Problem J
Progressive Scramble

Problem ID: progressivescramble
CPU Time limit: 1 second
Memory limit: 1024 MB

Author: Mario Nakazawa
Source: 2017 ICPC North American Qualifier Contest

You are a member of a naive spy agency. For secure communication, members of the agency use a very simple encryption algorithm – which changes each symbol in the message 'progressively', i.e., based on the symbols preceding it. The allowed symbols are space and the 26 lowercase English letters. For encryption purposes we assign them the values 0 (for space) and 1 through 26 (for a–z). We'll let $v(s)$ represent the numeric value of symbol s.

Photo by Chilanga Cement

Consider a message with symbols $s_1, s_2, \ldots, s_n$. The encryption algorithm starts by converting the first symbol s_1 into its associated value $u_1 = v(s_1)$. Then for each subsequent symbol s_i in the message, the computed value is $u_i = v(s_i) + u_{i-1}$ — the sum of its associated value and the computed value for the previous symbol. (Note that when there is a space in the input message, the previous scrambled letter is repeated.) This process continues until all the u_i are computed.

At this point, the message is a sequence of numeric values. We need to convert it back to symbols to print it out. We do this by taking the value u_i modulo 27 (since there are 27 valid symbols), and replacing that value with its corresponding symbol. For example, if $u_i = 32$, then $32 \bmod 27 = 5$, which is the symbol 'e' (since $v(e) = 5$).

Let's look at an example. Suppose we want to encrypt the string "my pie".

1. First, convert each symbol s_i into $v(s_i)$: $[13, 25, 0, 16, 9, 5]$.
2. Next, compute each u_i : $[13, 38, 38, 54, 63, 68]$.
3. Then, use modulus on the u_i : $[13, 11, 11, 0, 9, 14]$.
4. Finally, convert these back to symbols: "mkk in".

Create a program that takes text and encrypts it using this algorithm, and also decrypts text that has been encrypted with this algorithm.

Input

The input to your program consists of a single integer $1 \leq n \leq 100$ on its own line. This number is followed by n lines, each containing the letter 'e' or 'd', a single space, and then a message made up of lowercase letters (a–z) and spaces, continuing to the end of the line. Each message is between 1 and 80 characters long. The letters 'd' and 'e' indicate that your program decrypts or encrypts the subsequent string, respectively.

Output

Output the result of encrypting or decrypting each message from the input on its own separate line. Note that differences in whitespace are significant in this problem. Therefore your output must match the correct output character-for-character, including spaces.

Sample Input 1

```
7
e testing multiple letters rrrrrrrrrrrrr
e this particularly long  sentence can test encryption
d tajbbrsjcloiuvmywwhwjqqqinauzmpuuxyllejbvv nqhfvoxlz
e my pie
d mkk in
e the quick brown fox jumps over the lazy dog
d taffwqzbmmofuqddjyvvezlatthchzzs eeqrqoosgn
```

Sample Output 1

```
tyqjsfmmzteygwhmmycwpulddvmdvmdvmdvmdv
tajbbrsjcloiuvmywwhwjqqqinauzmpuuxyllejbvv nqhfvoxlz
this particularly long  sentence can test encryption
mkk in
my pie
taffwqzbmmofuqddjyvvezlatthchzzs eeqrqoosgn
the quick brown fox jumps over the lazy dog
```