**Rally Health / Software Engineer, Entery Level**

At Rally, we pride ourselves in putting health in the hands of the individual. That energy permeates everything we do.

Your problem today is to make that happen -- literally.

You’ve probably heard of the Levenshtein distance between strings (if you haven’t, no problem! Check it out on Wikipedia). Your task is to transform one word into another, with four operations: add a letter, delete a letter, change a letter, and take an anagram of the existing word. Additionally, you have to obey the following rules:

* Every interim step between the first and the last word must also be a word
* No interim step can be less than three letters
* The first line of input will contain the “cost” of each operation in the order above
* The second line of input will contain the starting word
* The third line of input will contain the ending word

Your goal is to find the lowest possible “cost” of transforming the starting word into the ending word. You can use any word list you like -- feel free to include your word list or a link to it as part of your solution. (Depending on your word list, your answer might not be exactly the same as ours below.)

Your solution should detect and handle invalid input, and return -1 if there is no solution.

Example input:

1 3 1 5

HEALTH

HANDS

(output: 7) (HEALTH - HEATH - HEATS - HENTS - HENDS - HANDS)

(If your dictionary doesn’t have a couple of these words in there, don’t worry -- you’re scored on your code, not your word list.)

1 9 1 3

TEAM

MATE

(output: 3) (TEAM - MATE)

7 1 5 2

OPHTHALMOLOGY

GLASSES

(output: -1)