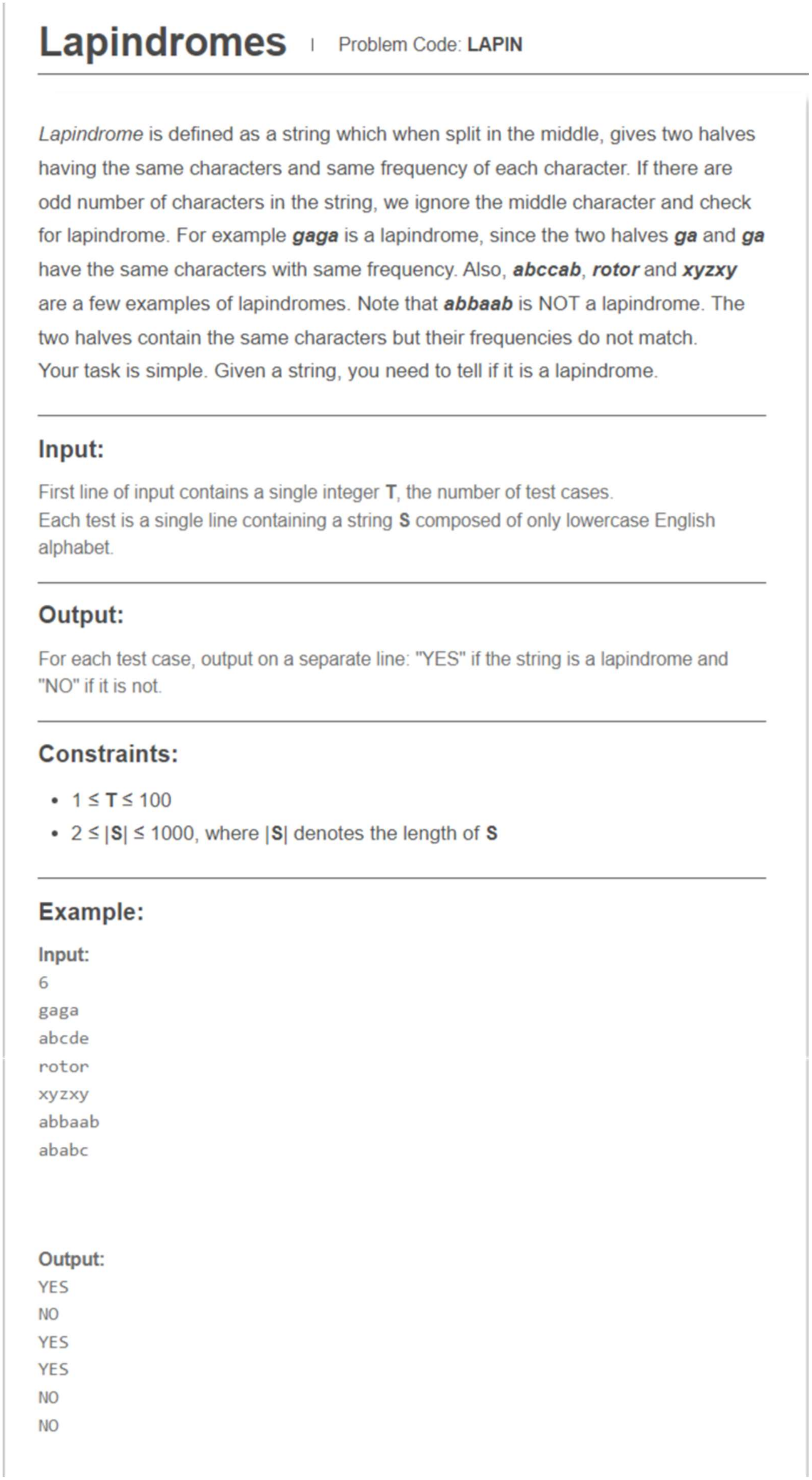


Lapindromes

Problem Code: **LAPIN**

Lapindrome is defined as a string which when split in the middle, gives two halves having the same characters and same frequency of each character. If there are odd number of characters in the string, we ignore the middle character and check for lapindrome. For example ***gaga*** is a lapindrome, since the two halves ***ga*** and ***ga*** have the same characters with same frequency. Also, ***abccab***, ***rotor*** and ***xyzxy*** are a few examples of lapindromes. Note that ***abbaab*** is NOT a lapindrome. The two halves contain the same characters but their frequencies do not match.
Your task is simple. Given a string, you need to tell if it is a lapindrome.

Input:

First line of input contains a single integer **T**, the number of test cases.
Each test is a single line containing a string **S** composed of only lowercase English alphabet.

Output:

For each test case, output on a separate line: "YES" if the string is a lapindrome and "NO" if it is not.

Constraints:

- 1 ≤ **T** ≤ 100
- 2 ≤ |**S**| ≤ 1000, where |**S**| denotes the length of **S**

Example:

