Generators

Many cryptographic keys make use of so-called generators. This is for example the case for the Diffie-Hellman key exchange protocol.

We say that the number $g \in \mathbb{N}_0$ is a **generator** of the prime number $p \in \mathbb{N}$ (with $g < p$) if the smallest number $n \in \mathbb{N}$ (with $n > 1$) so that $g^n\ \text{mod}\ p \equiv g$ is equal to p. The modulo operator mod finds the remainder after (integer) division.

Suppose that $g = 5$ and $p = 7$. We then compute the successive powers of $g = 5$, starting with the second power, each time computing the remainder after division by the prime number $p = 7$. $$ \begin{array}{rcl} 5^2 \mod 7 &=& 4 \\ 5^3 \mod 7 &=& 6 \\ 5^4 \mod 7 &=& 2 \\ 5^5 \mod 7 &=& 3 \\ 5^6 \mod 7 &=& 1 \\ 5^7 \mod 7 &=& 5 \end{array} $$ This process will always produce a result that equals $g = 5$ for a certain power (a property of prime numbers), which in this case happens for the first time with the seventh power. Because 7 is also the value of the prime number p, we say that $g = 5$ is a generator of $p = 7$.

Suppose that $g = 4$ and $p = 7$, and we compute again the successive powers of $g = 4$, starting with the second power, until the process results in the number $g = 4$. $$ \begin{array} {rcl} 4^2 \mod 7 &=& 2 \\ 4^3 \mod 7 &=& 1 \\ 4^4 \mod 7 &=& 4 \end{array} $$ Because this happens for the first time with the fourth power, and 4 is not equal to the prime number p, we say that $g = 4$ is not a generator of $p = 7$.

Input

The input contains two numbers $g, p \in \mathbb{N}_0$, each on a separate line. The value p is a prime number, with $g < p$.

Output

Determine whether of not the number g is a generator of the number p. You do this by computing the successive powers of g, starting with the second power, and finding the remainder after division by the number p. Do this until you obtain the number g for the first time (this will always happen because p is a prime number). For each power evaluated, a single line of output must be generated whose format and content can be derived from the examples given below (we represent m^n as m^n). After all powers have been evaluated, and the corresponding lines of output have been generated, a last line of output must be generated that indicates whether or not g is a generator of p. Again, you can derive the format of this line of output from the examples given below.

Example

Input:

```
5
7
```

Output:

(5^2) mod 7 = 4
(5^3) mod 7 = 6
(5^4) mod 7 = 2
(5^5) mod 7 = 3
(5^6) mod 7 = 1
(5^7) mod 7 = 5
5 is a generator of 7

Example

Input:

4
7

Output:

(4^2) mod 7 = 2
(4^3) mod 7 = 1
(4^4) mod 7 = 4
4 is not a generator of 7

In de cryptografie maakt men vaak gebruik van zogenaamde generatoren. Dat is bijvoorbeeld het geval voor het Diffie-Hellman-sleuteluitwisselingsprotocol.

We zeggen dat een getal $g \in \mathbb{N}_0$ een **generator** is van het priemgetal $p \in \mathbb{N}$, waarbij geldt dat $g < p$, als het kleinste getal $n \in \mathbb{N}$ (met $n > 1$) waarvoor $g^n\ \text{mod}\ p \equiv g$ gelijk is aan p. Hierbij staat de operator mod voor de rest na (gehele) deling.

Stel dat $g = 5$ en $p = 7$. Dan berekenen we de opeenvolgende machten van $g = 5$, te beginnen bij de tweede macht, en bepalen telkens de rest na deling door het priemgetal $p = 7$. $$ \begin{array}{rcl} 5^2 \mod 7 &=& 4 \\ 5^3 \mod 7 &=& 6 \\ 5^4 \mod 7 &=& 2 \\ 5^5 \mod 7 &=& 3 \\ 5^6 \mod 7 &=& 1 \\ 5^7 \mod 7 &=& 5 \end{array} $$ Uiteindelijk zal dit proces altijd terug het getal $g = 5$ opleveren, en dat gebeurt hier voor het eerst bij de zevende macht. Omdat 7 ook de waarde is van het priemgetal p, zeggen we in dit geval dat $g = 5$ een generator is van $p = 7$.

Stel dat $g = 4$ en $p = 7$, en dat we opnieuw de opeenvolgende machten van $g = 4$ bepalen, te beginnen bij de tweede macht, totdat dit het getal $g = 4$ oplevert. $$ \begin{array} {rcl} 4^2 \mod 7 &=& 2 \\ 4^3 \mod 7 &=& 1 \\ 4^4 \mod 7 &=& 4 \end{array} $$ Omdat dit voor het eerst gebeurt bij de vierde macht, en 4 niet gelijk is aan het priemgetal p, zeggen we in dit geval dat $g = 4$ geen generator is van $p = 7$.

Invoer

De invoer bestaat uit twee getallen $g, p \in \mathbb{N}_0$, die elk op een afzonderlijke regel staan. Hierbij geldt dat p een priemgetal is waarvoor geldt dat $g < p$.

Uitvoer

Bepaal of het getal g al dan niet een generator is van p. Hiervoor bepaal je de opeenvolgende machten g, te beginnen vanaf de tweede macht, waarvan je telkens de rest na

deling door p bepaalt. Dit doe je totdat je voor het eerst terug het getal g bekomt (omdat p een priemgetal is, is dat altijd het geval). Voor elke macht die hierbij berekend wordt, moet een regel naar de uitvoer uitgeschreven worden waarvan het formaat en de inhoud kan afgeleid worden uit onderstaande voorbeelden (we stellen hierbij m^n voor als m^n). Nadat je alle machten hebt berekend, en daarbij telkens een bijhorende regel hebt uitgeschreven, moet er nog een laatste regel uitgeschreven worden die aangeeft of g al dan niet een generator is van p. Het formaat van deze regel kan je opnieuw afleiden uit onderstaande voorbeelden.

Voorbeeld

Invoer:

5
7

Uitvoer:

(5^2) mod 7 = 4
(5^3) mod 7 = 6
(5^4) mod 7 = 2
(5^5) mod 7 = 3
(5^6) mod 7 = 1
(5^7) mod 7 = 5
5 is een generator van 7

Voorbeeld

Invoer:

4
7

Uitvoer:

(4^2) mod 7 = 2
(4^3) mod 7 = 1
(4^4) mod 7 = 4
4 is geen generator van 7