Break me (`breakme`)

One of the simplest and most ancient encryption schemes is the ROT-k cipher, which works by replacing each letter in a text with another letter that is k positions away in the alphabet, wrapping around if k leads to a character past the end of the alphabet. For k=3, we have:

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
D E F G H I J K L M N O P Q R S T U V W X Y Z A B C

For instance, the ROT-3 encoding of "HELLO" is "KHOOR". According to Suetonius 1 , ROT-3 was used by Caesar to communicate secret messages to Cicero and to his own relatives back in the days of the Roman republic. Unfortunately, ROT-k ciphers are easy to break if one knows in which language the original text was written. This can be done as follows:

- for each possible value of k (e.g., from 0 to 25 for the English alphabet) decrypt the input text assuming it was encrypted with ROT-k, obtaining a version text k . Clearly, only one version will be equal to the original text. We just don't know which one yet.
- compute the k that minimizes the following formula (cross-entropy):

$$H_k(p_k, q) = -\sum_{c \in text_k} p_k(c) * \log q(c)$$

  where p k (c) is the frequency 2 of letter c in text k and q(c) is the frequency of c in the language of the text (e.g., English).
- output text k

Write a program that, given an English text encrypted with ROT-k for some unknown k in [0,25], automatically decrypts it by finding the k that minimizes the cross-entropy as explained above.

Here are the frequencies q of the 26 letters of the English alphabet:

```
0.08167, 0.01492, 0.02782, 0.04253, 0.12702, 0.02228, 0.02015,
0.06094, 0.06966, 0.00153, 0.00772, 0.04025, 0.02406,
0.06749, 0.07507, 0.01929, 0.00095, 0.05987, 0.06327, 0.09056,
0.02758, 0.00978, 0.02360, 0.00150, 0.01974, 0.00074
```

For instance, letter 'A' (regardless of its case) has a frequency of 8.167%, i.e., occurs 8.167% of the times in typical English text. Letter 'Z', on the other hand, occurs just 0.074% of the times.

Implementation

You should submit a single file, with either a `.c`, `.cpp`, `.java` or `.py` extension.

Your program must read input data from `stdin` and write the output data into `stdout`.

`stdin` consists of only one line:

- Line 1: The ROT-k encrypted english text.

`stdout` consists of only one line:

- Line 1: The decrypted text.

Constraints

- The encrypted text is at most 4096 characters long.

Scoring

Your program will be tested against 10 testcases, each of which is worth 10 points.

Examples

stdin	stdout
`Va pelcgbtencul, n Pnrfne pvcure, nyfb xabja nf Pnrfne'f pvcure, gur fuvsg pvcure, Pnrfne'f pbqr be Pnrfne fuvsg, vf bar bs gur fvzcyrfg naq zbfg jvqryl xabja rapelcgvba grpuavdhrf.`	`In cryptography, a Caesar cipher, also known as Caesar's cipher, the shift cipher, Caesar's code or Caesar shift, is one of the simplest and most widely known encryption techniques.`