

Decryption

— Problem Description

Decryption is the process of converting encrypted data into a required message. It is nothing but a reverse process of encryption. There are many encryption techniques to protect data. Few of them are SSL, AES, DES etc. Every encryption technique has a key associated with it to provide more security and uniqueness to the encrypted data.

Here the decryption is done in the following way:

- The maximum length of the encrypted string can be 32 characters.
- If the length of the encrypted string is greater than 16 but less than 32, then pad the string with a space character to make it 32 characters long. If the encrypted string is exactly 16 characters, no need of any padding
- Split the 32-characters padded encrypted string into two parts of 16 characters each
- Similarly, if the length of the encrypted string is less than 16, then pad it with a space character to make it 16 characters long
- Convert each character of all such 16-characters long strings into their corresponding ASCII values. ASCII values can be obtained at <https://www.ascii-code.com/>
- Arrange the ASCII values of each such string of 16 characters row-wise in a 4 X 4 matrix. This matrix is called as M_E
- To convert the encrypted string to a decrypted/clear text format, a key will be required. The key will be provided as input. The maximum length of key string can be 16 characters
- The key matrix is also obtained by following the same procedure used to arrive at encrypted matrix M_E i.e.
 - If the length of the key is less than 16 characters, append a space character (' ') at the end of the string
 - Take ASCII values of each such character and arrange them row-wise in a 4 X 4 matrix
 - The matrix thus obtained is termed as M_K
- Next, the encrypted matrix M_E is XOR-ed with rotated key matrix
- The rotated key matrix is obtained by rotating the key matrix M_K such that the outer elements of key matrix are rotated anti-clockwise, and the inner elements are rotated clockwise, in one rotation
- Now the resultant matrix obtained from the previous step is XOR-ed with key matrix M_K
- This matrix is then XOR-ed with decimal integer 4
- Now each element row-wise is converted back into character value and then joined together, if needed, to produce a decrypted text

Note: only the corresponding elements can be XOR-ed (i.e. $A_{ij} \wedge B_{ij}$).

So, the task is to decrypt the encrypted string using the key string and display the decrypted string.

— Constraints

$0 < \text{Length of key string} \leq 16$

$0 < \text{Length of encrypted string} \leq 32$

— Input

First line contains an Encrypted string E which needs to be decrypted.

Second line contains a key string K which is to be used to convert encrypted string into decrypted string using the process of decryption stated above.

— Output

Display the decrypted string.

— Time Limit (secs)

1

— Examples

Example 1

Input

\$we&vmwlje\$\$\$\$\$

sai

Output

saikrishna

Explanation

Given '\$we&vmwlje\$\$\$\$\$' is an encrypted string E. Since length (E) is 16, no need to append spaces at the end. Key string is 'sai', since its length is less than 16 pad with space character to make it 16 characters long.

Given '\$we&vmwlje\$\$\$\$\$' is arranged row-wise in a 4 X 4 matrix form as

36 119 101 38

118 109 119 108

106 101 36 36

36 36 36 36

Here \$, w, e and & are 36, 119, 101 and 38 respectively. Similarly for the remaining characters. Similar structure needs to be applied to the key matrix.

Obtain matrices M_E and M_K using decryption procedure mentioned above. Once M_K is obtained, rotated M_K can be easily obtained. Finally, $M_E \wedge \text{Rotated } M_K \wedge M_K \wedge 4$ will give the resultant matrix. This matrix elements converted from ASCII values to ASCII characters will produce the decrypted string 'saikrishna'. Hence, the output is 'saikrishna'.

Example 2

Input

bkn`.bc mnohijktr{~p>rs0)~\$\$\$\$\$

hello

output

abcdefghijklmnopqrstuvwxyz

Explanation

Here the encrypted string length is 32 instead of 16. Also, the key string is different. Everything else to obtain decrypted string from encrypted string remains same.