**Good String**

[strings](http://www.practice.geeksforgeeks.org/tag-page.php?tag=strings%20&isCmp=0)

Given a string **S** of length **N**, you have to tell whether it is good or not. A *good* string is one where the distance between any two adjacent character is exactly 1. Consider that the alphabets are arranged in cyclic manner from '**a**' to **'z'**. Hence, distance between any character **'i'** and **'j'** will be defined as minimum number of steps it takes **'i'** to reach **'j'**. Here, character **'i'** can start moving clockwise or anti-clockwise in order to reach at position where character**'j'** is placed. Your task is simple, where you just need to print **"YES"** or **"NO"**(without quotes) depending on whether the given string is Good or not

**Input:**  
First line of the input contains**T** denoting the number of test cases.Then **T** lines follow. Each line contains a string **S**.

**Output:**  
Print  the answer for each testcase in a separate line.

**Constraints:**

1≤**T**≤50  
1≤**|S|**≤50  
  
**Note: S** contains only lowercase alphabetic characters

**Input:**  
3  
aaa  
cbc  
bcd

**Output:**  
NO  
YES  
YES

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

<http://www.practice.geeksforgeeks.org/problem-page.php?pid=1366>

#include <iostream>

#include <stdio.h>

using namespace std;

int main() {

int t;

scanf("%d", &t);

while(t--) {

string s;

cin >> s;

std::string ans = "YES";

for(int i = 1; i < s.size(); i++) {

if( (s[i-1] == 'a' && s[i] == 'z') || (s[i] == 'a' && s[i-1] == 'z')){

}

else if(std::abs( s[i-1] - s[i] ) != 1) {

ans = "NO";

break;

}

}

cout << ans << endl;

}

return 0;

}