```
#include <stdio.h>
#include <math.h>

int main(){
    long N, newN = 0, temp, blocks, bPow;
    int k, len, i;
    scanf("%ld%d", &N, &k);
    len = log10(N) + 1; // Number of digits in the number
    blocks = len / k;
    if(len % k) blocks++; // One more (incomplete) block needed
    bPow = (long)pow(10,k); // How long is a block
    for(i = 0; i < blocks; i++){
        temp = N % bPow; // Find out the next block of numbers
        newN = newN + ((temp + temp%10) % bPow)*((long)pow(bPow,i)); // add the last digit,
discard any extra leading digits, and add to the front
                // The last multiplication with pow(bPow,i) is needed otherwise the leading block
may look like 00008 or so.
        N = N / bPow; // Discard this block
    }
    printf("%ld\n%ld", blocks, newN);
    return 0;
}
```