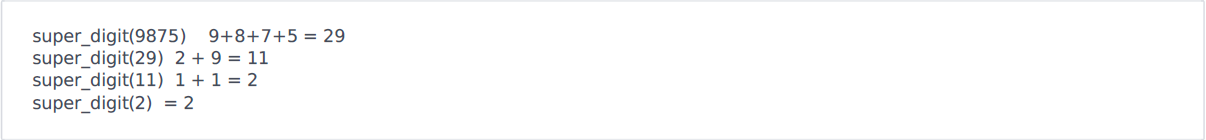
**3. RECURSIVE DIGIT SUM**

Definimos el super digito de un entero **X** usando las siguientes reglas:

Dado un entero necesitamos encontrar el super digito de este.

* Si **X** tiene solo 1 digito, entonces el super digito es **X**.
* En otro caso, el super digito de **X** es igual al super digito de la suma de los dígitos de **X**.

Por ejemplo, el super digito de 9875 se calculará como:



Se darán dos números **n** y **k**. El número p está creado concatenando los string **n**, **k** veces.

Continuando sobre el ejemplo, donde **n** = 9875, asuma el valor **k** = 4. Tú **p** inicial es: 9875 9875 9875 9875 (espacios agregados para claridad)



Todos los dígitos de **p** suman 116. Los dígitos de 116 suman 8. 8 es solo un digito. Entonces este es el super digito.

**Descripción de la función**

Complete la función **superDigit**. Esta debe de retornar el super digito calculado como un entero.

**superDigit** tiene los siguientes parámetros

* **n**: un string representando un entero
* **k**: un entero, las veces de la concatenación de n para construir p

**Restricciones**

* 1 <= n <= 10100000
* 1 <= k <= 105

**Entrada**

la primera línea contiene 2 enteros separados por un espacio, n y k.

**Salida**

Retorna el super digito de p, donde p es creado como se describió.

**Ejemplo de entrada 0**

148 3

**Ejemplo de salida 0**

3

**Ejemplo de entrada 1**

9875 4

**Ejemplo de salida 1**

8

Para más información consultar: <https://www.hackerrank.com/challenges/recursive-digit-sum/problem?h_r=internal-search> también puedes participar en el contest del laboratorio: <https://www.hackerrank.com/st0245-laboratorio1-ed1>