Recursive Digit Sum

locked

Problem Submissions Leaderboard Discussions

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

- $1 \leq n < 10^{100000}$
- $1 \leq k \leq 10^5$

Sample Input 0

```
148 3
```

Sample Output 0

```
3
```

Explanation 0

Here $n = 148$ and $k = 3$, so $p = 148148148$.

```
super_digit(P) = super_digit(148148148)
               = super_digit(1+4+8+1+4+8+1+4+8)
               = super_digit(39)
               = super_digit(3+9)
               = super_digit(12)
               = super_digit(1+2)
               = super_digit(3)
               = 3
```

Sample Input 1

```
9875 4
```

Sample Output 1

```
8
```

Sample Input 2

```
123 3
```

Sample Output 2

```
9
```

Explanation 2

Here $n = 123$ and $k = 3$, so $p = 123123123$.

```
super_digit(P) = super_digit(123123123)
               = super_digit(1+2+3+1+2+3+1+2+3)
               = super_digit(18)
               = super_digit(1+8)
               = super_digit(9)
               = 9
```

Submissions: 8
Max Score: 30
Difficulty: Medium

Rate This Challenge:

More

Python 3

```
1  import math
2  import os
3  import random
4  import re
5  import sys
6
7  def superDigit(n, k):
8      # Write your code here
9      total = 0
10     if(len(n) == 1):
11         return n
12     else:
13         newList = list(n)
14         newList = list(map(int, n))
15         total = sum(newList) * k
16         return superDigit(str(total), 1)
17
18 if __name__ == '__main__':
19     fptr = open(os.environ['OUTPUT_PATH'], 'w')
20
21     first_multiple_input = input().rstrip().split()
22
23     n = first_multiple_input[0]
24
25     k = int(first_multiple_input[1])
26
27     result = superDigit(n, k)
28
29     fptr.write(str(result) + '\n')
30
31     fptr.close()
```

Line: 31 Col: 17

Upload Code as File Test against custom input Run Code Submit Code

Testcase 0 ✔ | Testcase 1 ✔ | Testcase 2 ✔

Congratulations, you passed the sample test case.

Click the **Submit Code** button to run your code against all the test cases.

Input (stdin)

```
148 3
```

Your Output (stdout)

```
3
```

Expected Output

3