1054 – Efficient Pseudo Code

Sometimes it's quite useful to write pseudo codes for problems. Actually you can write the necessary steps to solve a particular problem. In this problem you are given a pseudo code to solve a problem and you have to implement the pseudo code efficiently. Simple! Isn't it? :)

```
pseudo code
{
   take two integers n and m
   let p = n ^ m (n to the power m)
   let sum = summation of all the divisors of p
   let result = sum MODULO 1000,000,007
}
```

Now given **n** and **m** you have to find the desired result from the pseudo code. For example if **n = 12** and **m = 2**. Then if we follow the pseudo code, we get

```
pseudo code
{
   take two integers n and m
   so, n = 12 and m = 2
   let p = n ^ m (n to the power m)
   so, p = 144
   let sum = summation of all the divisors of p
   so, sum = 403, since the divisors of p are 1, 2, 3, 4, 6, 8, 9, 12, 16, 18, 24, 36, 48, 72, 144
   let result = sum MODULO 1000,000,007
   so, result = 403
}
```

Input

Input starts with an integer **T (≤ 5000)**, denoting the number of test cases.

Each test case will contain two integers, **n (1 ≤ n)** and **m (0 ≤ m)**. Each of **n** and **m** will be fit into a **32** bit signed integer.

Output

For each case of input you have to print the case number and the result according to the pseudo code.

Sample Input	Output for Sample Input
3 12 2 12 1 36 2	Case 1: 403 Case 2: 28 Case 3: 3751