



## Problem C. Careful Multiplication

Time Limit 1 second

### Problem

We learned about modular operations. In the modular world, these three formulas are always true:

- $(A + B) \bmod M = (A \bmod M + B \bmod M) \bmod M$
- $(A - B) \bmod M = (A \bmod M - B \bmod M) \bmod M$
- $(A \times B) \bmod M = (A \bmod M \times B \bmod M) \bmod M$

And by using 64-bit integers, we could easily calculate some numbers modulo  $10^9$ . In this way, we could solve problems that required us to print the last 9 digits or the last 9 digits of the number (by always storing the last 9 digits during the calculation, and carefully using 64-bit integers during multiplication). So, what is this problem about? This problem is so simple that given  $A$ ,  $B$  and  $M$ , you just need to calculate  $(A \times B) \bmod M$ . So simple, right? However, in this problem we decided to set the constraints as  $1 \leq A, B, M \leq 10^{18}$ . With this large constraint, IT IS IMPOSSIBLE TO MULTIPLY TWO NUMBERS MODULO  $M$  USING A 64-BIT INTEGER, because  $(10^{18} - 1) \times (10^{18} - 1) \approx 10^{36} \gg 2^{63} - 1$ .

So, what method should you use? Take a look at the hints section, if you want.

### Input

Our input consists of an arbitrary number of lines, but no more than 10,000.

Each line contains three integers  $A$ ,  $B$  and  $M$  ( $1 \leq A, B, M \leq 10^{18}$ ), each separated by a space. Note that you need to use 64-bit integers to store these integers.

The end of input is indicated by a line containing only the value  $-1$ .

### Output

For each input line, print the value of  $(A \times B) \bmod M$ .

Sample Input 1	Sample Output 1
5 6 7	2
2017 7 19	2
7 19 2017	133
943492 189348291412 32418318	25220980
999999999999999999 999999999999999999 999999999999999999	999999998200000001
-1	

### Hints

Do you remember how we calculated  $r = A \bmod m$  using the recursive definition of  $r$  in time complexity  $O(\log n)$ ? In that lecture, we also stated that we can calculate the value of  $f(n)$  in  $O((T(n) \times \log(n))$  time if:

- We can calculate  $f(n)$  from  $f(n - 1)$  in  $O(T(n))$  time.
- We can calculate  $f(n)$  from  $f(n/2)$  in  $O(T(n))$  time.



**acm** International Collegiate  
Programming Contest

**ASTU**  
Competitive Programming Contest 2011 E.C.

Try to use this fact.

Just adding  $A \cdot B$  times will NOT WORK, as its time complexity is  $O(B) \cdot 10^{18}$  operations are too much..