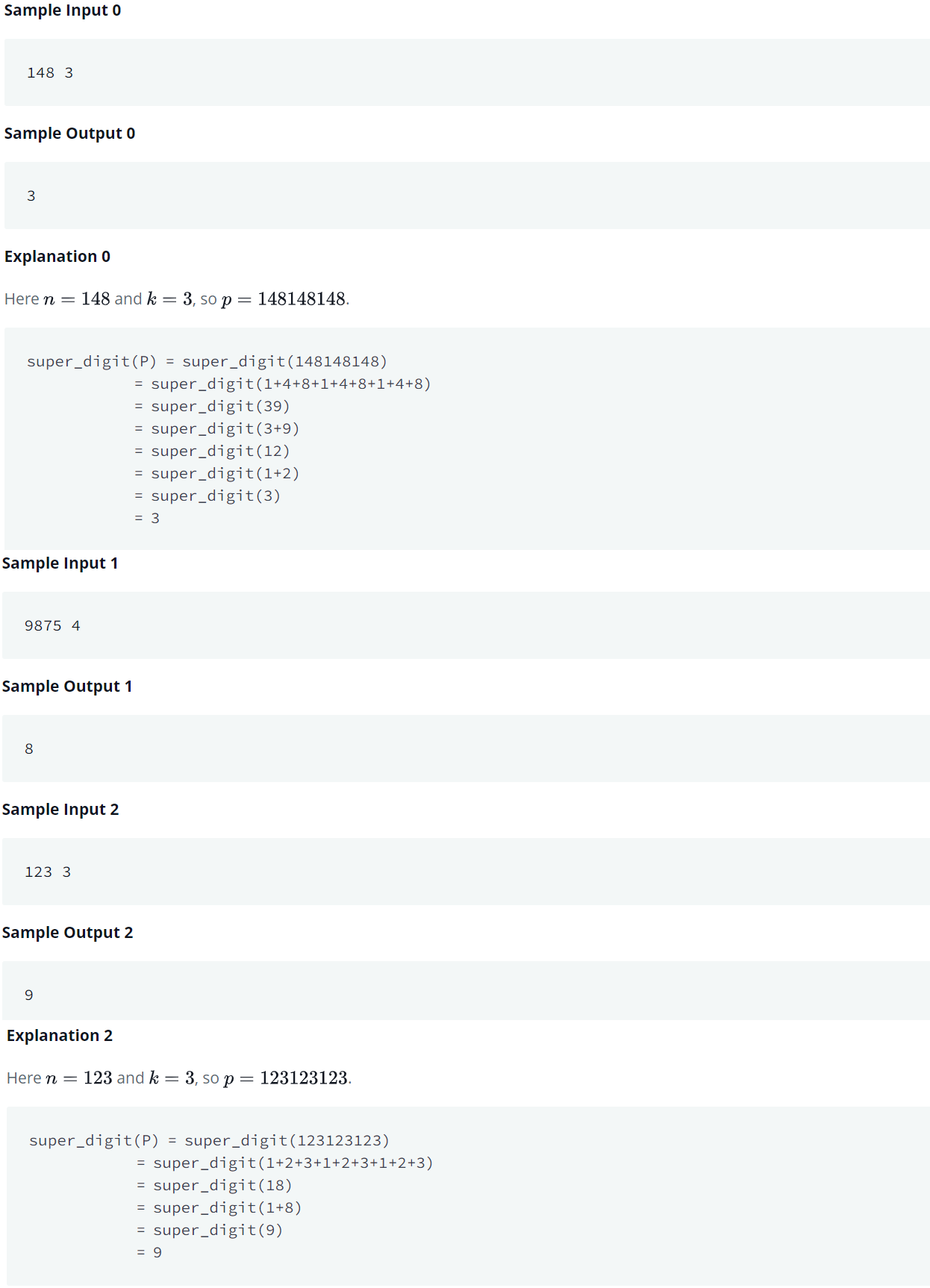
[A - Recursive Digit Sum](https://vjudge.net/problem/HackerRank-recursive-digit-sum" \t "_blank)





Recursive Digit Sum

Since number is very big as per constraints, it will be a good idea to read the number as string and first get sum of digits of number represented as string.

Original String S, Max Length can be = 100000  
Concatenating it as per MAX value of K 10^5, times will give a string of length = 10000000000 = 10^10  
Each char is 1 byte so 10^10 chars will be 10^10 bytes = 10^7 KB = 10^4 MB = 10 GB  
That will cause Memory Limit exceeded. Right?

Calculate sum of digits of String and Multiply that with K, it will give us sum of digits of concatenated S.  
  
e.g. S = "123" K = 5.  
  
**Approach 1**: Concatenate and then find sum.  
S = "123123123123123"  
sum = 30  
  
**Approach 2**: Find sum of digits of S and then multiply with K  
sum = 6 \* k = 30

Since it is also mentioned that given string is concatenated k times to form the number, so of course sum of digits of concatenated number will be equal to k \* sum of digits of original string.

Let's do the pre-work

string snum;  
int k;  
Read snum, k;  
int sumWithoutConcatenation = 0;  
for i = [0, len(snum)-1]  
    sumWithoutConcatenation += (snum[i] - '0'); // Convert char digit say '9' to int digit 9, by '9' - '0'  
  
int sumWithConcatenation = k \* sumWithoutConcatenation;  
  
Write **recursiveDigitSum**(sumWithConcatenation), "\n";

Now we have a sum that we need to reduce to single digit, so we can write a separate function.

int recursiveDigitSum(int num){  
    if(num < 10  
        return num;  
  
    int s = 0;  
    while(num != 0){  
        s += num % 10;  
        num /= 10;  
    }  
  
    if( s < 10)  
        return s; // if sum is already single digit, return that  
  
    return sumOfDigits(s); // Otherwise, call recursively to reduce s to single digit  
}

#include <iostream>

#include <algorithm>

#include <string>

using namespace *std*;

int getSum(long long int sumOfDigits)

{

int sum = 0;

while (sumOfDigits != 0) {

sum = sum + sumOfDigits % 10;

sumOfDigits = sumOfDigits / 10;

}

return sum;

}

int main(void) {

*ios*::*sync\_with\_stdio*(0);

*cin*.*tie*(0);

*cout*.*tie*(0);

*string* n;

int k;

*cin* >> n >> k;

int sum = 0;

for (auto i = 0; i < n.*size*(); i++)

{

int ia = n[i] - '0';

sum += ia;

}

long long int sumOfDigits = 0;

for (auto i = 0; i < k; i++)

{

sumOfDigits += sum;

}

auto num = getSum(sumOfDigits);

while (true)

{

num = getSum(num);

if (num < 10)

break;

}

*cout* << num << "\n";

return 0;

}