Verifying barcodes

* Universal product codes (UPC for short) are the little barcodes you find on basically any packaged item.
* When we view barcodes, we interpret them as thin or thick bars and wide or narrow gaps
	+ The bars have four different widths:
		- Thicker bars being 2, 3, or 4 times the width of the thinnest bar
	+ Similarly, the gaps also have 4 different widths:
		- Wider gaps being 2, 3, or 4 times the width of the thinnest gap
* Although they may not look like it, UPC’s are binary codes with 95 bits

* Each bar is a one while each gap is a 0
	+ The width determines the amount of 0’s or 1’s

Example 1

* To verify a barcode, we must verify the modulo check character
	+ This is the last number in our barcode
* Assign each digit a letter A-L
* A BCDEF GHIJK L
* Now we calculate the following:
	+ - * + (3\*(A + C + E + G + I + K) + (B + D + F + H + J)
				+ (3\*(0 + 6 + 0 + 2 + 1 + 5) + (3 + 0 + 0 + 9 + 4) = 58
* After calculating that we now want to subtract the number we got from the next highest multiple of 10
	+ - * + 60 – 58 = 2

Example 2

0 39402 01700 4

Example 3

0 14113 91002 6

Using R

* R can be used a calculator, so let’s plug in 2+3
* Instead of calculating that every time, we can store the answer in a variable
	+ Use <- to save 2+3 in variable x
	+ After doing this we must tell R to print it out by entering print(x)
* We are also able to store list of numbers. These are called vectors
	+ To store the vector (3, 6, 9) in variable y, we need to use the c() function
	+ y <- c(3, 6, 9)
* Sometimes we want to know the sum of the elements in our variable
	+ Try inputting sum(y)
* Let’s do a sequence
	+ Input seq(1, 12)
	+ Also try seq(1, 12, by=2)

Example 4

Let’s try to solve this barcode on R:

0 36000 29145 2



* After storing the barcode, we want to start the process of solving:
	+ (3\*(A + C + E + G + I + K) + (B + D + F + H + J)



* We want to subtract the number we got from the next highest multiple of 10



Try on R

* 0 98307 10048 9
* 0 17600 02177 9