Reverse Rot

Problem ID: reverserot
CPU Time limit: 1 second
Memory limit: 1024 MB
Difficulty: 1.4

Author(s): James Feher
Source: ACM/ICPC Mid-Centr
USA 2014 Regional Contest

A very simplistic scheme, which was used at one time to encode information, is to rotate the characters within an alphabet and rewrite them. ROT13 is the variant in which the characters A-Z are rotated 13 places, and it was a commonly used insecure scheme that attempted to "hide" data in many applications from the late 1990's and into the early 2000's.

It has been decided by Insecure Inc. to develop a product that "improves" upon this scheme by first reversing the entire string and then rotating it. As an example, if we apply this scheme to string `ABCD` with a reversal and rotation of 1, after the reversal we would have DCBA and then after rotating that by 1 position we have the result `EDCB`.

Your task is to implement this encoding scheme for strings that contain only capital letters, underscores, and periods. Rotations are to be performed using the alphabet order:

```
ABCDEFGHIJKLMNOPQRSTUVWXYZ_.
```

Note that underscore follows Z, and the period follows the underscore. Thus a forward rotation of 1 means `A` is shifted to `B`, that is:

- `A` $\rightarrow$ `B`
- `B` $\rightarrow$ `C`
- $\vdots$
- `Z` $\rightarrow$ `_`
- `_` $\rightarrow$ `.`
- `.` $\rightarrow$ `A`

Likewise a rotation of 3 means:

- `A` $\rightarrow$ `D`
- `B` $\rightarrow$ `E`
- $\vdots$
- `.` $\rightarrow$ `C`

Input

The input is composed of T test cases ($1 \leqslant T \leqslant 30$), each on one line.

Each input line will consist of an integer N, followed by a string. N is the amount of forward rotation, such that $1 \leqslant N \leqslant 27$. The string is the message to be encrypted, and will consist of 1 to 40 characters, using only capital letters, underscores, and periods.

The end of the input will be denoted by a final line with only the number `0`.

Output

For each test case, display the "encrypted" message that results after being reversed and then shifted.

Sample Input 1

```
1 ABCD
3 YO_THERE.
1 .DOT
14 ROAD
9 SHIFTING_AND_ROTATING_IS_NOT_ENCRYPTING
2 STRING_TO_BE_CONVERTED
1 SNQZDRQDUDQ
0
```

Sample Output 1

```
EDCB
CHUHKWBR.
UPEA
ROAD
PWRAYF_LWNHAXWH.RHPWRAJAX_HMWJHPWRAORQ.
FGVTGXPQEAGDAQVAIPKTVU
REVERSE_ROT
```