This guide will follow in a similar vein as the previous guide. As with the previous exam, material will be divided between "proofs" (all taken from problems on this sheet) and more explicit examples. Congruences are the largest part of this exam, so by the nature of the material there will be more explicit examples with actual "numbers". The exam will not be cumulative, but will only cover what we have done since the first exam.

§1.4: In this section, you are expected to know the basic definitions and notation related to congruences and congruence classes, and it is especially important to have a good intuitive understanding of how to manipulate them. Recall that we showed that congruence modulo \( n \) is an equivalence relation, and the corresponding equivalence classes are called congruence classes. We denote the set of \( n \) congruence classes by \( \mathbb{Z}_n \), and we usually choose as representatives the numbers \( 0, 1, \ldots, n - 1 \). We also learned that \( \mathbb{Z}_n \) has a sum and product structure, meaning that we can add or multiply two congruence classes in the natural way by taking the class of the sum or product of the representatives, and that this is a well-defined operation. As we saw, one must be careful with division, however, as there exist elements called zero divisors which mess up the ability to divide congruence equations by a common number. We then learned that \( a \) is invertible modulo \( n \) if and only if it is relatively prime to \( n \), in which case we can divide equations mod \( n \) by \( a \), and otherwise that \( a \) is a zero-divisor. We also saw that the inverses of these relatively prime \( a \)'s may be computed via the Euclidean algorithm. Finally, we denoted by \( \mathbb{Z}_n^\times \) the set of invertible classes (note that the book calls it \( G_n \)), and saw that this set is closed under multiplication (note that it isn’t closed under addition). Here are a few problems from this section.

1. Recall the proof of Theorem 1.4.1, which we used to show that the “natural” definitions of sums and products of congruences classes are well-defined. The proof essentially followed by writing down what it means for two numbers to be congruent (i.e., giving an explicit name for the multiple of \( n \) which is the difference of the two) and expanding.

2. Show Theorem 1.4.3, which says that \([a] \in \mathbb{Z}_n^\times \iff (a, n) = 1\), and gives us a procedure to compute \([a]^{-1}\) when \((a, n) = 1\) (the Euclidean algorithm). Both parts of this proof followed by our frequently used fact that \((a, n) = 1\) if and only if there are integers \( k \) and \( t \) such that \( ak + nt = 1\).

3. Know how to use Corollary 1.4.4, which states that if \((a, n) = 1\), then we can cancel factors of \( a \) out from both sides of modulo \( n \) equations.
(4) Compute some of the modular inverses from Exercise 3 from the book again. Sometimes you can guess the answer by trying a few values, but in general it can be done using the Euclidean algorithm.

(5) Compute $\mathbb{Z}_n^*$ for a few $n$, as in Exercise 4.

(6) Recall problems 5-7 from the book. Hints: the extra exercise on the course website from that section is a hint for problem 5, problem 6 requires factoring the polynomial $x^2 - 1$ and using the facts mentioned above, and problem 6 is used in the solution of problem 7. We also sketched the solutions of 6-7 in class, so you can also review your notes from that day.

§1.5: In this section, we learned how to solve linear congruences, that is congruences of the form $ax \equiv b \pmod{n}$ for fixed $a, b, n$, as well as systems of several linear congruences. For this section, your goal is to be able to efficiently solve congruences which I give to you explicitly, so make sure you fully understand all of the exercises from this section and feel comfortable with any such set of congruences. Recall that the procedure for solving linear congruences is nicely summarized on page 52 (or in your class notes). To solve two simultaneous congruences, recall that one must first either by guess-and-check or by the Euclidean algorithm determine a combination of the two moduli which gives 1, and then multiply the summands in these terms by the right hand sides of the equations to get the unique solution modulo the product of the moduli. Be careful, however, in recalling that this only works when the moduli are relatively prime (simple counterexample when they aren’t relatively prime: we obviously can’t solve $x \equiv 1 \pmod{5}$, $x \equiv 2 \pmod{5}$ at the same time), and that when a congruence is not of the form $x \equiv b \pmod{n}$ but of the more general shape $ax \equiv b \pmod{n}$, we first reduce it to the first form using the methods for solving single linear congruences (you should understand why this will give you either that there is no solution if $(a, n) \nmid b$, or else that there is a unique solution modulo $n/(a, n)$). Finally, if there are more than two equations which must be simultaneously solved, we simply solve them two at a time.

