

Program Name: `typing.cpp`Input File: `typing.dat`

In this problem, you will be writing the piece of a word processor that “looks ahead” for a typist and suggests a word. Typically, the look-ahead function would receive a prefix input from the word processor expecting a returned word. However, your program will use flat text input files.

This look-ahead function selects the shortest word beginning with the given prefix from a dictionary. If multiple words qualify as the shortest, the look-ahead function will choose the first one in the dictionary. For example, given the following dictionary segment and the prefix “reg”, your program would select “register” because it is the shortest of the words starting with “reg”.

Dictionary Words	Possible Match (Y/N)	Word Length
...	N	Not Applicable
red	N	Not Applicable
register	Y	8
regular	Y	7
regulate	Y	8
regulation	Y	10
reheat	N	Not Applicable
...	N	Not Applicable

Input

There are two components to the input file: the dictionary and the prefixes. The dictionary component begins on Line 1. Line 1 of the input file gives the number of dictionary entries as a single integer ($1 \leq D \leq 100$) beginning in column 1. Lines 2 through $D+1$ each contain a single dictionary entry. Entries consist of lower case alphabetic characters and are between 1 and 20 characters long. The entries are sorted alphabetically. The prefixes begin on the line after the last dictionary entry, $D+2$. Each of these lines contains a single prefix. Prefixes consist of lower case alphabetic characters and are between 1 and 20 characters long.

Output

For each prefix in the input file, your program will print a response, one per line, to the screen. If no word in the dictionary matches the prefix, your program should print the prefix. Otherwise, the program should print the best look-ahead word from the dictionary.

Example: Input File

```
11
acropolis
read
red
regal
regalia
register
regular
regulate
regulation
reheat
zebra
regu
regali
ra
```

Output to screen

```
regular
regalia
ra
```