G. Coprime Arrays

time limit per test: 3.5 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Let's call an array a of size n *coprime* iff $gcd(a_1, a_2, ..., a_n) = 1$, where gcd is the greatest common divisor of the arguments.

You are given two numbers n and k. For each i $(1 \le i \le k)$ you have to determine the number of *coprime* arrays a of size n such that for every j $(1 \le j \le n)$ $1 \le a_j \le i$. Since the answers can be very large, you have to calculate them modulo $10^9 + 7$.

Input

The first line contains two integers n and k $(1 \le n, k \le 2 \cdot 10^6)$ — the size of the desired arrays and the maximum upper bound on elements, respectively.

Output

Since printing $2 \cdot 10^6$ numbers may take a lot of time, you have to output the answer in such a way:

Let b_i be the number of *coprime* arrays with elements in range $[1, i]$, taken modulo $10^9 + 7$. You have to print , taken modulo $10^9 + 7$. Here denotes bitwise xor operation (^ in C++ or Java, `xor` in Pascal).

Examples

input
3 4
output
82

input
2000000 8
output
339310063

Note

Explanation of the example:

Since the number of *coprime* arrays is large, we will list the arrays that are non-coprime, but contain only elements in range $[1, i]$:

For $i = 1$, the only array is coprime. $b_1 = 1$.

For $i = 2$, array $[2, 2, 2]$ is not coprime. $b_2 = 7$.

For $i = 3$, arrays $[2, 2, 2]$ and $[3, 3, 3]$ are not coprime. $b_3 = 25$.

For $i = 4$, arrays $[2, 2, 2]$, $[3, 3, 3]$, $[2, 2, 4]$, $[2, 4, 2]$, $[2, 4, 4]$, $[4, 2, 2]$, $[4, 2, 4]$, $[4, 4, 2]$ and $[4, 4, 4]$ are not coprime. $b_4 = 55$.