Problem B
Bicryption
Time limit: 2 seconds



The enigma, an encryption machine that solves this problem.
CC BY 2.0 by William Warby on Wikimedia Commons

After AKing[1] the practice session, the team Alternative Better Encryption are discussing the solutions to the problems. In particular, they were underwhelmed to see yet another multi-pass problem themed around encryption.

Instead of the boring encryption protocol of that problem, they invent an alternative and better encryption for secure communication to use in the contest tomorrow. They are calling this their *Newer Well-Encrypted Robust Communication*.

Anna and Ben each encrypt a word to send to Charlotte to test their new invention. These may be two different words. To encrypt a word w, they use two words x and y of the same length as w, where the ith letters of all three words are pairwise distinct for all $1 \leq i \leq |w|$. Charlotte then receives two words x' and y', which are either the two encrypted words from Anna or those from Ben. These words may be out of order, that is, she only knows that either $x' = x$ and $y' = y$, or $x' = y$ and $y' = x$.

Find a strategy for Anna and Ben to encrypt their words such that Charlotte can decrypt the words again. Charlotte should also determine who of Anna or Ben sent the word.

Input

The input consists of:

- One line with "`Anna`", "`Ben`", or "`Charlotte`", representing pass 1, 2, and 3 respectively.
- One line, depending on the pass:
  - **Pass 1:** A string w_a ($1 \leq |w_a| \leq 10^5$), the word Anna encrypts.
  - **Pass 2:** A string w_b ($1 \leq |w_b| \leq 10^5$), the word Ben encrypts.
  - **Pass 3:** Two strings x' and y' ($1 \leq |x'| = |y'| \leq 10^5$), the encrypted words of either Anna or Ben, in arbitrary order.

All input strings consist of only English lowercase letters.

The input strings w_a and w_b are fixed before the interaction, the input of Pass 3 might be chosen adaptively and adversarially depending on your outputs of the first two passes.

This is a multi-pass problem. Your program will be invoked three separate times per test case.

A testing tool is provided to help you develop your solution.

Output

- **Pass 1:** Output two strings x_a and y_a ($|x_a| = |y_a| = |w_a|$) that encode w_a.
- **Pass 2:** Output two strings x_b and y_b ($|x_b| = |y_b| = |w_b|$) that encode w_b.
- **Pass 3:** Output the name of the sender, either "`Anna`" or "`Ben`", and that person's original decrypted word.

[1] An alternative term for fully solving a problemset.

All output strings, except for the name, must consist exclusively of **English lowercase letters** and contain no spaces. The name may additionally contain uppercase letters.

In passes 1 and 2, the ith letters of w, x and y of that pass must all be distinct for $1 \leq i \leq |w|$, that is: $w_i \neq x_i, w_i \neq y_i, x_i \neq y_i$.

Sample Case 1

Sample Input	Pass 1	Sample Output
`Anna` `koeri`	`curyr wrust`	

Sample Input	Pass 2	Sample Output
`Ben` `plaintext`	`notreally encyrpted`	

Sample Input	Pass 3	Sample Output
`Charlotte` `notreally encyrpted`	`Ben` `plaintext`	

Sample Case 2

Sample Input	Pass 1	Sample Output
`Anna` `koeri`	`curyr wrust`	

Sample Input	Pass 2	Sample Output
`Ben` `plaintext`	`notreally encyrpted`	

Sample Input	Pass 3	Sample Output
`Charlotte` `wrust curyr`	`Anna` `koeri`	