Number Theory

Math 522, Fall 2005
Professor: Mike O'Sullivan

Computational Exercises: Based on Rosen, *Elementary Number Theory and Its Applications* 5th ed.

A Maple worksheet with computational experiments is required in this course. There are lots of suggestions for computational explorations and programming projects in Rosen's book. I have elaborated on several in the list below.

There is no prescribed set of exercises to do. You may do anything that interests you. Your work should show a spirit of curiosity and inquiry! Your computer code should be well organized, with commentary. I will meet with you for 15-20 minutes near the end of the course to go over your project. You should be able to explain your work.

The computer project is worth 100 points, with 20 points each for mathematical content, your ability to explain the work, quality of presentation, originality/creativity, difficulty of mathematics.

1. §1.2 PP # 1 The tower of Hanoi puzzle.

2. §1.3 Fibonacci numbers and their ilk. Given g_1, g_2, a and b:
   a) Generate the sequence defined by $g_n = ag_{n-1} + bg_{n-2}$.
   b) Find the explicit definition of g_n as a function of n.
   c) Check that the explicit definition agrees with the recursive definition.

3. §2.1,2 Base b representations:
   a) Convert from base b to base 10 and vice-versa.
   b) Convert from base b to base b^r.

4. §3.1 Prime numbers:
   a) PP #4 Test Goldbach's conjecture.
   b) CE #5 3 Compute twin primes.
   c) Use `nextprime[]` to compute $p/\ln p$ for the first 1000 primes.
   d) Graph $(n, p_n/\ln p_n)$ where p_n is the nth prime. Explain your result.

5. §3.3 The Euclidean Algorithm: Given a, b
   a)Find the greatest common divisor of a and b.
   b) Write the greatest common divisor as a linear combination of a and b using the Euclidean algorithm and report the number of steps it takes.
   c) Compare your results with Lamés Theorem.
   d) Compare your linear combination with what should be obtained according to the Theorem given in class.
   e) Write the greatest common divisor as a linear combination of a and b using the least remainder algorithm and report the number of steps it takes. Compare with the Euclidean algorithm.
   f) Extend these algorithms to find the g.c.d. of $a_1, \ldots, a_r$.

6. §3.4 Unique factorization:
   a) CE #2 Compare the number of primes less than n which are $1 \mod 4$ with the number which are $3 \mod 4$.

b) Extend this to primes of the form $b \mod m$.
c) CE #3 Find the smallest prime congruent to $b \mod m$.
d) PP #2,3 Find the g.c.d. and l.c.m. of a, b from their prime factorizations. Extend to $a_1, \ldots, a_r$.
e) PP #1 List all of the divisors of n from its prime factorization.
f) PP #1 Find the number of divisors of n from its prime factorization.

7. §3.6 Linear Diophantine Equations:
a) PP #1 Find the solutions of a linear diophantine equation in 2 variables.
b) PP #2 Find the positive solutions.
c) CE #1 For given a, b find all linear combinations $ax + by$ with x and y nonnegative.

8. §4.1,2 Modular arithmetic:
a) PP §1#4 Experiment with efficient ways to perform modular exponentiation.
b) PP §2#3 Compute inverses $\mod n$.
c) PP §2#1,2 Solve linear congruences $\mod n$.

9. §4.3 The Chinese remainder theorem:
a) Solve systems of congruences with coprime moduli using the Chinese remainder theorem.
b) Now try it when the moduli are not coprime.

10. §4.5 Systems of linear congruences.
a) Invert a 2×2 matrix over $\mathbb{Z}/n$.
b) Solve a system of congruences over $\mathbb{Z}/n$.
c) Extend to systems in n equations and n unknowns.

11. §5.3 Tournaments.
a) Schedule round-robin tournaments for n teams.
b) Assign a home team for each game in the case where n is odd.

12. §5.4 Hash functions:
a) Write a hashing function for Social Security numbers for m students and $n > m$ memory locations.
b) Experiment with your hashing function. How large should n/m be to make it rare for there to an instance where more than three probes are necessary for a success.

13. §5.5 Coding
a) PP #1 Compute the parity bit for a bit string. Check whether an encoded string has an even or odd number of errors.
b) PP #2 Compute the check digit for and ISBN number. Check whether an ISBN number has an error.
c) Ex. #21 Implement the computation of check digits, and the correction of a single error.

14. Ch. 8 Cryptography:
a) Encrypt and decrypt using an affine transformation modulo n.
b) Encrypt and decrypt using an affine matrix transformation modulo n (a Hill cipher).
c) Encrypt and decrypt using an exponentiation cipher.
d) Encrypt and decrypt using the RSA cryptosystem.