Recursive Digit Sum

by wanbo

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:3396

Max Score:30
Difficulty: Medium

Rate This Challenge:

More

Current Buffer (saved locally, editable) C++14

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

void digit_sum(string initial_sum, int &ans)
{

    string ch;
    unsigned long long _sum=0;
    string ssum;
    if(initial_sum.length()==1)
    {
        ans =stoi(initial_sum)%10;
        return;
    }
    else
    {
        for(int i=0;i<initial_sum.length();i++)
        {
            ch=initial_sum[i];
            //cout <<stoi(ch) <<endl;
            _sum=stoi(ch)+_sum;

        }
        ssum = to_string(_sum);
        digit_sum(ssum, ans);
    }



}

int main() {
    /* Enter your code here. Read input from STDIN. Print output to STDOUT */
    string n;
    int k;

    cin >> n >> k;

    int input_len= n.length();
    unsigned long long initial_sum=0;
    string ch,ssum;
    int ans=0;
    for(int i=0;i<input_len;i++)
    {
        ch=n[i];
        //cout <<stoi(ch) <<endl;
        initial_sum=stoi(ch)+initial_sum;
```

```
53
54      }
55      initial_sum=initial_sum*k;
56      ssum=to_string(initial_sum);
57
58      digit_sum(ssum, ans);
59
60
61      //cout << n << " "<<k<< " "<< input_len << " "<<initial_sum<<endl;
62      cout<< ans;
63
64
65      return 0;
66  }
```

Line: 66 Col: 2

Test against custom input

Run Code | Submit Code

Upload Code as File

```
449207134995567705170 0
618860129870409856113 1
283923735846220610758 8
649940788514195131880 7
519645386474 100000
```

Congrats, you solved this challenge!

- ✔ Test Case #0
- ✔ Test Case #1
- ✔ Test Case #2
- ✔ Test Case #3
- ✔ Test Case #4
- ✔ Test Case #5
- ✔ Test Case #6
- ✔ Test Case #7
- ✔ Test Case #8
- ✔ Test Case #9

You've earned 30.00 points! Next Challenge