Problem A

Some people say ‘The shortest distance between two points is a straight line.’ However, this depends on the distance metric employed. Between points (x1,y1) and (x2,y2), the Euclidean (aka straight-line) distance is

(x1−x2)2+(y1−y2)2

However, other distance metrics are often useful. For instance, in a city full of buildings, it is often impossible to travel in a straight line between two points, since buildings are in the way. In this case, the so-called Manhattan (or city-block) distance is the most useful:

|x1−x2|+|y1−y2|

Both Euclidean and city-block distance are specific instances of what is more generally called the family of p-norms. The distance according to norm p is given by

(|x1−x2|p+|y1−y2|p)1/p

If we look at Euclidean and Manhattan distances, these are both just specific instances of p=2 and p=1, respectively.

For p<1 this distance measure is not actually a metric, but it may still be interesting sometimes. For this problem, write a program to compute the p-norm distance between pairs of points, for a given value of p.

**Input**

The input file contains up to 1000 test cases, each of which contains five real numbers, x1 y1 x2 y2 p, each of which have at most 10 digits past the decimal point. All coordinates are in the range (0,100] and p is in the range [0.1,10]. The last test case is followed by a line containing a single zero.

**Output**

For each test case output the p-norm distance between the two points (x1,y1) and (x2,y2). Your answer may have absolute or relative error of at most 0.0001.

|  |  |
| --- | --- |
| **Sample Input 1** | **Sample Output 1** |
| 1.0 1.0 2.0 2.0 2.0  1.0 1.0 2.0 2.0 1.0  1.0 1.0 20.0 20.0 10.0  0 | 1.4142135624  2.0000000000  20.3636957882 |

CPU Time limit1 second

Memory limit1024 MB

Downloads

[Sample data files](https://sigmatechhungary.kattis.com/test/programming/zasror2fs6myvjydtrs3kxtfkthud46i/problems/4cfe1c30b3f1b81e/file/statement/samples.zip)

Author

Greg Hamerly

Problem B



*Photo by Pedro Ribeiro Simões (pedrosimoes7) on flickr*

The color of a cat is primarily decided by three genes, coding for Black (Eumelanin), Red (Phaeomelanin) and Dilution (Melanophilin). The cat genetics experts among the readers will note that we are ignoring a wide range of genes, e.g., Tyrosinase and the white masking gene – we apologize in advance for the oversimplifications that are to ensue.

Black is usually written as B, Red as O, Dilution as D. In general, a cat has two copies of each gene, one from each parent (except for the red gene which is sex-bound, we’ll return to that later). Each copy can be either *dominant*, which we’ll denote using an upper case letter, or *recessive*, which we’ll denote using a lower case letter. Thus for each gene there are three possibilities: either both copies of the gene are dominant, both are recessive, or one is dominant and the other is recessive. When describing a cat we’ll write this as a pair of letters, with the second one possibly being a “-” to indicate a wildcard. For example, “bb” indicates that the cat has two recessive black genes, and “B-” indicates that the cat has a dominant black gene and that we don’t care about whether the other one is dominant or recessive.

When a cat has no dominant red (O) gene, its coat will be a variation of black, according to the following four combinations:

|  |  |
| --- | --- |
| B-D-: Black | B-dd: Blue |
| bbD-: Chocolate | bbdd: Lilac |

Now, red (O or o) is sex bound on the X chromosone. This means a male cat, or tom, only has a single red gene, while a female has two. Red dominates black, meaning that a cat having an O gene will always have red coloring. Like black, red is also affected by the dilution gene. For male cats, we have the following two possible red colors (here, the black genes have no effect on the coat color):

|  |  |
| --- | --- |
| D-O: Red | ddO: Cream |

For female cats the situation gets a bit more complicated. If the female has two O genes she will be be red, with variations depending on the dilution gene in exactly the same way as for male red cats:

|  |  |
| --- | --- |
| D-OO: Red | ddOO: Cream |

However, a female cat with the Oo combination will have a partially black, partially red, coat. These cats are known as Torties (Tortoiseshells), and named after both their coat colours, starting with the black. The four possible Tortie colorings are thus as follows:

|  |  |
| --- | --- |
| B-D-Oo: Black-Red Tortie | B-ddOo: Blue-Cream Tortie |
| bbD-Oo: Chocolate-Red Tortie | bbddOo: Lilac-Cream Tortie |

When a pair of cats have kittens, the offspring will get a random set of genes from their parents according to the following simple process: for each gene, independendtly, one of the copies is selected uniformly at random from the father’s two copies, and the other copy is selected uniformly at random from the mother’s two copies. For the red gene, there is a 50% chance that the offspring will be female in which case it will get the father’s red gene (and one of the mother’s red genes at random), and a 50% chance that the offspring will be male in which case it will only get one of the mother’s red genes (at random).

Write a program which, given the colors of two cats, computes the most likely color for their offspring. For cats having a color where some genes do not affect the color, assume that the corresponding genes are uniformly random (over the two choices dominant/recessive). For example, if a cat is red, you should assume that the possibilities “DD” and “Dd” for the dilution gene are equally likely, and that the four possibilities “BB”, “Bb”, “bB” and “bb” are equally likely for the black gene (but note that “Bb” and “bB” are equivalent).

**Input**

The input consists of two lines. The first line gives the color of the female cat, and the second line gives the color of the male cat. Each color is spelled and capitalized exactly as when it is listed above, and the color is valid (i.e., the male cat won’t be a Tortie).

