Problem H. Big Mod 3

Time Limit	2000 ms
Mem Limit	1572864 kB
Code Length Limit	50000 B
OS	Linux

Exponentiation is a mathematical operation, written as b^n, involving two numbers, the base b and the exponent n. When n is a positive integer, exponentiation corresponds to repeated multiplication of the base: that is, b^n is the product of multiplying n bases:

$b^n = $ b × b × b × ... × b

In computing, the modulo operation finds the remainder after division of one number by another (sometimes called *modulus*). Given two positive numbers, a (the dividend) and n (the divisor), a modulo n (abbreviated as a mod n) is the remainder of the Euclidean division of a by n. For instance, the expression "5 mod 2" would evaluate to 1 because 5 divided by 2 leaves a quotient of 2 and a remainder of 1, while "9 mod 3" would evaluate to 0 because the division of 9 by 3 has a quotient of 3 and leaves a remainder of 0; there is nothing to subtract from 9 after multiplying 3 times 3.

Now, you are given the value of a, b and m. print the value of $\mathbf{a^b \bmod m}$.

Input

First line contains the number of test cases t ($1 \le t \le 10^4$). Next t line contains three integers a, b and m. where $1 \le a, b \le 10^9$ and $1 \le m \le 2^{64}$.

Output

For each test case print the answer of the problem.

Sample input

```
2
2 3 4
3 4 5
```

Sample output

```
0
1
```