Boring Factorials (Reloaded)

Factorial is one of the most attractive word this week, it is proposed to reload a famous [problem](#).
Is it so boring?

Sameer and Arpit want to overcome their fear of Maths and so they have been recently practicing
Maths problems a lot. Aman, their friend has been helping them out. But as it goes, Sameer and
Arpit have got bored of problems involving factorials. Reason being, the factorials are too easy to
calculate in problems as they only require the residue modulo some prime and that is easy to
calculate in linear time. So to make things interesting for them, Aman - The Mathemagician, gives
them an interesting task. He gives them a prime number **P** and an integer **N** (not so) close to **P**,
and asks them to find **N! modulo P**. He asks **T** such queries.

Input

The first line of input contains an integer **T**, the number of test cases.

On each of the next **T** lines, your are given two integers **N**, and **P** a prime number.

Output

For each test case, you have to print **N! modulo P**.

Example

Input:
3
2 5
5 11
21 71

Output:
2
10
6

Constraints

$0 < T < 10^4$
$0 < N < 4 \times 10^{18}$
$1 < P < 4 \times 10^{18}$, a prime number
$Abs(N-P) < 10^4$

Problem designed to be solvable using some 'slow' languages like Python (333B of code). It is
highly recommended to solve in a very fast way the original problem! Basic solution, even in fast
language should give TLE.
Time limit is set as sqrt(correct × basic). The "basic" code gave me 0.00s to the original problem
with slow IO.
If you don't find this reload edition hard enough, please consider the [challenge edition](#) with more
serious constraints.

;-) Have fun.