Word Ladder

? Ask Doubt

Time-Limit: 2 sec Score: 0/100 Difficulty :

Memory: 256 MB Accepted Submissions: 100 Relevant For: AZ-201 AZ-202

Description

Given two words **beginWord** and **endWord** and a dictionary of words. Find a minimum number of moves to transform **beginWord** into **endWord**. In one move, you can change one letter of the current word. New formed word should always be present in the given dictionary. Note that **beginWord** does not need to be in the dictionary.

Return minimum moves to transform from **beginWord** to **endWord**. If no such transformation exists, return 0.

Input Format

The first line of input contains **N** - the number of words in the dictionary.
Next **N** lines contain dictionary words separated by space.
The last line of input contains **beginWord** and **endWord**.

Constraints

1 ≤ |beginWord| ≤ 10
|endWord| == |beginWord|
1 ≤ Total words in dictionary ≤ 10000
Each word in a dictionary is the same in length as beginWord.
beginWord, endWord, and dictionary words consist of lowercase English letters.
beginWord != endWord
All the words in the dictionary are unique.

Sample Input 1

Copy

```
6
hot dot dog lot log cog
hit cog
```

Sample Output 1

Copy

```
5
```

Note

Explanation: hit → hot → dot → dog → cog

C++14[GCC]

Submit

```
1  #include <bits/stdc++.h>
2  using namespace std;
3
4  class WordLadder
```