**Finding-Pairs**

Submissions: [221](https://practice.geeksforgeeks.org/problem_submissions.php?pid=2827)  Accuracy:

41.61%

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

Show Topic Tags   

For N number of alphabet pairs(in upper or lower case), count the occurrence of the given alphabet pairs in a string S of size M. Output the number of pairs found in the string S, out of total N pairs.  
Note: Finding the pair means finding both the characters in a particular pair in the string.

**Input:**  
First line of input is an integer T, denoting the number of test cases. For each test case, enter two integers N and M on the same line, denoting the number of pairs of alphabets to be searched for and the size of the string S, respectively. Next line of input comprises of the N alphabet pairs separated by a single space in between. The last line of input comprises of the string S of size M.

**Output:**  
The only line of output for each test case is an integer that tells the number of pairs (<=N) that occured in the string S. If no pair is found, the output will be zero.

**Constraints:**  
1<=T<=100  
1<=N<=10  
2<=M<=20

**Example:  
Input:**  
3  
3 5  
A G d i P o  
APiod  
1 3  
r e  
ghe  
3 6  
F E n O F s  
FOrnEs  
**Output:**  
2  
0  
3

\*\* For More Input/Output Examples Use ['Expected Output'](https://practice.geeksforgeeks.org/problems/finding-pairs/0#ExpectOP) option \*\*

[Author: Anushree\_20](https://auth.geeksforgeeks.org/user/Anushree_20/practice/)

<https://practice.geeksforgeeks.org/problems/finding-pairs/0>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp1

{

class Program

{

static int CantPares(char[] pares, string s)

{

Dictionary<char, int> dic =

new Dictionary<char, int>();

for (int i = 0; i < s.Length; i++)

{

if (dic.ContainsKey(s[i])) dic[s[i]]++;

else dic[s[i]] = 1;

}

int cont = 0;

for (int i = 0; i + 1 < pares.Length; i += 2)

{

if (dic.ContainsKey(pares[i]) && dic.ContainsKey(pares[i + 1]))

{

cont++;

}

}

return cont;

}

static void Main(string[] args)

{

int t = int.Parse(Console.ReadLine().Trim());

while(t-- > 0)

{

string[] input = Console.ReadLine().Trim().Split(' ');

int n = int.Parse(input[0]);

int m = int.Parse(input[1]);

char[] a = Array.ConvertAll( Console.ReadLine().Trim().Split(' '), e => char.Parse(e));

string b = Console.ReadLine().Trim();

Console.WriteLine(CantPares(a, b));

}

Console.ReadLine();

}

}

}