factor; in particular the prime numbers p and q should be very large.