

## Math Challenge #5

There is a special number called *Champernowne's constant*:

**0.123456789101112131415161718192021...**

It is an irrational decimal number formed by placing each integer after the decimal point, one after the other. Above, you can see that so far, the integers:

1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21

have been stuck together after the decimal point.

Let's say that  $d_n$  represents the  $n$ th digit in the decimal part of Champernowne's constant. If we split the first part of Champernowne's constant up into its single digits:

0.	1	2	3	4	5	6	7	8	9	1	0	1	1	1	2	1	3	1	4	1	5
$d_n$	$d_1$				$d_5$					$d_{10}$					$d_{15}$					$d_{20}$	

We can see that  $d_1$  is 1,  $d_5$  is 5,  $d_{10}$  is 1,  $d_{15}$  is 2,  $d_{20}$  is 1, and so on. Before you start writing code, make sure you understand how to find  $d_n$  for any value of  $n$ . You may find it helpful to fill in the rest of the table above for practice.

If you were asked to calculate the following product

$$d_1 \times d_5 \times d_{10} \times d_{15} \times d_{20} = ?$$

it would surely be easy for you. Just by looking at the table above, we can see that

$$d_1 \times d_5 \times d_{10} \times d_{15} \times d_{20} = 1 \times 5 \times 1 \times 2 \times 1 = \mathbf{10}$$

Your actual challenge is to write a Python program that calculates the following product:

$$d_1 \times d_{10} \times d_{100} \times d_{1000} \times d_{10000} \times d_{100000} \times d_{1000000}$$

As always, start simple and take small steps to reach your goal. Don't start by using 1,000,000 for  $n$ ; instead, use 100. First, get your program to print out every digit of Champernowne's constant up to the 100<sup>th</sup> digit. Then, modify your program to only print out the 1<sup>st</sup>, 10<sup>th</sup>, and 100<sup>th</sup> digits. Finally, modify your program to multiply just those three digits together, then print out the final product for the user to see. If you do this correctly, your program will display the correct answer as 5, and you'll be ready to use 1,000,000 for  $n$  to solve this challenge.