
Problem A. Make ZYB Happy

Input file: **standard input**
Output file: **standard output**
Time limit: **2 seconds**
Memory limit: **256 megabytes**

It's known to all that ZYB is godlike, so obviously he has a large number of titles, such as **jsking**, **bijingzyb** and **nbazyb**. ZYB likes his titles very much.

Each of ZYB's titles is a string consisting of lower case letters 'a'-'z' associated with a happiness value h_i , which shows how much ZYB likes this title. If you say any substring of some title with happiness value x , he will get x happiness points. Moreover, a string may appear in more than one title. In this case, the happiness points ZYB gets are multiplied. If the string you say is not the substring of any of his titles, he gets no happiness point.



For example, let's say ZYB has two titles: **zybnb** (with happiness value 3) and **ybyb** (with happiness value 5). If you say **y**, **b** or **yb**, ZYB will get 15 happiness points; if you say **z**, **zy** or **zyb**, ZYB will only get 3 happiness points; if you say **ybz** or **ybac** he will get 0 happiness points.

One day, you find ZYB pretty sad. As a big fan of ZYB, you want to say a word to ZYB to cheer him up. However, ZYB is really busy, so you can only say no more than m letters. As you haven't seen ZYB for a long time, you are so excited that you forget what you want to say, so you decide to choose to say a nonempty string no longer than m and only containing 'a'-'z' with equal probability. You want to know the expectations of happiness points you will bring to ZYB for different m .

Input

The first line contains an integer n ($1 \leq n \leq 10000$), the number of titles ZYB has.

The i -th of the next n lines contains a nonempty string t_i , which only contains lower case letters 'a'-'z', representing the i -th title. The sum of lengths of all titles does not exceed 3×10^5 .

Then follows a line with n integers h_i ($1 \leq h_i \leq 10^6$), the happiness value of i -th title.

The next line is a single integer Q ($1 \leq Q \leq 3 \times 10^5$), the number of queries.

For the next Q lines, each contains a single integer m ($1 \leq m \leq 10^6$), meaning that you can say no more than m letters to ZYB.

The input data contains only one test case.

Output

For each query, display a single line of integer, representing the answer. It can be proved that the answer can be uniquely written as p/q where p and q are non-negative integers with $\gcd(p, q) = \gcd(q, 10^9 + 7) = 1$, and you should display $p \cdot q^{-1} \bmod (10^9 + 7)$, where q^{-1} means the multiplicative inverse of q modulo $10^9 + 7$.

Example

standard input	standard output
2	769230776
zybnb	425925929
ybyb	891125950
3 5	633120399
4	
1	
2	
3	
4	

Note

For the first query, you can bring him 3 happiness points if you say **z** or **n**, and 15 happiness points if you say **y** or **b**; all other strings of length 1 bring no happiness point to ZYB. Therefore, the expectation is $(2 \times 3 + 2 \times 15)/26 = 18/13$, and the answer is $18 \times 13^{-1} \bmod (10^9 + 7) = 769230776$.