Problem Set 7 — Number Theory 15-295 Spring 2019

A. Beautiful Numbers

2 seconds, 256 megabytes

Vitaly is a very weird man. He's got two favorite digits a and b. Vitaly calls a positive integer *good*, if the decimal representation of this integer only contains digits a and b. Vitaly calls a good number *excellent*, if the sum of its digits is a good number.

For example, let's say that Vitaly's favourite digits are 1 and 3, then number 12 isn't good and numbers 13 or 311 are. Also, number 111 is excellent and number 11 isn't.

Now Vitaly is wondering, how many excellent numbers of length exactly n are there. As this number can be rather large, he asks you to count the remainder after dividing it by 1000000007 $(10^9 + 7)$.

A number's length is the number of digits in its decimal representation without leading zeroes.

Input

The first line contains three integers: a, b, n $(1 \le a < b \le 9, 1 \le n \le 10^6)$.

Output

Print a single integer — the answer to the problem modulo 1000000007 $(10^9 + 7)$.

input
1 3 3
output
1

input
2 3 10
output
165

B. Remainders Game

1 second, 256 megabytes

Today Pari and Arya are playing a game called Remainders.

Pari chooses two positive integer x and k, and tells Arya k but not x. Arya have to find the value $x \mod k$. There are n ancient numbers $c_1, c_2, \ldots, c_n$ and Pari has to tell Arya $x \mod c_i$ if Arya wants. Given k and the ancient values, tell us if Arya has a winning strategy independent of value of x or not. Formally, is it true that Arya can understand the value $x \mod k$ for any positive integer x?

Note, that $x \mod y$ means the remainder of x after dividing it by y.

Input

The first line of the input contains two integers n and k ($1 \le n, \ k \le 1\,000\,000$) — the number of ancient integers and value k that is chosen by Pari.

The second line contains n integers $c_1, c_2, \ldots, c_n$ ($1 \le c_i \le 1\,000\,000$).

Output

Print "Yes" (without quotes) if Arya has a winning strategy independent of value of x, or "No" (without quotes) otherwise.

input
4 5 2 3 5 12
output
Yes

input
2 7 2 3
output
No

In the first sample, Arya can understand $x \mod 5$ because 5 is one of the ancient numbers.

In the second sample, Arya can't be sure what $x \mod 7$ is. For example 1 and 7 have the same remainders after dividing by 2 and 3, but they differ in remainders after dividing by 7.

C. Kolya and Tanya

1 second, 256 megabytes

Kolya loves putting gnomes at the circle table and giving them coins, and Tanya loves studying triplets of gnomes, sitting in the vertexes of an equilateral triangle.

More formally, there are $3n$ gnomes sitting in a circle. Each gnome can have from 1 to 3 coins. Let's number the places in the order they occur in the circle by numbers from 0 to $3n$ - 1, let the gnome sitting on the i-th place have a_i coins. If there is an integer i ($0 \le i < n$) such that $a_i + a_{i+n} + a_{i+2n} \neq 6$, then Tanya is satisfied.

Count the number of ways to choose a_i so that Tanya is satisfied. As there can be many ways of distributing coins, print the remainder of this number modulo $10^9 + 7$. Two ways, a and b, are considered distinct if there is index i ($0 \le i < 3n$), such that $a_i \neq b_i$ (that is, some gnome got different number of coins in these two ways).

Input

A single line contains number n ($1 \le n \le 10^5$) — the number of the gnomes divided by three.

