A. Number of Divisors

1 second, 256 megabytes

For a positive integer \( n \), we use \( f(n) \) to denote the number of divisors of \( n \).

For example, \( f(1) = 1 \), \( f(2) = 2 \), \( f(3) = 2 \), \( f(4) = 3 \).

Given \( n \), output \( \sum_{i=1}^{n} f(n) \).

**Input**

A single integer \( n \) \((1 \leq n \leq 10^6)\).

**Output**

Display \( \sum_{i=1}^{n} f(n) \).

```
input
4
output
8
```

B. Mertens Function

1 second, 256 megabytes

For a given positive integer \( n \), let define \( \mu(n) \) to be

\[
\mu(n) = \begin{cases} 
0 & \text{if } n \text{ has one or more repeated prime factors} \\
1 & \text{if } n = 1 \\
(-1)^k & \text{if } n \text{ has } k \text{ distinct prime factors}
\end{cases}
\]

For a given positive integer \( n \), the Mertens function is defined to be \( M(n) = \sum_{i=1}^{n} \mu(n) \).

Given a given positive integer \( n \), output the Mertens function \( M(n) \).

**Input**

A single integer \( n \) \((1 \leq n \leq 10^7)\).

**Output**

Display the Mertens function \( M(n) \).

```
input
5
output
-2
```
C. GCD Queries
1 second, 256 megabytes

There are \( n \) integers. Initially they are \( a_1, a_2, \ldots, a_n \). There are \( Q \) modifications. The \( i \)-th modification will increase all the \( n \) integers by \( \Delta_i \). Note that \( \Delta_i \) could be negative. For each modification, output the greatest common divisor of the \( n \) integers after that modification.

**Input**
The first line contains two integers \( n \) and \( Q \) (\( 1 \leq n, Q \leq 10^5 \)).
Each of the following \( n \) lines contains a single integer. The \( i \)-th line gives \( a_i \).
Each of the following \( Q \) lines contains a single integer. The \( i \)-th line gives \( \Delta_i \).
It is guaranteed that all the \( n \) integers are in \([-10^{16}, 10^{16}]\) at any time.

**Output**
Display \( Q \) lines, where each line contains a single integer. The \( i \)-th integer is the greatest common divisor of the \( n \) integers after the \( i \)-th modification.

```
input
3 2
 1
-5
 7
-1
 1
output
6
1
```

D. Divisible by Seven
1 second, 256 megabytes

You have number \( a \), whose decimal representation quite luckily contains digits 1, 6, 8, 9. Rearrange the digits in its decimal representation so that the resulting number will be divisible by 7.

Number \( a \) doesn’t contain any leading zeroes and contains digits 1, 6, 8, 9 (it also can contain another digits). The resulting number also mustn’t contain any leading zeroes.

**Input**
The first line contains positive integer \( a \) in the decimal record. It is guaranteed that the record of number \( a \) contains digits: 1, 6, 8, 9. Number \( a \) doesn’t contain any leading zeroes. The decimal representation of number \( a \) contains at least 4 and at most \( 10^6 \) characters.

**Output**
Print a number in the decimal notation without leading zeroes — the result of the permutation.

If it is impossible to rearrange the digits of the number \( a \) in the required manner, print 0.

```
input
1689
output
1869

input
18906
output
18690
```
E. Multipliers

2 seconds, 256 megabytes

Ayrat has number \( n \), represented as its prime factorization \( p_1 \) of size \( m \), i.e. \( n = p_1 \cdot p_2 \cdot \ldots \cdot p_m \). Ayrat got secret information that the product of all divisors of \( n \) taken modulo \( 10^9 + 7 \) is the password to the secret data base. Now he wants to calculate this value.

**Input**
The first line of the input contains a single integer \( m \) \((1 \leq m \leq 200\,000)\) — the number of primes in factorization of \( n \).

The second line contains \( m \) primes numbers \( p_i \) \((2 \leq p_i \leq 200\,000)\).