§1.6: In this section, we learned about (multiplicative) orders of integers modulo $n$. Specifically, we learned that the order is finite iff it is relatively prime to $n$, and we learned a la Euler’s theorem that the order divides $\varphi(n)$. Be sure that you know how to compute the Euler phi function for general numbers $n$ (first write $n$ as a product of prime powers, then use Theorems 1.6.5 and 1.6.6). Be sure you feel comfortable using Euler’s theorem, as well as its special corollary Fermat’s Little Theorem. Finally, you should recall how the RSA encoding/decoding procedure works (but you don’t have to prove how it works). Here are a few proofs and problems to review.

(1) Prove Euler’s Theorem (Theorem 1.6.7).
(2) Prove Theorem 1.6.5. Recall that this is simply a counting argument whereby you look at a set of representatives of the congruences classes modulo $p^n$ and list all which are divisible by $p$ (which is the same as not relatively prime to $p$) and throw them out.
(3) Prove Theorem 1.6.6.
(4) Revisit problems 1,2,3,5,12 from the book.

§4.1: Here, we learned about the symmetric group $S(n)$, which is the set of all permutations on $n$ elements, that is, the set of all bijections on the set $\{1, 2, \ldots, n\}$. We saw in Theorem 4.1.1 that composition of these bijections satisfies properties somewhat reminiscent of addition of real numbers, integers, or congruence classes (we will later see that these are all examples of groups). You should be comfortable interpreting the two-row equation for these permutations, and in computing the compositions and inverses of them. We also learned about cycles and transpositions (which are 2-cycles). You should also be able to interpret cycle notation, and given a composition of cycles rewrite it as a permutation in the two-row notation. You should also be adept at computing the cycle decompositions of arbitrary permutations. Consider the following problems.

(1) Prove Theorem 4.1.2, which states that disjoint permutations (which don’t move any of the same elements) commute.

(2) Prove Theorem 4.1.3, which says that any permutation has a unique cycle decomposition (up to rearranging the order of the factors).

(3) Redo problems 1-4. If you feel that you are not fully comfortable at doing these problems, or that they took you some time, try also inventing similar problems by writing down some arbitrary permutations and practice computing these operations until you are fluent.

§4.2: Here we learned about orders of permutations (which are like the concept of orders we saw in modular arithmetic) and about the sign or parity of permutations. Unlike in the situation of modular arithmetic, we saw that all permutations have a finite order. We also saw how to compute orders using the cycle decomposition. Make sure you can compute this order in any numerical example. We also learned about the sign of permutations, which can be described in several ways, the simplest being that it is given by the parity of the number of transpositions in a decomposition (though we had to show that such decompositions always exist and that although they aren’t unique, the parity of the number of transpositions is).

(1) Show Theorem 4.2.2, which states that permutations all have finite orders (which follows from two facts: that $S(n)$ is finite, and that all elements of $S(n)$ are invertible).

(2) Show Theorem 4.2.4, which states that the length of a cycle is the same as its order (if you forget why this is true, just write it down in a few examples and you should then see why it should be true).

(3) Show Theorem 4.2.8, which states that the sign of a composition of permutations is the product of their signs.

(4) Prove Theorems 4.2.10 and 4.2.11, which claim that arbitrary permutations are products of transpositions, and that the parity of the number of transpositions is fixed.

(5) Review problems 1,2,3,4,6,7,9,10 from the book.