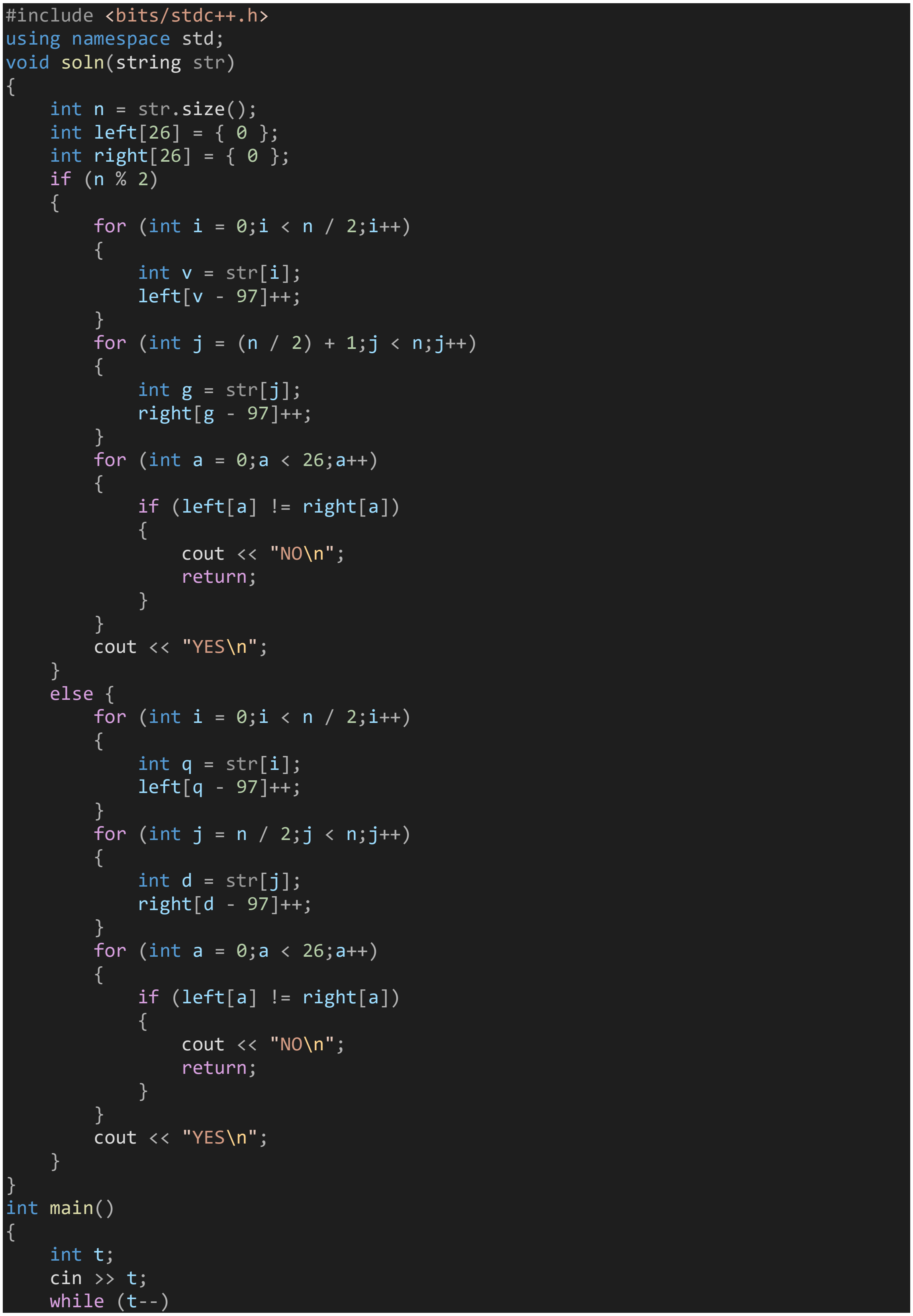
Input:

```
6
gaga
abcde
rotor
xyzxy
abbaab
ababc
```

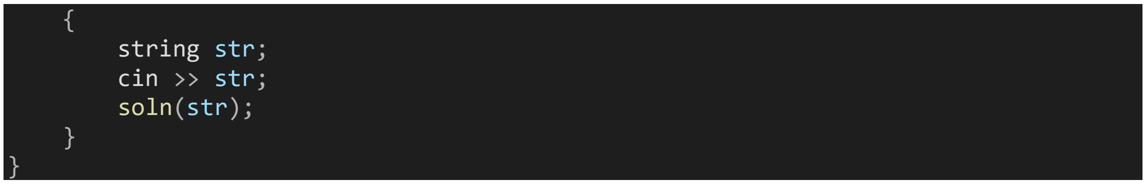
Output:

```
YES
NO
YES
YES
NO
NO
```

CODE



```
#include <bits/stdc++.h>
using namespace std;
void soln(string str)
{
    int n = str.size();
    int left[26] = { 0 };
    int right[26] = { 0 };
    if (n % 2)
    {
        for (int i = 0;i < n / 2;i++)
        {
            int v = str[i];
            left[v - 97]++;
        }
        for (int j = (n / 2) + 1;j < n;j++)
        {
            int g = str[j];
            right[g - 97]++;
        }
        for (int a = 0;a < 26;a++)
        {
            if (left[a] != right[a])
            {
                cout << "NO\n";
                return;
            }
        }
        cout << "YES\n";
    }
    else {
        for (int i = 0;i < n / 2;i++)
        {
            int q = str[i];
            left[q - 97]++;
        }
        for (int j = n / 2;j < n;j++)
        {
            int d = str[j];
            right[d - 97]++;
        }
        for (int a = 0;a < 26;a++)
        {
            if (left[a] != right[a])
            {
                cout << "NO\n";
                return;
            }
        }
        cout << "YES\n";
    }
}
int main()
{
    int t;
    cin >> t;
    while (t--)
```



```
    {
        string str;
        cin >> str;
        soln(str);
    }
}
```

LOGIC

We are first declaring two arrays for left and right sides of the string. The arrays are of 26 size (no. of alphabets). Initialize all the elements of the array by 0.

Now for all the elements of the string, we are storing the count of the alphabets in the array by doing character- 97. (ASCII value of a=>97);