**Output**
Print one integer — the product of all divisors of \( n \) modulo \( 10^9 + 7 \).

```
input
2
2 3

output
36

```

```
input
3
2 3 2

output
1728
```

In the first sample \( n = 2 \cdot 3 = 6 \). The divisors of 6 are 1, 2, 3 and 6, their product is equal to \( 1 \cdot 2 \cdot 3 \cdot 6 = 36 \).

In the second sample \( 2 \cdot 3 \cdot 2 = 12 \). The divisors of 12 are 1, 2, 3, 4, 6 and 12. \( 1 \cdot 2 \cdot 3 \cdot 4 \cdot 6 \cdot 12 = 1728 \).
F. The Sum of the k-th Powers

2 seconds, 256 megabytes

There are well-known formulas: \( \sum_{i=1}^{n} i = 1 + 2 + \ldots + n = \frac{n(n+1)}{2}, \sum_{i=1}^{n} i^2 = 1^2 + 2^2 + \ldots + n^2 = \frac{n(n+1)(2n+1)}{6} \).

\( \sum_{i=1}^{n} i^3 = 1^3 + 2^3 + \ldots + n^3 = \left( \frac{n(n+1)}{2} \right)^2 \). Also mathematicians found similar formulas for higher degrees.

Find the value of the sum \( \sum_{i=1}^{n} i^k = 1^k + 2^k + \ldots + n^k \) modulo \( 10^9 + 7 \) (so you should find the remainder after dividing the answer by the value \( 10^9 + 7 \)).

**Input**
The only line contains two integers \( n, k \) (\( 1 \leq n \leq 10^9 \), \( 0 \leq k \leq 10^6 \)).

**Output**
Print the only integer \( a \) — the remainder after dividing the value of the sum by the value \( 10^9 + 7 \).

<table>
<thead>
<tr>
<th>input</th>
<th>output</th>
</tr>
</thead>
<tbody>
<tr>
<td>4 1</td>
<td>10</td>
</tr>
<tr>
<td>4 2</td>
<td>30</td>
</tr>
<tr>
<td>4 3</td>
<td>100</td>
</tr>
<tr>
<td>4 0</td>
<td>4</td>
</tr>
</tbody>
</table>
G. On Iteration of One Well-Known Function

1 second, 256 megabytes

Of course, many of you can calculate \( \varphi(n) \) — the number of positive integers that are less than or equal to \( n \), that are coprime with \( n \). But what if we need to calculate \( \varphi(\varphi(\ldots \varphi(n))) \), where function \( \varphi \) is taken \( k \) times and \( n \) is given in the canonical decomposition into prime factors?

You are given \( n \) and \( k \), calculate the value of \( \varphi(\varphi(\ldots \varphi(n))) \). Print the result in the canonical decomposition into prime factors.

**Input**
The first line contains integer \( m \) (1 ≤ \( m \) ≤ 10^5) — the number of distinct prime divisors in the canonical representation of \( n \).

Each of the next \( m \) lines contains a pair of space-separated integers \( p_i, a_i \) (2 ≤ \( p_i \) ≤ 10^6; 1 ≤ \( a_i \) ≤ 10^17) — another prime divisor of number \( n \) and its power in the canonical representation. The sum of all \( a_i \) doesn’t exceed 10^{17}. Prime divisors in the input follow in the strictly increasing order.

The last line contains integer \( k \) (1 ≤ \( k \) ≤ 10^{18}).

**Output**
In the first line, print integer \( \omega \) — the number of distinct prime divisors of number \( \varphi(\varphi(\ldots \varphi(n))) \), where function \( \varphi \) is taken \( k \) times.

Each of the next \( \omega \) lines must contain two space-separated integers \( q_i, b_i \) (\( b_i \) ≥ 1) — another prime divisor and its power in the canonical representation of the result. Numbers \( q_i \) must go in the strictly increasing order.


You can read about function \( \varphi(n) \) here: http://en.wikipedia.org/wiki/Euler%27s_totient_function.