

# Matrix Modulo

Time limit: 1 sec

Given a 2x2 non negative integer matrix **A**, we would like to calculate of **A<sup>n</sup> mod k**, where n and k are given positive integers.

In a modular arithmetic, a mod k is the remainder of a / k. For example 14 mod 5 equals to 4. In c++, we can calculate a mod k by this expression **a % k**. A matrix M mod k where  $M = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$  is defined as  $M = \begin{bmatrix} a \bmod k & b \bmod k \\ c \bmod k & d \bmod k \end{bmatrix}$  and we also know that, for a positive integers **p** and **q**

- **(p + q) mod k = ((p mod k) + (q mod k)) mod k**
- **(p \* q) mod k = ((p mod k) \* (q mod k)) mod k**
- This property is also valid for matrices **M** and **N**
- **(MN) mod k = ((M mod k)(N mod k)) mod k**

## Input

- The first line of input contains two integer **n** and **k** where  $1 \leq N \leq 2^{30}$  and  $1 \leq K \leq 10,000$ .
- The second line contains four integers a, b, c, d which describe the matrix **A** =  $\begin{bmatrix} a & b \\ c & d \end{bmatrix}$  where  $1 \leq a, b, c, d \leq 9,999$

## Output

There must be exactly **1** line that contains four integers that describe the matrix **A<sup>n</sup> mod k**

## Suggestion

- For 50% of the testcases, the value of n and k is very small such that a direct approach in O(N) would pass
- **It will help to write a function that calculate (MN) mod k.** Be sure that that function must return a matrix. For example, the function could be written as `vector<int> matrix_multiply(const vector<int> &M,const vector<int> &N,int k);`

## Example

Input	Output
2 1000 1 2 3 4	7 10 15 22
2 10 1 2 3 4	7 0 5 2

5 10000 1 2 3 4	1069 1558 2337 3406
5 10 1 2 3 4	9 8 7 6
999888777 4726 3 8 7 2	337 2916 3733 1154