**The Monk and Kundan**

Attempted by: **487**

/

Accuracy: **74%**

/

Maximum Score: **20**

/

0 Votes

Tag(s):

Ad-Hoc, Easy, Easy, Hashing

**PROBLEM**

**EDITORIAL**

**MY SUBMISSIONS**

**ANALYTICS**

Kundan being a good friend of Monk, lets the Monk know that he has a following string **Initial** which consists of the following letters in the mentioned order: "**abcdefghijklmnopqrstuvwxyz1234567890ABCDEFGHIJKLMNOPQRSTUVWXYZ**".

He also has various lists of strings, and now he wants the Monk to compute the *Hash* value of each list of strings.   
Here's the following algorithm used by the Monk to do it.   
So, the *Hash* is the summation of all the character values in the input:   
(**currentIndex + (position of the character In the string initial) ).** And then this *hash* is multiplied by the **Number of strings in the list**.

Let's assume that the list of strings is: **["aA1", "b"]**. So, our answer would be:  
a: 00 + 00 = 00.  
A: 11 + 3636 = 3737.   
1: 22 + 2626 = 2828.  
b: 00 + 11 = 11.

So, 22 \* (00 + 11 + 3737 + 2828) = 22 \* (6666) = 132132.

**Input format:**  
The first line contains an integer **T**, denoting the number of test cases. For every test case, on a single line, there will be **N** number of strings all of them separated by a space, denoting all the strings of that particular list of strings.

**Output format:**  
Print the required *hash* for each of the mentioned list of strings.

**Constraints:**  
11 ≤ Test Cases ≤ 5050  
11 ≤ Length of a string ≤ 3030  
11 ≤ Number of strings in a list ≤ 2020

**Note:**  
All the characters in the input will be valid, that is, will be part of the string **Initial.**

**SAMPLE INPUT**

3

aA1 b

a b c d

aa BB cc DD

**SAMPLE OUTPUT**

132

24

640

**Explanation**

Refer to the statement again for the sample explanation.

**Time Limit:**1.0 sec(s) for each input file.

**Memory Limit:**256 MB

**Source Limit:**1024 KB

**Marking Scheme:**Marks are awarded when all the testcases pass.

**Allowed Languages:**C, C++, Clojure, C#, D, Erlang, F#, Go, Groovy, Haskell, Java, Java 8, JavaScript(Rhino), JavaScript(Node.js), Lisp, Lisp (SBCL), Lua, Objective-C, OCaml, Octave, Pascal, Perl, PHP, Python, Python 3, R(RScript), Racket, Ruby, Rust, Scala, Scala 2.11.8, Swift, Visual Basic

<https://www.hackerearth.com/practice/data-structures/hash-tables/basics-of-hash-tables/practice-problems/algorithm/the-monk-and-kundan/>

using System;

using System.Numerics;

class MyClass

{

static void Main(string[] args)

{

string initial = "abcdefghijklmnopqrstuvwxyz1234567890ABCDEFGHIJKLMNOPQRSTUVWXYZ";

//string[] list = { "aA1", "b" };

//string[] list = { "a" ,"b", "c", "d"};

//string[] list = { "aa", "BB", "cc", "DD"};

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

while (t-- > 0)

{

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

int ans = 0;

foreach (string elem in list)

{

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

{

ans += i + initial.IndexOf(elem[i]);

}

}

ans \*= list.Length;

Console.WriteLine(ans);

}

}

}