NAME : ______________________________

Math 25

December 3, 2011 - December 6, 2011
Final

Instructions: This is an open book exam (see below for specific rules). You are not to provide or receive help from any outside source during the exam. You may email the instructor for clarification on questions. The exam has no time limit.

- Print out this exam, and solve the written problems on your printout.
- ***Print*** your name in the space provided.
- You must show your work and justify your solutions to receive full credit. Work that is illegible may not be graded; work that is scratched out will not be graded, even if it is correct.
- There are six questions on this exam. Four are written problems, and two are programming problems. When you want to turn in the written problems, staple the entire contents of the exam together, and submit them to Kemeny 316. The programming assignments are turned in at the usual place, and should be titled [your lastname]5.py. Both parts of the exam are due by 1:00pm on Tuesday afternoon.
- This is an open book exam. The following materials, and only the following materials, are allowed: the course textbook, notes from this class (either from the website or your own), your old homework assignments from this class, solutions to old homework assignments in this class, and the official online Python documentation. In particular, you are not allowed to use anything on the Internet, besides the Python documentation, once you start this exam.
- You may use the Python interpreter and any modules included with the Python standard library as a computational assistant during this exam. You may also use any programs you wrote during the course of this class. On the other hand, you may not use other computational programs (such as Wolfram Alpha, Mathematica, Maple, etc.), nor may you use any Python extensions which are not part of the standard library (matplotlib, NumPy).
- Once you open this exam, you are not allowed to discuss the exam with anyone else, either in this class or not in this class, until 1:00pm on Tuesday afternoon.
- See the webpage `http://www.math.dartmouth.edu/~m25f11/final.html` for more detailed rules.

Problem	Points	Score
1	20	
2	20	
3	15	
4	15	
5	15	
6	15	
Total	100	

1. (20) For this problem, you may assume the following fact: let G be a finite abelian group of order p^e, where p is prime. Then G is isomorphic to a group of the form $\mathbb{Z}/p^{e_1}\mathbb{Z} \times \ldots \times \mathbb{Z}/p^{e_r}\mathbb{Z}$, where the $e_1, \ldots, e_r$ are positive integers satisfying $e_1 + \ldots + e_r = e$. (If you know the fundamental theorem of finitely generated abelian groups, you are not allowed to cite that theorem on this problem. You must solve the problems using techniques covered in class or the course textbook.)

(a) (10) Show that $\mathbb{Z}/p\mathbb{Z} \times \mathbb{Z}/p\mathbb{Z}$ and $\mathbb{Z}/p^2\mathbb{Z}$ are not isomorphic. Conclude that any abelian group G of order p^2 is isomorphic to exactly one of $\mathbb{Z}/p\mathbb{Z} \times \mathbb{Z}/p\mathbb{Z}$ or $\mathbb{Z}/p^2\mathbb{Z}$.

(b) (10) Find a list of abelian groups of order p^3 such that (1) every abelian group G of order p^3 is isomorphic to one of the groups on the list, and (2) no two different groups on the list are isomorphic to each other. Prove that your answer is correct.

2. (20) Let n be a positive integer. The *Carmichael lambda function* is defined to be the function λ whose value at n is the smallest positive integer $m = \lambda(n)$ such that

$$a^m \equiv 1 \bmod n,$$

for all integers a with $\gcd(a, n) = 1$. For example, $\lambda(5) = 4$, because $a^4 \equiv 1 \bmod 5$ for all a with $\gcd(a, 5) = 1$ (by Fermat's Little Theorem), and because there actually exists an a such that $a^4 \equiv 1 \bmod 5$, but $a^k \not\equiv 1 \bmod 5$ for $1 \leq k < 4$.

(a) (10) Compute $\lambda(n)$ for $n = p$, where p is prime, and for $n = 12, 16$. Your work should justify why your answers are correct.

(b) (10) Show that $\lambda(n) \mid \phi(n)$, where $\phi(n)$ is the Euler totient function. (You can get partial credit if you show that $\lambda(n) \leq \phi(n)$).

3. (15) Classify all odd primes p such that both 2 and 5 are quadratic residues mod p. Your answer should be in the form $p \equiv a_1, \ldots, a_r \bmod n$, for various integers $a_1, \ldots, a_r, n$. (For example, if this question had asked you to classify all primes p such that 2 is a quadratic residue mod p, your answer would be primes $p \equiv 1, 7 \bmod 8$.)

4. (15) Find a primitive root g mod $66049 = 257^2$, and prove that your answer is correct. You can use a computer to perform arithmetic calculations, but your solution below should list every calculation you did and why your calculations prove that your answer is correct. If you want, you can use the function you program for question 5 on this problem.

The following two questions are programming questions. Submit them at `https://www.math.dartmouth.edu/~m25f11/dropbox/index.phtml`, with your file named [your lastname]5.py. We strongly suggest that you use the provided template file, located at `http://www.math.dartmouth.edu/~m25f11/Programming/template5.py`, which contains test cases and correctly named functions. Your file should run when you select "run module" under IDLE; submissions which require editing to run correctly will lose points. If you want to use functions you have previously written, you MUST INCLUDE THEM in your file.

5. (15) Write a function, order(a, n), which computes the multiplicative order of a mod n. You may assume that gcd$(a, n) = 1$. Your function will be tested on inputs of size $1 \leq a < n, 1 < n \leq 10^6$. (Hint: the ranges on a, n are small enough so that you don't have to do anything fancy.)

6. (15) Write a function, sigmatwo(n), which computes the sum of the squares of all positive divisors of n (including both 1 and n). For example, since 4 has factors $1, 2, 4$, sigmatwo(4) $= 4^2 + 2^2 + 1^2 = 21$. You will receive 8 points if your function works for $1 \leq n \leq 10^7$, and will receive 7 more points if your function works for $1 \leq n \leq 10^{12}$, and also for larger n which are easy to factor (like numbers which are (small) products of high powers of small primes, like $2^{100} \cdot 5^{35}$). The smaller range is small enough that the most obvious algorithm will work, but the larger range will require a bit of number theory. (Hint: the beginning of Chapter 8 in the textbook and your factorization algorithm might be useful.)