Recursive Digit Sum

HackerRank

We define super digit of an integer x using the following rules:

Given an integer, we need to find the *super digit* of the integer.

- If x has only 1 digit, then its super digit is x.
- Otherwise, the super digit of x is equal to the super digit of the sum of the digits of x.

For example, the super digit of 9875 will be calculated as:

```
	super_digit(9875)   	9+8+7+5 = 29 
	super_digit(29) 	2 + 9 = 11
	super_digit(11)		1 + 1 = 2
	super_digit(2)		= 2
```

Example
$n ='9875'$
$k = 4$

The number p is created by concatenating the string n k times so the initial $p = 9875987598759875$.

```
    superDigit(p) = superDigit(9875987598759875)
                  9+8+7+5+9+8+7+5+9+8+7+5+9+8+7+5 = 116
    superDigit(p) = superDigit(116)
                  1+1+6 = 8
    superDigit(p) = superDigit(8)
```

All of the digits of p sum to 116. The digits of 116 sum to 8. 8 is only one digit, so it is the super digit.

Function Description

Complete the function *superDigit* in the editor below. It must return the calculated super digit as an integer.

superDigit has the following parameter(s):

- *string n:* a string representation of an integer
- *int k:* the times to concatenate n to make p

Returns

- *int:* the super digit of n repeated k times

Input Format

The first line contains two space separated integers, n and k.

Constraints

- $1 \leq n < 10^{100000}$

- $1 \leq k \leq 10^5$