**Distinct Substrings**

[ad-hoc](http://www.practice.geeksforgeeks.org/tag-page.php?tag=ad-hoc&isCmp=0)[string](http://www.practice.geeksforgeeks.org/tag-page.php?tag=string&isCmp=0)

Given a string S consisting of uppercase alphabatic characters. Return the  number of different substrings of size 2 that appear in S as contiguous substrings.

**Input:**  
The first line contains 'T' denoting the number of testcases. Then follows description of testcases. The next T lines contains the string S.  
  
**Output:**  
Output the number of different substrings of size 2 in S.

**Constraints:**  
1<=T<=50  
1<=|S|<=100

**Example:**  
Input :  
2  
ABCAB  
XYZ

Output :  
3  
2

Explanation:  For "ABCAB", the three distinct substrings of size 2 are "AB", "BC" and "CA".  For "XYZ", the two distinct substrings of size 2 are "XY" and "YZ".

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

<http://www.practice.geeksforgeeks.org/problem-page.php?pid=236>

#include <iostream>

#include <stdio.h>

#include <set>

#include <vector>

#include <map>

#include <math.h>

#include <string.h>

//#include <conio.h>

using namespace std;

int main() {

int t;

scanf("%d", &t);

while(t--) {

std::string S;

cin >> S;

std::set<std::string> set;

for(int i =0; i < S.length() -1; i++) {

std::string actual = S.substr(i, 2);

set.insert(actual);

}

cout << set.size() << endl;

}

//getch();

return 0;

}