Written problems are due at the beginning of class on Friday, September 18. Programming problems must be electronically submitted by 10:50am according to the instructions at the end of this document.

Written problems

1. The following passage has been encrypted using Caesar encryption. That is, there is a secret key k, and each letter of the plaintext has been advanced k places in the alphabet (where the letter Z advanced to the letter A). All punctuation and spaces have been left unchanged.

```
Nzky kyv yvcg fw kyv arezkfi yv jtivnvu fekf kyv
jzuv fw kyv uvjb r gvetzc jyrigvevi -- kyrk yzxycp
jrkzjwpzex, yzxycp gyzcfjfgyztrc zdgcvdvek kyrk xfvj
kztfeuvifxr-kztfeuvifxr, wvvuzex fe kyv pvccfn wzezjy reu
jnvvk nffu, reu veuj lg ze r bzeu fw jfleucvjjcp jgzeezex
vkyvivrc mfzu rj nv rcc dljk.
```

   Determine the secret key k. (You can solve this by hand, but I suggest trying to write some code to make your job easier. If you can fully automate the process, you may submit your work for extra credit; see the last problem on this problem set.)

2. Throughout this course, we will say that an integer a is an "n-bit (nonnegative) integer" if $0 \leq a < 2^n$. This means that it can be written in binary with at most n symbols; it is a convenient shorthand to specify the size of a number, such as a cryptographic key. The number of bits in a key is a rough guide to its strength.

   (a) The secret key of a Caesar cipher is a number k from 1 to 25 inclusive. How many bits are needed to specify this secret key?

   (b) The *Data Encryption Standard* (DES) is a private-key encryption algorithm that was a government standard from 1977 to 2002. DES uses 56-bit secret keys. Suppose that Eve attempts a brute-force attack on DES by trying to decrypt an intercepted cipher text with every possible 56-bit key until she finds something that looks like English text. If Eve's system can try 1 billion keys per second, how long would it take her to try all of the keys (and thus be sure to break the encryption)?

   (By 1999, a distributed system was able to break DES encryption in less than 24 hours. DES was replaced in 2002 by a new standard, called AES, which uses keys of at least 128 bits. For "top secret" communication, the government uses AES with 256 bit keys.)

   (c) A stronger (but still weak) version of the Caesar cipher is a *substitution cipher,* as described in §1.1. A secret key for a substitution cipher consists of a permutation of the letters from A to Z (see the book for some examples). How many such secret keys are there? How many bits would be required to specify such a key?

   (Although substitution ciphers have fairly long keys, they are still completely insecure; the textbook describes how they can be broken using methods much more efficient than brute force search.)

3. Textbook exercise 1.9. (same in first edition). If you wish, you may complete problem 7 first and run your code to answer this question.

4. Textbook exercise 1.10 (same in first edition).

5. Textbook exercise 1.11 parts (a) and (b) (same in first edition).

Programming problems

You do not need to submit anything written for the following problems. Instead, you will write your code (in a program language of your choice, although I recommend Python 2 for this course) and submit it online according to the instructions at the end of this document. Your program must be fast enough to finish the specified task in less than 10 seconds (this should not be an obstacle for this assignment, but will be important later).

6. Write a program which can factor a 16-bit integer into primes. More precisely, the program should take an integer n such that $2 \leq n < 2^{16}$, and print out the prime factors of n, in order, printing each prime the same number of times as it occurs in the factorization of n. For example, if the program reads "12", it should print "2 2 3."

   (Later in the course, you will write programs that can factor integers much longer than 16 bits.)

7. Write a program which takes a list of 1024-bit positive integers (i.e. each integer a in the list satisfies $1 \leq a < 2^{1024}$) and prints their greatest common divisor.

   Suggestion. First write a function to compute the greatest common divisor of two positive integers (we will discuss a classic, very efficient, algorithm for this in class), then figure out how to use this function to find the GCD of a longer list.

8. (*Extra credit*). Write a program which can solve problem 1 of this assignment automatically (i.e. decrypt a passage of English text encrypted with a Caesar cipher). (I am happy to provide hints for how to proceed).

Programming problem submission instructions

We will use a platform called hackerrank in this course. The platform is designed for programming contests, so unfortunately for our purposes the site will refer to the problem sets as "contests," and to you as "competitors." The problem set is not a contest. However, the features of the platform are so well-suited to our needs that I hope you will forgive and ignore this vocabulary. Hackerrank does offer a "for schools" platform but I have chosen not to use it because it lacks several useful features.

- Create an account on the website `hackerrank.com`. *Choose an anonymous-looking username* for your account; all other members of the course will be able to view your scores and read the code you submit, so this will keep everyone anonymous. However, please email your username to me (`pflueger@math.brown.edu`) so that I will know who is who.
- Visit `https://www.hackerrank.com/math158-pset-1` and enter the "contest." You will find the three problems listed above.

- You may submit code for each problem as many times as you like. After each submission, the website will run your code on a batch of test cases, and give you a score equal to the number of test cases your code completes successfully. You will also be able to view the input (and your code's output) for a small batch of sample test cases, which will help you debug. There is no penalty for multiple submissions, however you will not be able to view the content of the incorrect test cases until after the due date (this is to prevent you from being able to hard-code the individual answers).
- At 10:50am on the due date of the assignment, the assignment will stop accepting submissions for credit. At this time, all test cases will become viewable, and also all submitted code from the other students in the class will be viewable. You are encouraged to read the other solutions and learn from them. You will also be able to make more code submissions and view what your score would have been, although these will no longer count for credit.

Additional notes

- If you are new to programming, or new to Python, there are a number of good tutorials online. I will link to some on the main course webpage. If you discover any resources on your own that you find particularly useful, I would like to hear about it and may add a link on the course page.
- I will also spend some time showing how to do basic tasks in Python on Monday the 14th (if you have prior programming experience, you need not attend class on Monday).
- On Monday I will also work through the following sample problem set.

  `https://www.hackerrank.com/contests/math158-demonstration-pset`

  I encourage you to bring a computer to class to follow along. You may also submit solutions for this sample problem set in order to practice using the platform. You may also want to look at the problems on hackerrank's "warmup contest", which has example code in every language supported by the platform.

  `https://www.hackerrank.com/domains/algorithms/warmup`