

Rekurzív számjegyösszeadás - A megoldás magyarázata

A megoldást a `recursive_digit_sum.py` fájlban található `superDigit` függvény szolgáltatja, amelyet az alábbiakban részletesen elmagyarázunk. Mivel rekurzív függvényről van szó, ezért a megállási feltétellel kezdjük: ha az `n` szám egyetlen számjegyből áll, amelyet nem kell többször egymás után írunk, azaz `k=1`, akkor visszaadjuk `n`-et. A vonatkozó kódrészletet láthatjuk az alábbiakban.

```
# ha n már eleve egy számjegyből áll, amelyet pontosan  
# egyszer kell egymás után írni, akkor visszaadjuk n-et  
if len(n) < 2 and k == 1:  
    return n
```

Abban az esetben ha `n` több számjegyből áll vagy `k>1`, azaz `n`-et többször kell egymás után írunk, akkor első lépésben kiszámoljuk `n` számjegyeinek összegét az alábbiak szerint:

```
# különben kiszámoljuk n számjegyeinek összeget  
sum = 0  
for digit in n:  
    sum += int(digit)
```

Ezután pedig meghívjuk a függvényből önmagát (vö. rekurzió), azonban az új hívásban `n` értéke már a korábbi `n` számjegyeinek összege `k`-szor egymás után írva, míg `k` új értéke pedig `1` lesz már. Vegyük észre, hogy így kevesebb rekurzív függvényhívásra van szükség, mintha először egymás mellé írnánk `n`-t `k`-szor, hogy megkapjuk `p`-t, majd `p` számjegyeinek összegét számolnánk ki.

```
# majd vesszük azt a számot, amelyet úgy kapunk, hogy  
# n számjegyeinek összeget k-szor egymás után írjuk;  
# az egymás után írás, azaz k értéket 1-re állítjuk  
return superDigit(str(sum) * k, 1)
```