**Code Ques 1:**

Amazon Shopping provides a product-search feature that makes browsing products easier. Instead of showing exact matches only, it also displays transformable results for better browsing. A word a is said to be transformable to a word b if a is a subsequence of b. Given searchWord and resultWord, find the minimum number of characters that must be appended at the end of searchWorrl, such that resultWord is a subsequence of the modified searchWord.

Note: A subsequence of a string is a string that results from deleting 0 or more characters from the string without changing the order of the remaining characters. For example, amazon is a subsequence of abcmmdaqzxopn while abc is not a subsequence of cdhbqaab.

Example

searchWord = 'amaze"

resultWord = 'amazon"

armaze -> armazeo -> armazeon

amaz amazo amazon

Add 2 characters, 'on', to searchWord to make resultWord, a subsequence of searchWord.

Function Description

Complete the function findMinimumCharacters in the editor below.

findMinimumCharacters has the following parameters:

string searchWord: the search word, to which characters are appended

string resultWord: the result word, which should be a subsequence of searchWord

Returns

int: the minimum number of characters to be appended to searchWord, to make resultWord transformable to searchWord.

Constraints:

• 1 <= IsearchWord|, IresultWordI <= 105 where, |x| represents the length of the string x.

• searchWord and resultWord consist of lowercase English letters only.

Sample Case 0:

abcz -> searchword = “abcz”

azdb -> searchword = “azdb”

Sample Output:

2

Explanation:

Append ‘d’ and ‘b’ -> searchWord = “abczdb”

‘azdb’ is a subsequence of searchWord = “abczdb”.

**Code Ques 2:**

As an intern at Amazon, you have been assigned a task to implement the sign-in pages in the Amazon Dummy Website.

There are three sign-in pages, each with its own API:

|  |  |  |  |
| --- | --- | --- | --- |
|  | Register | Login | Logout |
| Function | Registers a new user with the username and password | Verifies the username and password, then grants or denies access | usemame logs out of the website |
| API Request | register <username> <password> | login <username> <password> | logout <username> |
| Returns | • If the registration was successful, registered successfully  • If the user already exists, username already exists. | • If the login was successful, login successfully  • If the login was unsuccessful, login unsuccessful. | • If the logout was successful, logout successfully.  • If the given username wasn't logged in, logout unsuccessful. |

Given a log of API requests, return the list of returns from the mock website.

Notes:

• Initially, there are no users registered. • If a user is already logged in and makes a login request, the new request is unsuccessful. The original login remains active. • Each log is an API request and is in one of the three allowed formats. • The order of execution of each request is the same as the order of input. • The usernames and passwords are case-sensitive.

Example

The website receives the following API requests in order.

Function Description

Complete the function implementAPI in the editor below.

ImplementAPI has the following parameter:

string logs[n]: each of the API requests

Returns

string[n]: an array of strings where ith string is the return value of the ith API request

Constraints

• 1 <= n <=10^5 • 1 <= | usemame |, | password | <= 10 (where |s| is the length of string s

• username and password are alphanumeric strings, ranges [0-9, a-z, A-Z]

• Sample Case 0

STDIN FUNCTION

----------- ---------------

5 -> logs size, n = 5

Register david david123 -> logs[] description

Register adam 1Adam1

Login david david123

Login adam 1adam1

Logout david

**Sample Output:**

Registered successfully

Registered successfully

Logged in successfully

Login unsuccessful

Logged out successfully

Explanation:

• There is no user with the username david" registered, so the registration is successful.

• There is no user with the username "adam" registered, so the registration is successful.

• The username and the password for the user match with that in the database, so the login is successful.

• The password C”1adam1”) does not match the actual password ("1Adam1"), so the login is unsuccessful.

• The user "david" is logged In currently, so the logout is successful.

#include<bits/stdc++.h>

/\*

\*Complete the ‘implementAPI’ function below.

\*

\*the function is expected to return a STRING\_ARRAY.

\*The function accepts STRING\_ARRAY logs as parameter.

\*/

vector<string> implementAPI(vector<string> logs){

}

int main() …