**crazy\_sum**

Write a method named crazy\_sum(numbers) that takes an array of numbers. crazy\_sum should multiply each number in the array by its position in the array, and return the sum.

crazy\_sum([2]) == 0 # (2\*0) crazy\_sum([2, 3]) == 3 # (2\*0) + (3\*1) crazy\_sum([2, 3, 5]) == 13 # (2\*0) + (3\*1) + (5\*2) crazy\_sum([2, 3, 5, 2]) == 19 # (2\*0) + (3\*1) + (5\*2) + (2\*3)

Make sure your code works for arrays with repeats of the same number (like [2, 3, 5, 2]). **Hint**: be careful about using Array#index; why?

**square\_nums**

Write a method square\_nums that takes a number maxand returns the number of perfect squares less than max. **Do not use Math.sqrt(x) or x \*\* 0.5**; we don't want you to calculate square roots for this problem. You will, however, want to find squares (x \* x is fine).

square\_nums(5) == 2 square\_nums(10) == 3 square\_nums(25) == 4

**crazy\_nums**

Write a method crazy\_nums that takes a number,max, and returns an array of the integers that

* are less than max
* are divisible by either three or five
* are not divisible by *both* three and five

You may wish to use the modulo operation: 5 % 2returns the remainder when dividing 5 by 2: 1. Ifnum is divisible by i, then num % i == 0.

crazy\_nums(3) == [] crazy\_nums(10) == [3, 5, 6, 9] crazy\_nums(20) == [3, 5, 6, 9, 10, 12, 18]