Recursive Digit Sum

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
```

```
import java.util.*;
import java.util.concurrent.*;
import java.util.regex.*;

class Result {

    /*
     * Complete the 'superDigit' function below.
     *
     * The function is expected to return an INTEGER.
     * The function accepts following parameters:
     *  1. STRING n
     *  2. INTEGER k
     */

    public static int superDigit(String n, int k) {
        // Calculate sum of digits of n
        long sum = 0;
        for (int i = 0; i < n.length(); i++) {
            sum += (n.charAt(i) - '0'); // convert char to int value
        }

        // Calculate super digit of the sum
        int superDigit = calculateSuperDigit(sum);

        // Multiply super digit by k and find its super digit
        long multipliedResult = superDigit * k;
        return calculateSuperDigit(multipliedResult);
    }

    private static int calculateSuperDigit(long num) {
        // Calculate super digit of a number (sum of its digits)
        while (num >= 10) {
            long sum = 0;
            while (num > 0) {
                sum += num % 10;
                num /= 10;
            }
            num = sum;
        }
        return (int) num;
    }
}

public class Solution {
    public static void main(String[] args) throws IOException {
        BufferedReader bufferedReader = new BufferedReader(new InputStreamReader(System.in));
        BufferedWriter bufferedWriter = new BufferedWriter(new
FileWriter(System.getenv("OUTPUT_PATH")));

        String[] firstMultipleInput = bufferedReader.readLine().replaceAll("\\s+$", "").split("
");

        String n = firstMultipleInput[0];

        int k = Integer.parseInt(firstMultipleInput[1]);

        int result = Result.superDigit(n, k);

        bufferedWriter.write(String.valueOf(result));
        bufferedWriter.newLine();

        bufferedReader.close();
        bufferedWriter.close();
    }
}

```

Line: 46 Col: 6

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