Problem 2 – Moving Letters

Nakov is a keen fan of cryptography. When he was a child, he invented an encryption algorithm called “**Moving Letters**”. It performs two steps: **extracts the letters** from the input message and **moves each letter** a few positions on the right according to its alphabetical order. Your task is to write a program that reads an input message, encrypts it through the “**Moving Letters**” algorithm and prints the result.

Extracting the Letters

The input message is given as a sequence of words separated by a space. The words are converted into a **sequence of letters** the following way: until all words disappear, the last letter of each word (if exists) is removed from the word and is appended to the output sequence of letters.

Moving the Letters

The next step in the encryption algorithm is to move each letter (from positions 0, 1, …, n-1) on the right **k** times. The number **k** is taken from the number of the letter in the Latin alphabet regardless of its casing ('a' → 1, 'b' → 2, …, 'z' → 26). When a letter is moved to the right, if it is the last letter of the sequence, its next position is the first position in the sequence, just before all the letters.

Example

Let's the input is "**`Fun exam right`**". It is first split into 3 words: {"**`Fun`**", "**`exam`**", "**`right`**"}. Then the last letters are extracted: {"**`Fun`**", "**`exam`**", "**`right`**"} → "**`nmt`**" → {"**`Fu`**", "**`exa`**", "**`righ`**"} → "**`nmtuah`**" → {"**`F`**", "**`ex`**", "**`rig`**"} → "**`nmtuahFxg`**" → {"**`e`**", "**`ri`**"} → "**`nmtuahFxgei`**" → {"**`r`**"} → "**`nmtuahFxgeir`**".

The moving of letters starts from the input sequence "**`nmtuahFxgeir`**" and sequentially moves right its first letter (at position 0), then its second letter (at position 1), and so on, and finally moves right its last letter (at position 11). First the letter '**`n`**' at position **`0`** is moved right **14** times: "**`nmtuahFxgeir`**" → "**`mtnuahFxgeir`**". Then the letter '**`t`**' at position **`1`** is moved **20** times right: "**`mtnuahFxgeir`**" → "**`mnuahFxgetir`**". Then the process continues: "**`mnuahFxgetir`**" → "**`mnahFxgetiru`**" → "**`mnaFxgetiruh`**" → "**`mnaFxgetiruh`**" → "**`gmnaFxetiruh`**" → "**`gmnaFxtiruhe`**" → "**`gmnaiFxtruhe`**" → "**`gmrnaiFxtuhe`**" → "**`gmrnaiuFxthe`**" → "**`gmrnaihuFxte`**" → "**`gmrneaihuFxt`**". The result is "**`gmrneaihuFxt`**".

Input

The input data should be read from the console. It consists of a single line holding a sequence of words separated by a single space (followed by the "end of line" character).

Output

The output data consists of a single text line holding the obtained result.

Constraints

- The input will be less than **256 KB** and will hold Latin letters separated by spaces.
- Allowed working time for your program: **0.35 seconds**.
- Allowed memory: **64 MB**.

Sample Input and Output

Input	Output
`Fun exam right`	`gmrneaihuFxt`

Input	Output
`Telerik Academy`	`AymlTiedkaerec`

Input	Output
`Hi exam`	`maiHex`