Caesar Cipher

*Extracted from:*WEGA 00013
Source file name: `ccaesar.py`
Time limit: 1 second

Julius Caesar protected his confidential information by encrypting it in a cipher. Caesar's cipher rotated every letter in a string by a fixed number, K, making it unreadable by his enemies. Given a string, S, and a number, K, encrypt S and print the resulting string.

Note: The cipher only encrypts letters; symbols, such as (−,?,#, ...), remain unencrypted.

Input

The first line contains an integer, N, which is the number of strings to be encrypted. The next line containg unencrypted string, S_1. The next line contains the integer encryption key, K_1, which is the number of letters to rotate, and so on for the N strings. The string S is valid and doesn't contain spaces. $(0 \leq K \leq 100)$

The input must be read from standard input.

Output

For each test case, print the encoded string.

The output must be written to standard output.

Sample Input 1	Sample Output 1

`2`
`We-are-in-war`
`15`
`we-Need-Help`
`8`
	`Case1 = Lt-pgt-xc-lpg`
	`Case2 = em-Vmml-Pmtx`

Sample Input 2	Sample Output 2

`5`
`middle-Outz`
`2`
`Pz-/aI/J‘EvfthGH`
`66`
`D3q4`
`0`
`159357lcfd`
`98`
`Hello_World!`
`4`
	`Case1 = okffng-Qwvb`
	`Case2 = Dn-/oW/X‘SjthvUV`
	`Case3 = D3q4`
	`Case4 = 159357fwzx`
	`Case5 = Lipps_Asvph!`

Explanation

Each unencrypted letter is replaced with the letter occurring K spaces after it when listed alphabetically. Think of the alphabet as being both case-sensitive and circular; if K rotates past the end of the alphabet, it loops back to the beginning (i.e.: the letter after z is a , and the letter after Z is A).

This problem is based on Hackerrank: `https://www.hackerrank.com/challenges/caesar-cipher-1/problem`