

**Chapter 2:** 14, 22, 24, 25, 26, 28.

**Due date:** Friday, 9/12

- 14.** Show that the Principle of Well-Ordering for the natural numbers implies that 1 is the smallest natural number. Use this result to show that the Principle of Well-Ordering implies the Principle of Mathematical Induction; that is, show that if  $S \subseteq \mathbb{N}$  such that  $1 \in S$  and  $n + 1 \in S$  whenever  $n \in S$ , then  $S = \mathbb{N}$ .
- 22.** Let  $n \in \mathbb{N}$ . Use the division algorithm to prove that every integer is congruent mod  $n$  to precisely one of the integers  $0, 1, \dots, n - 1$ . Conclude that if  $r$  is an integer, then there is exactly one  $s$  in  $\mathbb{Z}$  such that  $0 \leq s < n$  and  $[r] = [s]$ . Hence, the integers are indeed partitioned by congruence mod  $n$ .
- 23.** Define the *least common multiple* of two nonzero integers  $a$  and  $b$ , denoted by  $\text{lcm}(a, b)$ , to be the nonnegative integer  $m$  such that both  $a$  and  $b$  divide  $m$ , and if  $a$  and  $b$  divide any other integer  $n$ , then  $m$  also divides  $n$ . Prove that any two integers  $a$  and  $b$  have a unique least common multiple.

N.B. Exercise 23 is not required. It is included here for you reference, since it defines *least common multiple*.

- 24.** If  $d = \text{gcd}(a, b)$  and  $m = \text{lcm}(a, b)$ , prove that  $dm = |ab|$ .
- 25.** Show that  $\text{lcm}(a, b) = ab$  if and only if  $\text{gcd}(a, b) = 1$ .
- 26.** Prove that  $\text{gcd}(a, c) = \text{gcd}(b, c) = 1$  if and only if  $\text{gcd}(ab, c) = 1$  for integers  $a$ ,  $b$ , and  $c$ .

N.B. The following problem (Exercise 27) is not required. It is included here for you reference, since it may be useful when solving 26 (depending on the proof strategy you use). You may use the result stated in Exercise 27, even if you have not yet proved it.

- 27.** Let  $a, b, c \in \mathbb{Z}$ . Prove that if  $\text{gcd}(a, b) = 1$  and  $a \mid bc$ , then  $a \mid c$ .
- 28.** Let  $p \geq 2$ . Prove that if  $2^p - 1$  is prime, then  $p$  must also be prime.

**Optional Programming Exercise.** If you wish to earn a bit of extra credit, you are encouraged to try the following:

- (1) (1/4) Sign up for a Sage account at [sagemath.org](http://www.sagemath.org).
- (2) (1/4) Create a new project named "Math 301" and then create a Sage Worksheet within that project. Name your worksheet "Eratosthenes" or "Ackermann" or "DivisionAlgorithm" (depending on which problem you plan to solve in the next part).
- (3) (1/2) In the worksheet you created above, solve one of the three programming exercises described on page 31 of our textbook. These are also included below for your reference.

To get credit for this assignment, when you have finished, you must either send me your .sagews file by email, or invite me to be a collaborator on your Math 301 project. You must also email me a few sentences describing your experience and your first impressions of Sage.

### More notes on Sage Assignment 1

The number in parentheses indicates the fraction of the percentage point that each part is worth. The following are some notes about each part of the assignment:

- (1) Go to [cloud.sagemath.org](http://cloud.sagemath.org/).
- (2) First create a new project by clicking the link that says "New Project." Name the project "Math 301," then click the Math 301 link that appears. Wait for your new project to load, then either click "New" or "Create or Import a File, Worksheet..." In the box where it says, "Name your file," enter the name "Eratosthenes" or "Ackermann" or "DivisionAlgorithm" (depending on which problem you plan to solve in the next part). Then, where it says "Select the type," choose "Sage Worksheet."
- (3) You are to solve one of the three problems described on [page 35 of our textbook](https://github.com/williamallan2014/blob/master/homework/pdf/SageAssignment1.pdf).

There is plenty of online documentation for Sage. If you need help, please ask! Also, you can find some nice examples of Sage programs that are specifically related to our course at <http://abstract.ups.edu/sage-aata.html>.

### Programming Exercises.

- (1) **The Sieve of Eratosthenes.** One method of computing all of the prime numbers less than a certain fixed positive integer  $N$  is to list all of the numbers  $n$  such that  $1 < n < N$ . Begin by eliminating all of the multiples of 2. Next eliminate all of the multiples of 3. Now eliminate all of the multiples of 5. Notice that 4 has already been crossed out. Continue in this manner, noticing that we do not have to go all the way to  $N$ ; it suffices to stop at  $\sqrt{N}$ . Using this method, compute all of the prime numbers less than  $N = 250$ . We can also use this method to find all of the integers that are relatively prime to an integer  $N$ . Simply eliminate the prime factors of  $N$  and all of their multiples. Using this method, find all of the numbers that are relatively prime to  $N = 120$ . Using the Sieve of Eratosthenes, write a program that will compute all of the primes less than an integer  $N$ .

- (2) Let  $\mathbb{N}^0 = \mathbb{N} \cup \{0\}$ . Ackermann's function is the function  $A : \mathbb{N}^0 \times \mathbb{N}^0 \rightarrow \mathbb{N}^0$  defined by the equations

$$A(0, y) = y + 1,$$

$$A(x + 1, 0) = A(x, 1),$$

$$A(x + 1, y + 1) = A(x, A(x + 1, y)).$$

Use this definition to compute  $A(3, 1)$ . Write a program to evaluate Ackermann's function. Modify the program to count the number of statements executed in the program when Ackermann's function is evaluated. How many statements are executed in the evaluation of  $A(4, 1)$ ? What about  $A(5, 1)$ ?

- (3) Write a computer program that will implement the Euclidean algorithm. The program should accept two positive integers  $a$  and  $b$  as input and should output  $\gcd(a, b)$  as well as integers  $r$  and  $s$  such that

$$\gcd(a, b) = ra + sb.$$