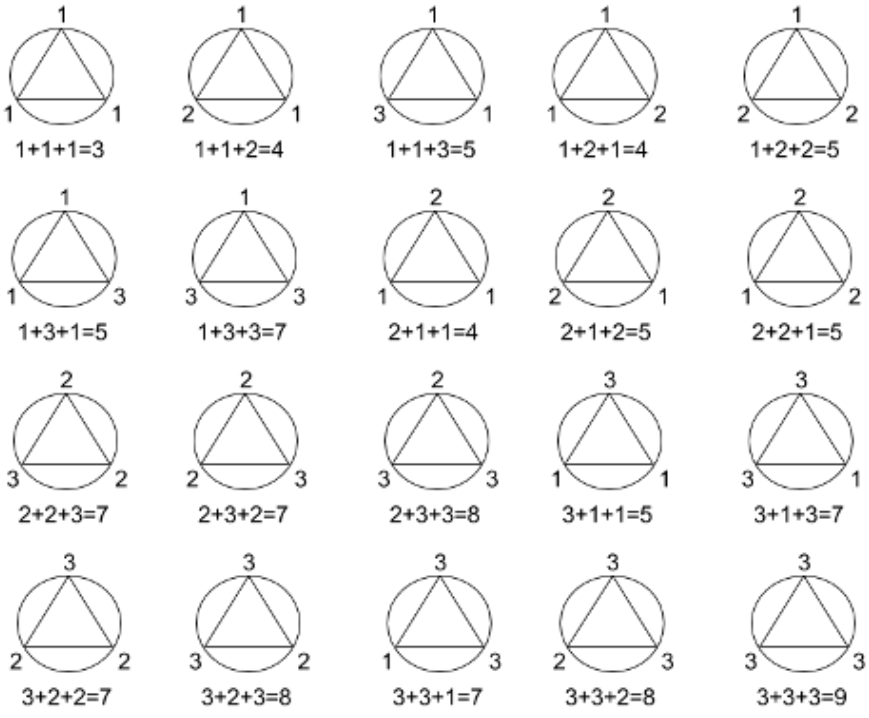
Output

Print a single number — the remainder of the number of variants of distributing coins that satisfy Tanya modulo $10^9 + 7$.

input
1
output
20

input
2
output
680

20 ways for $n = 1$ (gnome with index 0 sits on the top of the triangle, gnome 1 on the right vertex, gnome 2 on the left vertex):



D. Counting Arrays

3 seconds, 256 megabytes

You are given two positive integer numbers x and y. An array F is called an *y-factorization* of x iff the following conditions are met:

- There are y elements in F, and all of them are integer numbers;
- $\prod_{i=1}^{y} F_i = x$.

You have to count the number of pairwise distinct arrays that are y-factorizations of x. Two arrays A and B are considered different iff there exists at least one index i ($1 \le i \le y$) such that $A_i \neq B_i$. Since the answer can be very large, print it modulo $10^9 + 7$.

Input

The first line contains one integer q ($1 \le q \le 10^5$) — the number of testcases to solve.

Then q lines follow, each containing two integers x_i and y_i ($1 \le x_i, y_i \le 10^6$). Each of these lines represents a testcase.

Output

Print q integers. i-th integer has to be equal to the number of y_i-factorizations of x_i modulo $10^9 + 7$.

input
2 6 3 4 2
output
36 6

In the second testcase of the example there are six y-factorizations:

- $\{-4, -1\}$;
- $\{-2, -2\}$;
- $\{-1, -4\}$;
- $\{1, 4\}$;
- $\{2, 2\}$;
- $\{4, 1\}$.

E. The Sum of the k-th Powers

2 seconds, 256 megabytes

There are well-known formulas: $\sum_{i=1}^{n} i = 1 + 2 + \ldots + n = \frac{n(n+1)}{2}$, $\sum_{i=1}^{n} i^2 = 1^2 + 2^2 + \ldots + n^2 = \frac{n(2n+1)(n+1)}{6}$, $\sum_{i=1}^{n} i^3 = 1^3 + 2^3 + \ldots + n^3 = \left(\frac{n(n+1)}{2}\right)^2$. 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 ($1 \le n \le 10^9$, $0 \le k \le 10^6$).

