

Contando Ambiguidades

Por Gabriel Poesia, UFMG  Brazil**Timelimit: 6**

Enquanto vinham para a Escola de Verão de ônibus, um competidor atento notou uma placa que continha uma propaganda com a seguinte URL: `www.programs.healthyfood.ma` (ele suspeita que o `.ma` seja para "Marte", mas isto é improvável). Por estarem discutindo as novas features de C++11, o competidor e um amigo pensaram que o "programs" na URL fosse relacionado ao resultado da programação de computadores: programas. Porém, acontece que a URL foi infeliz: era, na verdade, a junção de duas palavras: o adjetivo "pro", que em inglês significa "em favor de", e o substantivo "grams", um tipo de legumes usados em pratos de comida. Isto explica o fato de a URL ter levado os dois amigos a uma página Web contendo fotos de mulheres saudáveis, ao invés de tutoriais sobre *variadic templates*!

Depois de rirem do acontecido, os dois amigos estão curiosos sobre essas ambiguidades em URLs. Eles começaram a olhar para todas as placas na estrada e a escrever as URLs que elas contém. Porém, eles rapidamente obtiveram muitas URLs para verificar - muito mais que um programador aceita fazer "na mão". Por isto, eles querem escrever um programa que, dada a lista de palavras que eles conhecem, conta de quantas formas distintas uma certa palavra pode ser obtida pela concatenação de uma ou mais palavras da lista.

Ambos são programadores muito habilidosos e já projetaram o algoritmo perfeito para este problema. O principal problema é que os laptops deles estão sem bateria, e eles não querem programar em seus smartphones (não há acesso à Internet nas estradas para acessar um computador com um compilador remotamente). Porque você está sentado próximo a eles, e seu laptop tem bateria para mais de 10 minutos (suficiente, não?) de programação, eles pediram para que você escrevesse tal programa para eles. É claro que eles não contaram a você o algoritmo que projetaram, porque confiam nas suas próprias habilidades de resolução de problemas. Você pode escrever o programa para eles?

Note que uma palavra pode ser usada várias vezes no processo de concatenação. Por exemplo, se os amigos conhecem as palavras "a" e "aa", eles podem formar a palavra "aaa" dos seguintes 3 jeitos (aqui, + significa concatenação).

- "a" + "a" + "a"
- "a" + "aa"
- "aa" + "a"

Assim, a resposta para este caso seria 3.

Da mesma forma, há duas maneiras de se obter a palavra "aa":

- "a" + "a"
- "aa"

Note que uma palavra pode ser usada sozinha, se os amigos a conhecem.

Entrada

A entrada contém vários casos de teste. Cada caso de teste começa com uma linha contendo um inteiro positivo **D** (no máximo 10^5), o número de palavras que os amigos conhecem. As próximas **D** linhas contém essas palavras. Cada linha contém uma palavra distinta. Depois, há uma linha contendo um inteiro positivo **U** (no máximo 1000), que é o número de palavras que os competidores escreveram. Após esta linha, há **U**

linhas, cada uma contendo uma dessas palavras.

Toda palavra no arquivo de entrada é composta por no máximo 100 caracteres alfabéticos.

Saída

Para cada caso de teste imprima **U** linhas. A *i*-ésima linha no arquivo de saída deve conter o número de maneiras distintas de se obter a *i*-ésima palavra que os dois competidores escreveram pela concatenação de palavras que eles conhecem. Este número pode ser muito grande. Por isto, imprima o resto da divisão da resposta por $10^9 + 7$.

Exemplo de Entrada	Exemplo de Saída
3 programs pro grams 1 programs	2