BEE 1100

FAVORITE | DESCRIPTION | RANKING | FORUM | UDEBUG

GRAPH | LEVEL 5 | + 5.4 POINTS | BASE TIME LIMIT: 1 SECOND | MEMORY LIMIT: 200 MB

beecrowd | 1100

Knight Moves

Unknown Author

Timelimit: 1

Peter is doing a research on the *Traveling Knight Problem (TKP)* where you have to find the shortest closed tour of knight moves that visits each square of a given set of *n* squares on a chessboard exactly once. He thinks that the most difficult part of the problem is determining the smallest number of knight moves between two given squares and that, once you have accomplished this, finding the tour would be easy.

Of course you know that it is vice versa. So you must offer him a program that solves the "difficult" part.

Your job is to write a program that takes two squares *a* and *b* as input and then determines the number of knight moves on a shortest route from *a* to *b*.

PROBLEM: 1100

LANGUAGE: C99

SOURCE CODE

```c
#include <stdio.h>

int main() {

    /**
     * Escreva a sua solução aqui
     * Code your solution here
     * Escriba su solución aquí
     */

    return 0;
}
```

CODE YOUR SOLUTION AND SUBMIT!

SUBMIT

Input

The input file will contain one or more test cases. Each test case consists of one line containing two squares separated by one space. A square is a string consisting of a letter (a-h) representing the column and a digit (1-8) representing the row on the chessboard. See the figure above.

Output

For each test case, print one line saying"`To get from` *xx* `to` *yy* `takes` *n* `knight moves.`".

Input Sample	Output Sample
e2 e4	To get from e2 to e4 takes 2 knight moves.
a1 b2	To get from a1 to b2 takes 4 knight moves.
b2 c3	To get from b2 to c3 takes 2 knight moves.
a1 h8	To get from a1 to h8 takes 6 knight moves.
a1 h7	To get from a1 to h7 takes 5 knight moves.
h8 a1	To get from h8 to a1 takes 6 knight moves.
b1 c3	To get from b1 to c3 takes 1 knight moves.
f6 f6	To get from f6 to f6 takes 0 knight moves.

Input and Output *by* Neilor