

Sherlock and the Valid String



Sherlock considers a string, s , to be *valid* if either of the following conditions are satisfied:

1. All characters in s have the same exact frequency (i.e., occur the same number of times). For example, $s = \text{"aabbcc"}$ is valid, but $s = \text{"baacdd"}$ is not valid.
2. Deleting exactly 1 character from s will result in all its characters having the same frequency. For example, $s = \text{"aabbccc"}$ and $s = \text{"aabbcc"}$ are valid because all their letters will have the same frequency if we remove 1 occurrence of c , but $s = \text{"abcccc"}$ is not valid because we'd need to remove 3 characters.

Given s , can you determine if it's valid or not? If it's valid, print **YES** on a new line; otherwise, print **NO** instead.

Input Format

A single string denoting s .

Constraints

- $1 \leq |s| \leq 10^5$
- String s consists of lowercase letters only (i.e., $[a-z]$).

Output Format

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

Sample Input 0

```
aabbcd
```

Sample Output 0

```
NO
```

Explanation 0

We would need to remove two characters from $s = \text{"aabbcd"}$ to make it valid, because a and b both have a frequency of 2 and c and d both have a frequency of 1. This means s is *invalid* because we'd need to remove more than 1 character to make all its letters have the same frequency, so we print **NO** as our answer.