**Match specific pattern**

Submissions: [2606](https://practice.geeksforgeeks.org/problem_submissions.php?pid=700321)  Accuracy:

45.25%

   Difficulty: [Easy](https://practice.geeksforgeeks.org/Easy/1/0/)   Marks: 2

Show Topic Tags   

[Amazon](https://practice.geeksforgeeks.org/company/Amazon/)[Microsoft](https://practice.geeksforgeeks.org/company/Microsoft/)

Given a dictionary of words and a pattern.  Every character in the pattern is uniquely mapped to a character in the dictionary.

The task is to complete a function**findMatchedWords** that returns a vector of  strings matching with given pattern. The function takes two argument the first argument is an array of strings which denote the dictionary and the second argument is the pattern to match.  
  
**Input:**  
The first line of input contains an integer T denoting the no of test cases . Then T test cases follow . The first line of each test case contains an integer N denoting the no of strings in the dictionary and then in the next line are N space separated strings denoting the strings of the dictionary. Then in the next line is a string pattern.  
  
**Output:**  
The output for each test case will be the space separated strings that matches the given pattern in lexicographical order.  
  
**Constraints:**  
1<=T<=100  
1<=N<=100  
1  
**Example(To be used only for expected output):  
Input**  
1  
4  
abb abc  xyz xyy  
foo  
  
**Output**  
abb xyy  
  
**Explanation**  
In the above test case xyy and abb have same character at index 1 and 2 like the pattern.

**Note:**The **Input/Ouput** format and **Example** given are used for system's internal purpose, and should be used by a user for **Expected Output** only. As it is a function problem, hence a user should not read any input from stdin/console. The task is to complete the function specified, and not to write the full code.

\*\* For More Input/Output Examples Use ['Expected Output'](https://practice.geeksforgeeks.org/problems/match-specific-pattern/1#ExpectOP) option \*\*

<https://practice.geeksforgeeks.org/problems/match-specific-pattern/1>

#include <iostream>

#include <map>

#include <stdio.h>

#include <set>

#include <vector>

using namespace std;

bool isIsomorphic(string str1, string str2) {

int m = str1.length(), n = str2.length();

// Length of both strings must be same for one to one

// corresponance

if (m != n)

return false;

// To mark visited characters in str2

bool marked[256] = {false};

// To store mapping of every character from str1 to

// that of str2. Initialize all entries of map as -1.

int map[256];

//memset(map, -1, 256);

for(int i =0; i<256; i++) {

map[i] = -1;

}

// Process all characters one by on

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

{

// If current character of str1 is seen first

// time in it.

if (map[str1[i]] == -1)

{

// If current character of str2 is already

// seen, one to one mapping not possible

if (marked[str2[i]] == true)

return false;

// Mark current character of str2 as visited

marked[str2[i]] = true;

// Store mapping of current characters

map[str1[i]] = str2[i];

}

// If this is not first appearance of current

// character in str1, then check if previous

// appearance mapped to same character of str2

else if (map[str1[i]] != str2[i])

return false;

}

return true;

}

vector<string> findMatchedWords(vector<string> dict,

string pattern)

{

std::vector<string> v;

//Your code here

for(int i =0; i<dict.size(); i++) {

if(isIsomorphic(dict[i], pattern) ) {

v.push\_back(dict[i]);

}

}

return v;

}

int main() {

return 0;

}