

Project Euler #188: The hyperexponentiation of a number

This problem is a programming version of [Problem 188](#) from [projecteuler.net](#)

The *hyperexponentiation* or *tetration* of a number a by a positive integer b , denoted by $a \uparrow\uparrow b$ or ${}^b a$, is recursively defined by:

$$a \uparrow\uparrow 1 = a,$$
$$a \uparrow\uparrow (k + 1) = a^{a \uparrow\uparrow k}.$$

Thus we have e.g. $3 \uparrow\uparrow 2 = 3^3 = 27$, hence $3 \uparrow\uparrow 3 = 3^{27} = 7625597484987$ and $3 \uparrow\uparrow 4$ is roughly $10^{3.6383346400240996 \cdot 10^{12}}$.

Find $a \uparrow\uparrow b \pmod m$.

Input Format

The first line of each test file contains a single integer T — the number of test cases. T lines follow, each containing three integers separated by single spaces: a , b and m .

Constraints

$$1 \leq T \leq 10$$
$$1 \leq a, b, m \leq 10^{18}$$

Output Format

Output exactly T lines with the corresponding $a \uparrow\uparrow b \pmod m$ on each line.

Sample Input 0

```
1
3 3 1000000000000000000
```

Sample Output 0

```
7625597484987
```