

## Babel

John and Mary, a pair of brothers who are studying languages, have come up with an interesting fact about languages: some have words in common. For instance, “Red” is the name of a color in English but is the translation of “Network” in Spanish, or “Amigo”, which means friend in both Spanish and Portuguese.

John proposed a game to Mary, in which she had to, given a start language, connect languages with words they have in common, until arriving to a given end language. For instance, if the start language is Portuguese and the end language is French, then Mary could use the sequence “amigo – actual – date”, as it connects the given languages (Portuguese/Spanish, Spanish/English, English/French). However, as John found that Mary could solve the problem easily, some rules were added subsequently:

- The solution must have the smallest number of letters in total
- Two consecutive words must not start with the same initial letter.

As the rules changed, the previous solution would not work, as “amigo” and “actual” both start with a. Another solution would be “amigo red date”, with a total length of 12.

John then compiles an enormous list of words in the internet, and challenges Mary to solve the problem. As there may be more than one solution, he asked her to answer if there is a solution; and, in such case, she must answer the number of letters in the best solution.

<b>Client:</b>	Mary
<b>User:</b>	Mary
<b>Environment:</b>	<ul style="list-style-type: none"><li>- A program must be written in order to solve the challenge given to Mary.</li><li>- Program’s inputs consist of several test cases, with each test case split into 2 parts. First part being an integer, and second part a group of strings.</li><li>- The integer from the first part of input is M, the number of words compiled by John for a test case.</li><li>- Second part of input are three strings <math>L_1</math>, <math>L_2</math> and P, separated by spaces, representing respectively two languages and one word that them both have in common.</li><li>- All the strings in input are at least 1 at most 50 in length.</li><li>- All the strings in input are lowercase letters.</li><li>- The same pair of languages may have several words associated to it, but a word P will never be repeated in a test case.</li><li>- End of input is indicated by a line containing only a 0.</li><li>- Output is, for each test case from input, a line with a single integer representing the length of the shortest sequence satisfying John’s restrictions.</li><li>- In case that the compiled words can not solve the problem, output must be a lowercase “imposivel” (Portuguese for impossible, without quotations).</li></ul>

## Requirement Specification

The following is the requirement specification for the problem specified previously:

<b>Name:</b>	FR1: Read input
<b>Summary:</b>	Input consists of several test cases represented by a list of strings. Each test case starts with an integer P, followed by a pair of string O and D, separated with a space, containing the starting and ending language. Then comes P lines consisting of 3 strings separated by a space. Being the first 2 a pair of languages that have the third string as a word in common. End of input is indicated when P is equal to 0. The program must read the specified input and interpret it in a data structure that can solve Mary's challenge.
<b>Input:</b>	A set of strings representing test cases, with each test case being an integer P, a line and a group of P lines. Input ends when P is equal to 0.
<b>Output:</b>	The program reads the test case and interprets it to a data structure capable of solving the problem.

<b>Name:</b>	NFR2: Determine the best solution.
<b>Summary:</b>	With the languages connected by different words, the program must find a way to arrive from a start language to an ending language. And, if such way exists, the program must determine the better way to achieve such goal.
<b>Input:</b>	A network of different languages connected by several different words, a starting and ending language, and said different words.
<b>Output:</b>	The length of the lexicographically shortest path from the starting to the ending language in case it exists. "Imposivel" otherwise.

<b>Name:</b>	FR2: Print output.
<b>Summary:</b>	After the program solves the problem for each test case, it must show the solution to all of them.
<b>Input:</b>	Solution to the problem in all the test cases.
<b>Output:</b>	Several lines, one for each test case, containing the number of the shortest path from a language to another, or "Imposivel" in case such path doesn't exist.