Output

Print the only integer a — the remainder after dividing the value of the sum by the value $10^9 + 7$.

input
4 1
output
10

input
4 2
output
30

input
4 3
output
100

input
4 0
output
4

F. Fuzzy Search

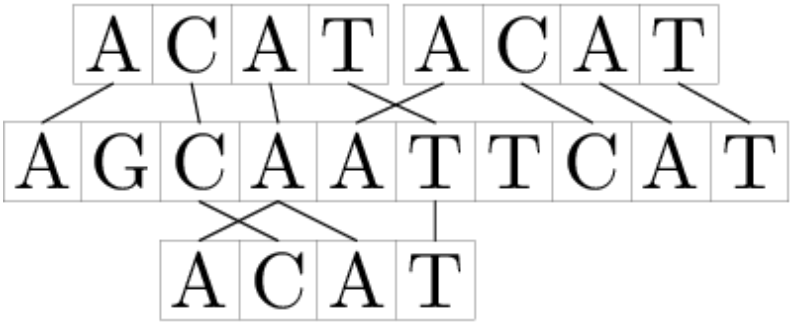
3 seconds, 256 megabytes

Leonid works for a small and promising start-up that works on decoding the human genome. His duties include solving complex problems of finding certain patterns in long strings consisting of letters 'A', 'T', 'G' and 'C'.

Let's consider the following scenario. There is a fragment of a human DNA chain, recorded as a string S. To analyze the fragment, you need to find all occurrences of string T in a string S. However, the matter is complicated by the fact that the original chain fragment could contain minor mutations, which, however, complicate the task of finding a fragment. Leonid proposed the following approach to solve this problem.

Let's write down integer $k \geq 0$ — the error threshold. We will say that string T occurs in string S on position i ($1 \leq i \leq |S| - |T| + 1$), if after putting string T along with this position, each character of string T corresponds to the some character of the same value in string S at the distance of at most k. More formally, for any j ($1 \leq j \leq |T|$) there must exist such p ($1 \leq p \leq |S|$), that $|(i + j - 1) - p| \leq k$ and $S[p] = T[j]$.

For example, corresponding to the given definition, string "ACAT" occurs in string "AGCAATTCAT" in positions 2, 3 and 6.



Note that at $k = 0$ the given definition transforms to a simple definition of the occurrence of a string in a string.

Help Leonid by calculating in how many positions the given string T occurs in the given string S with the given error threshold.

Input

The first line contains three integers $|S|$, $|T|$, k ($1 \leq |T| \leq |S| \leq 200\,000$, $0 \leq k \leq 200\,000$) — the lengths of strings S and T and the error threshold.

The second line contains string S.

The third line contains string T.

Both strings consist only of uppercase letters 'A', 'T', 'G' and 'C'.

Output

Print a single number — the number of occurrences of T in S with the error threshold k by the given definition.

input
10 4 1 AGCAATTCAT ACAT
output
3

If you happen to know about the structure of the human genome a little more than the author of the problem, and you are not impressed with Leonid's original approach, do not take everything described above seriously.

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 $\varphi(\varphi(...\varphi(n)))$, where function φ 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 $\varphi(\varphi(...\varphi(n)))$. Print the result in the canonical decomposition into prime factors.

Input

The first line contains integer m ($1 \le m \le 10^5$) — the number of distinct prime divisors in the canonical representaion of n.

Each of the next m lines contains a pair of space-separated integers p_i, a_i ($2 \le p_i \le 10^6$; $1 \le a_i \le 10^{17}$) — 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 ($1 \le k \le 10^{18}$).

Output

In the first line, print integer w — the number of distinct prime divisors of number $\varphi(\varphi(...\varphi(n)))$, where function φ is taken k times.

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

input
1 7 1 1
output
2 2 1 3 1

input
1 7 1 2
output
1 2 1

input
1 2 10000000000000000 10000000000000000
output
1 2 90000000000000000

You can read about canonical representation of a positive integer here:
http://en.wikipedia.org/wiki/Fundamental_theorem_of_arithmetic.

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