## Encode and Encrypt

You are currently employed as a Junior MERIN-J (Management of Expression Research, Identification and Normalization – Job). One of your first tasks is to handle electronic messaging in your company, by providing a safe way to transfer messages. Not only should the messages be **encrypted** (made unreadable for anyone who does not know the cipher) in some way. The CEOs don't want to be wasting extra money on Broadband, so you also have to take care of **compressing (encoding)** the messages.

Since you don't have much time (about 6 hours or so), you need to think up of something quickly. Luckily, you have an old article on encryption and encoding, stating the following:

* We are given a **message** and a **cypher**
  + The message is the text the user wants to transmit
  + The cypher is a string which is used to encrypt and decrypt the message
  + The encrypted message is called **cypherText**
* We define a function **Encrypt**, which takes a message and a cypher:
  + It iterates over the symbols in the message and the cypher
  + For each **pair of symbols**, it takes their **codes** (in the table below) and computes the **bitwise XOR** of the **symbol in the message** with the **symbol in the cypher**.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 |

* + The **resulting code** is **summed** with the **ASCII code of the character 'A'** (65), giving a new ASCII code
  + The **character corresponding to this new ASCII code** is the **encrypted representation** of the respective **character in the message**
  + If the cypher string is shorter than the message, using it symbols loops to the beginning of the cypher. E.g. for a message "**ABCDE**" and a cypher "**PQR**" we will have:
    - **'A' encrypted with 'P'** = 'P', **'B' encrypted with 'Q'** = 'R', **'C' encrypted with 'R'** = 'T', **'D' encrypted with 'P'** = 'M', **'E' encrypted with 'Q'** = 'U'
  + If the message string is shorter than the cypher, some of its symbols will be encoded several times, until all of the cypher symbols are used.
    - E.g. for a message "**ABC**" and a cypher "**PQRST**", we will have:
    - **'A' encrypted with 'P'** and **the result** ('P') **encrypted with 'S'** = '^' (ASCII 94),
    - **'B' encrypted with 'Q'** and **the result** ('R') **encrypted with 'T'** = 'C',
    - **'C' encrypted only with 'R'** = 'T'
* We define a function **Encode**, which takes a string of text to compress:
  + It looks for sequences of symbols which are the same(e.g. 'aaaaa')
  + For each sequence of same symbols, the function replaces the sequence with a number representing the count of repeated symbols, followed immediately by the symbol which is repeated. This is called run-length encoding. E.g. for the text "aaaabbbccccaa" we will have "4a3b4caa".
    - The function **replaces symbols** in the aforementioned way **ONLY** **if the run-length encoding** of the **same-symbol sequence** is **shorter than the sequence itself**
    - That's why in the example above the last two a's remain the same – '2a' has the same length as 'aa'
* Given the two functions, and given a message and a cypher, the produced result should be:
  + **Encode(Encrypt(message, cypher) + cypher) + lengthOfCypher**
    - + denotes concatenation, the two functions return strings and 'lengthOfCypher' is a number, which is equal to the number of symbols in the cypher
    - i.e. the message is encrypted with the cypher, then the cipher is added for decrypting purposes, then the result is compressed and a number denoting the length of the cypher before compression is added to the compressed string

Write a program, **using the method above**, which encodes and encrypts a message with a cypher.

### Input

The input data should be read from the console.

On the **first line** of the input, there will be a single string, representing the **message**.

On the **second line** of the input, there will be a single string, representing the **cypher**.

The input data will always be valid and in the format described. There is no need to check it explicitly.

### Output

The output data should be printed on the console. Print exactly one line – the cyphered message.

### Constraints

* All symbols in the message will be **capital English letters**
* Тhe message and the cypher will be no more than 1500 symbols each
* Allowed working time for your program: 0.1 seconds. Allowed memory: 16 MB.

### Examples

|  |  |
| --- | --- |
| **Input example** | **Output example** |
| TELERIKACADEMY  SOFTWARE | BKOXHI\EQOGX[YSOFTWARE8 |
| AAABB  BBBBBBA | ABBAA6BA7 |
| JOHNY  DEPPP | KKICXDE3P5 |