**Output**

Output a list of the possible color outcomes for the offspring of the two cats. Each line should contain a color followed by a number 0<p≤1 indicating the probability of this particular outcome.

The list should be sorted first in decreasing order by probability, then in alphabetic order by name (see sample input 3 below). You do not need to worry about the precise formatting of the probabilities (e.g., number of decimals), but the absolute error of each probability must be smaller than 10−9.

|  |  |
| --- | --- |
| **Sample Input 1** | **Sample Output 1** |
| Red  Red | Red 0.937500000  Cream 0.062500000 |

|  |  |
| --- | --- |
| **Sample Input 2** | **Sample Output 2** |
| Lilac-Cream Tortie  Blue | Blue 0.375000000  Cream 0.250000000  Blue-Cream Tortie 0.187500000  Lilac 0.125000000  Lilac-Cream Tortie 0.062500000 |

|  |  |
| --- | --- |
| **Sample Input 3** | **Sample Output 3** |
| Blue  Red | Black 0.328125000  Black-Red Tortie 0.328125000  Blue 0.109375000  Blue-Cream Tortie 0.109375000  Chocolate 0.046875000  Chocolate-Red Tortie 0.046875000  Lilac 0.015625000  Lilac-Cream Tortie 0.015625000 |

CPU Time limit1 second

Memory limit1024 MB

Downloads

[Sample data files](https://sigmatechhungary.kattis.com/test/programming/zasror2fs6myvjydtrs3kxtfkthud46i/problems/087b9eeb28e2c906/file/statement/samples.zip)

Authors

Per Austrin and Pehr Söderman

Problem C

In junior high, you were fascinated by secret codes. When you discovered the Playfair cipher, you shared the knowledge with a friend, and the two of you started using the cipher in order to pass notes in class without fearing that they would be read if intercepted. Because you enjoyed this so much at the time, you have continued to encode all of your correspondence in cipher. However, since you now attend different universities, your messages have become longer and more frequent. You would like to save time by writing a program to encode your messages.

The Playfair Cipher

The Playfair cipher uses a key word or phrase to generate a 5 by 5 table. To do this, fill in the spaces in the table with the letters of the keyword in row major order, ignoring subsequent occurrences of the same letter and any spaces and replacing any instances of the letter “J” with “I”. Fill any remaining spaces with the rest of the letters of the alphabet in order, excluding the letter “J”.

Example: The keyword “Playfair” would generate the key table:

PLAYF  
IRBCD  
EGHKM  
NOQST  
UVWXZ

To encrypt a message, break it into digraphs (groups of 2 letters) such that, for example, “Hello World” becomes “HE LL OW OR LD”. Make sure to apply the following rules:

* Ignore spaces, and convert all letters to upper case.
* Convert all instances of the letter “J” in your message to “I”.
* If both letters in the digraph are the same, add an "X" after the first letter, rearranging the subsequent digraphs if necessary (e.g. If the message is “Whoop”, it breaks into the digraphs, “WH OX OP”).
* If a letter is left without a partner, add an “X” at the end.

Then, encode each digraph in the message by applying the following rules:

* If the letters are not on the same row or column, look along the row of each letter in the digraph until you reach the column that has the other letter in the digraph. The letter at this intersection replaces the original letter.
* If the letters appear on the same row of the table, replace each with the letter to its immediate right. If a letter is at the end of its row, replace it with the letter at the beginning.
* If the letters appear on the same column of the table, replace each with the letter directly beneath it. If a letter is at the bottom of its row, replace it with the letter at the top.
* If the digraph is “XX”, always replace it with “YY”.

Example:

(1) (2) (3)

Z \* \* O \* \* \* O \* \* \* \* \* \* \*

\* \* \* \* \* \* \* B \* \* \* O Y R Z

\* \* \* \* \* \* \* \* \* \* \* \* \* \* \*

R \* \* X \* \* \* R \* \* \* \* \* \* \*

\* \* \* \* \* \* \* Y \* \* \* \* \* \* \*

Hence, OR -> ZX Hence, OR -> BY Hence, OR -> YZ

**Input**

Input will consist of up to 50 test cases. Each will begin with the number of lines to be encoded, 1≤n≤50. This will be followed by a single line that contains the key phrase for the cipher. After this comes n lines of text to encrypt. All input besides n will consist only of alphabetic characters (a–z, A–Z) or spaces with length in the range [1,50]. Input ends when n=0.

**Output**

For each test case, output the encoded lines of text. All letters should be in upper case. Output a blank line between adjacent cases

|  |  |
| --- | --- |
| **Sample Input 1** | **Sample Output 1** |
| 4  ThisIsATest  AQuickBrownFox  JumpedOverTheLazyDogs  Playfair ciphers are fun  Hooray for cryptology  2  Just another example  Meet me at eight o clock  Signed Agent Double O Eight  0 | IUQABLDPPVUNQV  AQKRBFVTDOHICGFASMVOIY  QKSZNFSQLCWBDOATODNZLZ  TPPUSZEUQDYSOHQGVOZY  LOLELOIANTFEUEGXNDCB  TUCEOGTKROUGMOXBRHNTFEIL |

CPU Time limit1 second

Memory limit1024 MB

Downloads

[Sample data files](https://sigmatechhungary.kattis.com/test/programming/zasror2fs6myvjydtrs3kxtfkthud46i/problems/c28d77a6dd8c40ed/file/statement/samples.zip)

Author

Jennifer Shimada