**Difficulty of sentence**

Show Topic Tags   

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

Given a sentence as a string S. Write a program to calculate difficulty of a given sentence. A word in the given string is considered hard if it has 4 consecutive consonants or number of consonants are more than number of vowels. Else the word is easy. Difficulty of sentence is defined as 5\*(number of hard words) + 3\*(number of easy words).  
**Note:** uppercase and lowercase characters are same.  
  
**Input:**  
First line of input contains a single integer T which denotes the number of test cases. Then T test cases follows. First line of each test case contains a string S which denotes the sentence.  
  
**Output:**  
For each test case print the difficulty of the sentence as described above.  
  
**Constraints:**  
1<=T<=100  
1<= length( S ) <= 105  
  
**Example:  
Input:**  
1  
Difficulty of sentence  
**Output:**  
13

\*\*For More Examples Use Expected Output\*\*

Contributor: Harsh Agarwal

<http://practice.geeksforgeeks.org/problems/difficulty-of-sentence/0>

/\*

\* To change this template, choose Tools | Templates

\* and open the template in the editor.

\*/

package javaapplication250;

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

import java.math.BigInteger;

import java.util.Arrays;

import java.util.HashMap;

import java.util.HashSet;

import java.util.LinkedHashSet;

/\*\*

\*

\* @author Administrador

\*/

public class JavaApplication250 {

/\*\*

\* @param args the command line arguments

\*/

public static void main(String[] args) throws IOException {

// TODO code application logic here

BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

int t = Integer.parseInt(br.readLine());

while(t-- > 0) {

String[] s = br.readLine().trim().split(" ");

String vowels = "aeiouAEIOU";

String cons = "bcdfghjklmnpqrstvwxyzBCDFGHJKLMNPQRSTVWXYZ";

int hard =0, easy =0;

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

String pal = s[i];

int c =0;

int consec =0, max\_consec=0;

for(int j =0; j<pal.length(); j++) {

if(cons.contains(String.valueOf(pal.charAt(j)))){

c++;

consec++;

max\_consec = Math.max(consec, max\_consec);

}else{

consec=0;

}

}

if(max\_consec >= 4 || c > (pal.length()-c) ) {

hard++;

}else{

easy++;

}

}

System.out.println(hard\*5 + easy\*3);

}

}

}