Cracking RSA

Problem ID: crackingrsa
CPU Time limit: 3 seconds
Memory limit: 1024 MB
Difficulty: 2.4

Author: Daniel Moyer
Source: 2016 Virginia Tech Hi School Programming Contest
License: (cc) BY-SA

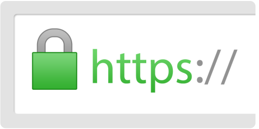

Image by Sean MacEntee (Flickr) [CC BY 2.0]

RSA is a widely used public-key cryptosystem that allows two parties (such as people or computers) to exchange secret messages without revealing information to anyone else listening on the conversation. Many websites use RSA when visited over a secure `https` connection. In RSA, each party has two different keys: a *public* key that is published and a *private* key that is kept secret. To encrypt a message intended for a specific recipient, the sender will use the recipient's public key to encrypt the message. The recipient will use their private key to decrypt the message.

An RSA public key (n, e) consists of two numbers n and e. The number n is a product of two distinct prime numbers, p and q. In real applications, n would be hundreds of decimal digits long for security.

Let $\varphi(n)$ be Euler's *totient* function, which in this case is equal to $(p-1)(q-1)$. The private key consists of (n, d), where n is the same as in the public key and d is the solution to the congruence

$$de \equiv 1 \bmod \varphi(n)$$

Formally, a congruence

$$a \equiv b \bmod c$$

holds for three integers a, b, and c, if there exists an integer k such that $a - b = kc$.

The sender will encrypt a message M (which, for simplicity, is assumed to be an integer smaller than both p and q) by computing $M^e \bmod n$ and sending it to the receiver. The recipient will calculate $(M^e)^d \equiv M^{ed} \equiv M^{k\varphi(n)+1} \equiv M^{\varphi(n)k} M \equiv M \bmod n$ since by Euler's theorem $M^{\varphi(n)} \equiv 1 \bmod n$. This will reconstruct the original message. Without the private key, no practical way has been found for a potential attacker to recover M from the knowledge of $M^e \bmod n$ and (n, e).

Your task is to crack RSA by finding the private key related to a specific public key.

Input

The first line of input has the number of test cases T, $(1 \leq T \leq 50)$. Each test case has one line that contains the two numbers n and e. You may assume that n is the product of two primes p, q such that $2 \leq p, q < 1000$. Also, e will be chosen so that d exists and is unique, and $1 < d, e < \varphi(n)$. Note that the product de may not fit into a 32-bit integer (e.g. Java's `int` type).

Output

For each test case, output the single number d.

Sample Input 1

```
2
33 3
65 11
```

Sample Output 1

```
7
35
```