# WC '97 P2 - Power of Cryptography

**Time limit:** 1.0s **Memory limit:** 16M

### 1997 Woburn Computer Programming Challenge

Current work in cryptography involves (among other things) computing large prime numbers and computing powers of numbers modulo these large primes. Work in this area has resulted in the practical use of results from number theory and other branches of mathematics once considered to be only of theoretical interest. This problem involves the efficient calculation of integer roots of numbers.

Given an integer $n \geq 1$ and an integer $p \geq 1$ you are to write a program that determines the $n$-th root of $p$ — it is guaranteed that $p$ is the $n$-th power of some **integer** $k$, i.e. $p = k^n$ for some integer $k$; this is the integer you are to find.

## Input Specification

The first line of the input is $M$, the number of test cases to consider.
The input consists of $M$ pairs of numbers $n$ and $p$ with each number on a line by itself. For all of these pairs, $1 \leq n \leq 200$, $1 \leq p \leq 10^{101}$ and there exists an integer $k$, $1 \leq k \leq 10^{101}$ such that $k^n = p$.

## Output Specification

For each set of values for $n$ and $p$ output the value of $k$.

## Sample Input

```
3
2
16
3
27
7
4357186184021382204544
```

## Sample Output

```
4
3
1234
```