Sherlock considers a string to be *valid* if all characters of the string appear the same number of times. It is also *valid* if he can remove just 1 character at 1 index in the string, and the remaining characters will occur the same number of times. Given a string s, determine if it is *valid*.

For example, if $s = abc$, it is a valid string because frequencies are $\{a:1, b:1, c:1\}$. So is $s = abcc$ because we can remove one c and have 1 of each character in the remaining string. If $s = abccc$ however, the string is not *valid* as we can only remove 1 occurrence of c. That would leave character frequencies of $\{a:1, b:1, c:2\}$.

Input Format

A single string s.

Constraints

- $1 \leq |s| \leq 10^5$
- Each character $s[i] \in ascii[a-z]$

Output Format

Print `YES` if string s is *valid*, otherwise, print `NO`.

Sample Input 0

```
aabbcd
```

Sample Output 0

```
NO
```

Explanation 0

Given $s =$ **"aabbcd"**, we would need to remove two characters, both `c` and `d` → `aabb` or `a` and `b` → `abcd`, to make it valid. We are limited to removing only one character, so s is *invalid*.

Sample Input 1

```
aabbccddeefghi
```

Sample Output 1

```
NO
```

Explanation 1

Frequency counts for the letters are as follows:

`{'a': 2, 'b': 2, 'c': 2, 'd': 2, 'e': 2, 'f': 1, 'g': 1, 'h': 1, 'i': 1}`

There are two ways to make the valid string:

- Remove 4 characters with a frequency of 1: **{fghi}**.

- Remove 5 characters of frequency 2: $\{abcde\}$.

Neither of these is an option.

Sample Input 2

```
abcdefghhgfedecba
```

Sample Output 2

```
YES
```

Explanation 2

All characters occur twice except for e which occurs 3 times. We can delete one instance of e to have a valid string.