**Prefix match with other strings :-**

Medium Accuracy: 53.02% Submissions: 23K+ Points: 4

Given an array of strings **arr[]** of size **n,**a string **str** and an integer **k**. The task is to find the count of strings in **arr[]** whose **prefix of length** **k** matches with the **k-length prefix** of **str**.

**Example 1:**

**Input**:

n = 6

arr[] = {“abba”, “abbb”, “abbc”, “abbd”,

“abaa”, “abca”}

str = “abbg”

k = 3

**Output:**   
4

**Explanation**:

“abba”, “abbb”, “abbc” and “abbd” have their prefix of length 3 equal to 3-length prefix of **str** i.e., **"abb".**

**Example 2:**

**Input:**

n = 3

arr[] = {“geeks”, “geeksforgeeks”, “forgeeks”}

str = “geeks”

k = 2

**Output:**2  
**Explanation**:  
“geeks” and “geeksforgeeks” have their prefix of length 2 equal to 2-length prefix of **str** i.e., **"ge".**

**Your Task:**  
You don't need to read input or print anything. Your task is to complete the function **klengthpref()** which takes the array of strings arr[], its size **n**and an integer **k,**a string**str**as input parameters and returns the count of strings in **arr[]** whose prefix of length **k** matches with the **k** length prefix of **str**.

**Expected Time Complexity:** O(n\*l) where l is the length of the longest word in arr[].  
**Expected Auxiliary Space:** O(n\*l) where l is the length of the longest word in arr[].

**Constraints:**  
1 <= n <= 1000  
1<= |arr[i]| , |str| <= 1000  
1 <= k <= 1000  
arr[i], str must contain only lowercase English alphabets

**Code :-**

//{ Driver Code Starts

//Initial template for C++

#include<bits/stdc++.h>

using namespace std;

// } Driver Code Ends

//User function Template for C++

class Solution{

public:

bool equal(string &s1, string &s2, int k){

for(auto i=0; i<k; i++){

if(i>=s1.size() || i>=s2.size() || s1[i]!=s2[i])

return false;

}

return true;

}

int klengthpref(string arr[], int n, int k, string str){

int ans=0;

for(auto i=0; i<n; i++){

if(equal(arr[i], str, k))

ans++;

}

return ans;

}

};

//{ Driver Code Starts.

int main()

{

int t;

cin>>t;

while(t--)

{

int n;

cin>>n;

string arr[n];

for(int i=0;i<n;i++)

{

string s;

cin>>arr[i];

}

int k;

cin>>k;

string str;

cin>>str;

Solution ob;

cout << ob.klengthpref(arr, n, k, str) << endl;

}

return 0;

}

// } Driver Code Ends

**T.C :- O(n\*l), l = max length of string in array**

**S.C :- O(1), ignoring function call stack**