COMPETITIVE CODING
with
RAMAN CLASSES

... An interactive series for those willing to learn and code

Problem: Game of thrones

Topic	: **Arrays**
Difficulty	: **EASY**
Programming Language	: **C++**
Time to Spend	: **15 min**

Problem Statement

Problem:

Dothraki are planning an attack to usurp King Robert's throne. King Robert learns of this conspiracy from Raven and plans to lock the single door through which the enemy can enter his kingdom.

But, to lock the door he needs a key that is an anagram of a palindrome. He starts to go through his box of strings, checking to see if they can be rearranged into a palindrome.

For example, given the strings=[aabbccdd] , one way it can be arranged into a palindrome is abcddcba.

Problem Statement

Input :

aaabbbb

Output :

YES

Input Description :

A single line which contains s, the input string.

Output Description :

A single line which contains YES or NO.

Let Us Revise

In order to solve this problem, go through the following concepts.

1. Palindrome
2. Count the occurrence of character in the string.
3. Using Loop statements

Problem Description

We are given a string ‘s’ and we need to find out if the string can be rearranged to form a palindrome string or not?

If it is possible then print “YES”, without quote, otherwise “NO”, without quote.

Note: Palindrome is a sequence, which remains same even if we reverse it. For example: “SANNAS” is a palindrome string.

Problem Description

For example: Assume a string s="ARKOARKO"

As you can see if can be rearranged like "ARKOOKRA" to form a palindrome string.

Output: YES

Let Us Think

In order to solve this problem, let us think and analyse how to get started with this problem.

Try out some examples in the paper. After analysing a little, you can see a pattern, i.e. for a even length string to be a palindrome, each character of the string occurs in the multiple of 2. And,
for a odd length string to be a palindrome, each character of the string occurs in the multiple of 2 ,except one.

Let Us Think

So the problem boils down to counting the occurrence of each character in the string and find out if the given string is palindrome or not?

NOTE: An array of size of distinct character can be used to count the occurrence.

Let Us Think

Now you know the logic. Lets proceed with the code.

Let Us Think

Now you know the logic. Lets proceed with the code.

Things we need to do for this problem:

1. Count the number of occurrence of distinct character in the string.

2. Find out whether the string is palindrome or not.
   i) when the string length is even.
   ii) when the string length is odd.

Let Us Code

1. Count the number of occurrence of distinct character in the string.

```
//An array alphabet of size 26
//to store the occurrence of each character
int alphabets[26] = { 0 };
for (int i = 0;i < s.length();i++) {
    alphabets[s[i] - 'a']++;
}
```

Let Us Code

2. Find out whether the string is palindrome or not.
   i)when the string length is even.

```
//if string length is even
//then for palindrome
//no character should have odd occurence
if (s.length() % 2 == 0) {
    int odd = 0;
    for (int i = 0;i < 26;i++) {
        if (alphabets[i] % 2 == 1) {
            odd++;
        }
    }

    if (odd == 0) cout << "YES\n";
    else cout << "NO\n";
}
```

2. Find out whether the string is palindrome or not.
ii)when the string length is odd.

```
//if string length is even
//then for palindrome
//exactly one character should have odd occurrence

else {
    int odd = 0;
    for (int i = 0;i < 26;i++) {
        if (alphabets[i] % 2 == 1) {
            odd++;
        }
    }

    if (odd == 1) cout << "YES\n";
    else cout << "NO\n";
}
```

Thank You !

... brought to you by

www.ramanclasses.in