**CSE115L Final Exam**

Name: ID:

1. You are given an odd-length array of integers, in which all of them are the same, except for one single number.

Complete the method which accepts such an array, and returns that single different number.

The input array will always be valid! (Odd-length >= 3)

Examples:

[1, 1, 2] ==> 2

[17, 17, 3, 17, 17, 17, 17] ==> 3

1. In this problem, you must create a **digital root** function.

Given n, take the sum of the digits of n. If that value has more than one digit, continue reducing in this way until a single-digit number is produced. This is only applicable to the natural numbers.

Examples:

|  |  |
| --- | --- |
| digital\_root(132189)  => 1 + 3 + 2 + 1 + 8 + 9  => 24 ...  => 2 + 4  => 6 | digital\_root(493193)  => 4 + 9 + 3 + 1 + 9 + 3  => 29 ...  => 2 + 9  => 11 ...  => 1 + 1  => 2 |

1. Count the Digit

Take an integer n (n >= 0) and a digit d (0 <= d <= 9) as an integer. Square all numbers k (0 <= k <= n) between 0 and n. Count the numbers of digits d used in the writing of all the k\*\*2.

Call nb\_dig (or nbDig or ...) the function taking n and d as parameters and returning this count.

#Examples:

n = 10, d = 1, the k\*k are 0, 1, 4, 9, 16, 25, 36, 49, 64, 81, 100

We are using the digit 1 in 1, 16, 81, 100. The total count is then 4.

nb\_dig(25, 1):

The numbers of interest are

1, 4, 9, 10, 11, 12, 13, 14, 19, 21 which squared are 1, 16, 81, 100, 121, 144, 169, 196, 361, 441

So there are 11 digits `1` for the squares of numbers between 0 and 25.

Note that 121 has twice the digit 1.