Recursive Digit Sum

by wanbo

Problem Submissions Leaderboard Discussions Editorial

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

- 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 digit-sum of x. Here, digit-sum of a number is defined as the sum of its digits.

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

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

You are given two numbers n and k. You have to calculate the super digit of P.

P is created when number n is concatenated k times. That is, if $n = 123$ and $k = 3$, then $P = 123123123$.

Input Format

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

Constraints

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

Output Format

Output the super digit of P, where P is created as described above.

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.
```

Submissions: 1228
Max Score: 30
Difficulty: Medium

Rate This Challenge:

More

Current Buffer (saved locally, editable) C++

```cpp
#include <cmath>
#include <cstdio>
#include <vector>
#include <iostream>
#include <algorithm>
using namespace std;

unsigned int super_digit(unsigned int x) {
    if (x < 10) {
        return x;
    }

    int sum = 0;

    while (x) {
        sum += x % 10;
        x /= 10;
    }

    return super_digit(sum);
}

unsigned int super_digit(std::string x) {
    if (x.length() == 1) {
    return x[0] - '0';
    }

    unsigned int sum = 0;
    unsigned int i;

    for (i = 0; i < x.length(); i++) {
        sum += x[i] - '0';
        if (sum > 1000000000) sum = super_digit(sum);
    }

    sum = super_digit(sum);

    return super_digit(std::to_string(sum) + x.substr(i));
}

int main() {
    string n;
    int k;
    cin >> n;
    cin >> k;

    cout << super_digit(k * super_digit(n)) << endl;

    return 0;
}
```

Line: 1 Col: 1

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