Week 3

Question 1. This weekend, Rowan tried to bake a chocolate cake! The recipe he was using said he needed exactly 1 cup of milk; however, because he's a poor college student and doesn't like cooking, he only owns two measuring jugs — one that holds exactly 5 cups, and one that holds 12 cups. What's more, the measurement markings on the sides have all rubbed off, so he can't measure anything with them except by checking whether they're full up or not.

Is it possible for Rowan to use these two jugs to measure exactly 1 cup of milk, by pouring milk back and forth between them in some order?

Question 2. Alex is better at cooking than Rowan, so he owns *three* jugs: one that holds 10 cups, one that holds 15 cups, and one that holds 20 cups. Can he measure out exactly 1 cup of milk using these jugs? If not, why not?

When you've answered these questions, turn to the next page...

This week, we'll investigate *common factors*.

How can we check whether some numbers have a common factor? I know of two ways:

1. List out all the numbers starting from 1, and check whether your particular numbers are divisible by them. This isn't too hard for small numbers like 5 and 12, but for big numbers like 20966437751, it takes *ages*.
2. Or, use the Euclidean Algorithm!

"What's the Euclidean Algorithm?", you're probably asking. Let me explain!

Suppose you want to find a common factor of 375 and 135. *(Do you see any common factors already?)* Start by putting them in a table:

375	
135	

In the next column, move the 135 up to the top row, and in the spot beneath it, put the remainder when you divide 375 by 135 (which is 105, since $375 = 2 \times 135 + 105$):

375	135	
135	105	

Keep going like this — for each new column, move the lower entry from the previous column to the top, then below it put the remainder when dividing the top entry by the bottom one.

375	135	105	30	15
135	105	30	15	0

And stop when you get a 0. The number above the 0 is a 15, so that's the greatest common factor between 375 and 135 ! *(Is this right? Check that it is really a common factor!)* So the common factors are 15 and everything else that 15 is divisible by, i.e. 1, 3 and 5.

I hope you agree that this is a lot quicker than checking every number to see whether it divides into 375 and 135...

As a reminder, to follow the Euclidean algorithm, if one column of your table contains a and b, the next column should contain b and $a \bmod b$ (where "$a \bmod b$" is shorthand for "the remainder when you divide a by b").

	a	b	
	b	$a \bmod b$	

Question 3. Use the Euclidean algorithm to find a common factor of:

(a) 12 and 5,

(b) 20 and 15,

(c) 1024 and 48,

(d) 4749 and 3687,

(e) the 11th and 10th Fibonacci numbers (89 and 55),

(f) your favourite two numbers.

Question 4. *(Challenge.)* You might be wondering "why does the Euclidean algorithm work?" The trick is that whenever you add a new column in the table, the common factors of the new numbers don't change.

(a) Check that this works in one of the tables you made above.

(b) Why is this fact always true? In other words, if d is a factor of both a and b, why is it also a factor of both b and $a \bmod b \ldots$?

(c) $\ldots$ and if e is a factor of both b and $a \bmod b$, why is it also a factor of a and b?

(d) Does the algorithm always eventually hit 0 and stop? Is it possible to never get a 0?

(e) Why do we stop when we get to 0? What does it mean for $a \bmod b$ to equal 0?

We can extend the Euclidean algorithm to do more than just calculate common factors!

Remember from last week: two numbers x and y are called "multiplicative inverses modulo q" if $x \times y$ has a remainder of 1 when you divide by m. We noticed a pattern — a number x has a multiplicative inverse modulo q if the only common factor of x and q is 1. Let's see why!

Question 5. First, what if x and q have a common factor d that isn't 1? For x to have a multiplicative inverse y, then $x \times y$ must be 1 more than a multiple of q — that is, $x \times y = n \times q + 1$. Explain why this equation is impossible if d is a factor of both x and q.

(Hint: Is the left hand side of the equation divisible by d? What about the right hand side?)

OK, now for the fun part — the extended Euclidean algorithm!

For example, here's a table for the greatest common factor of 319 and 61:

319	61	14	5	4	1
61	14	5	4	1	0

For the extended version of the algorithm, we're going to add some more rows. First, for each column, calculate the top number divided by the bottom number, round this number **downwards** to the nearest integer, and write the result below. Don't worry about the column with the 0. For example, using the table for 319 and 61,

319	61	14	5	4	1
61	14	5	4	1	0
5	4	2	1	4	

(For example, the first number in the new row is 5, because $319 \div 61$ is $5.2295\cdots$.)

Next, add a 0 and a 1 like this:

	319	61	14	5	4	1
	61	14	5	4	1	0
	5	4	2	1	4	
0	1					

Then fill in the rest of this row so that each new cell is

(cell two places to the left) − (cell to the left) × (cell above left).

Pictorially, the rule looks like this:

	b	
a	c	$a - b \times c$

For example, in the table above, the first empty cell should be $0 - 5 \times 1 = -5$. The next empty cell is $1 - 4 \times (-5) = 21$, and so on. Once the whole row is filled, our example table looks like this:

	319	61	14	5	4	1
	61	14	5	4	1	0
	5	4	2	1	4	
0	1	−5	21	−47	68	

Now, look at the last number in this row — in the example, it's 68. If the numbers we started with are q and x, and if their only common factor is 1, this number should be the multiplicative inverse of x modulo q! For example, 68×61 is 1 modulo 319.

Question 6. Use this algorithm to compute multiplicative inverses of:

(a) 5 modulo 17,

(b) 7 modulo 10,

(c) 19 modulo 125,

(d) 121 modulo 731,

(e) the 10th Fibonacci number modulo the 11th (55 and 89),

(f) the 9th Fibonacci number modulo the 10th (34 and 55),

(g) your favourite number modulo your least favourite number. *Check that their only common factor is* 1 *first!*

Check that your answers are right, by multiplying x with its multiplicative inverse and finding the remainder modulo q: it should be 1.

Question 7. *(Challenge.)* Why does this algorithm work?