

4145 Hyper-Mod

Hyper operator $a^{(n)}b$ is a family of operators, where each level n operator is defined as iteration of the previous level $(n - 1)$ operator. It is defined recursively as follows:

$$a^{(n)}b = \begin{cases} b + 1, & \text{if } n = 0 \\ a, & \text{if } n = 1, b = 0 \\ 0, & \text{if } n = 2, b = 0 \\ 1, & \text{if } n \geq 3, b = 0 \\ a^{(n-1)}(a^{(n)}(b - 1)) & \text{if } n \geq 1, b \geq 1, a \geq 0 \end{cases}$$

With some algebraic manipulations, it can be shown that the operators for $n = 1, 2$ and 3 correspond respectively to addition, multiplication and exponentiation; as given below:

$$a^{(1)}b = a + 1 + 1 + 1 + \cdots + 1 = a + b$$

$$a^{(2)}b = a + a + a + \cdots + a = a \times b$$

$$a^{(3)}b = a \times a \times a \times \cdots \times a = a^b$$

For values of $n > 3$, $a^{(n)}b$ typically corresponds to a (very) tall exponentiation tower even for small values of a and b , which makes it difficult or even impossible to calculate exactly. Nevertheless, as they are being iterations (or iterations of iterations of ...) of exponents, their modulus can still be calculated by means of modular hyper-exponentiation (a form of iterated modular exponentiation). Your task is to compute $a^{(n)}b \bmod m$, given the values of a, b, m and n .

Input

The first line of input contains an integer T (not more than 100), the number of test cases follow. The next T lines of input describe the input for each test case on one line, in the form of ' $m n a b$ '. You can safely assume that $0 \leq n \leq 10$; $0 \leq a, b \leq 1000$; and $1 \leq m \leq 10000$.

Output

For each of the test cases, print the test case number followed by the value of $a^{(n)}b \bmod m$ on one line. The sample output shows the exact format for printing the test case number.

Sample Input

```
4
78 1 123 456
257 3 16 16
2008 4 8 3
1000 4 13 27
```

Sample Output

```
Case #1: 33
Case #2: 1
Case #3: 816
Case #4: 53
```