Atharv's "Just Friends" Map

`Time Limit: 1 second`

`Memory Limit: 256MB`

The *hottest tea* is circulating, and it's about Atharv *bhaiya*. He has a secret girlfriend. The tea gets spicier: she's in the **same college**. The tea is scalding: she's in... *that other* tech society. 🤫

This is a high-stakes, cross-society scandal waiting to happen. To keep their relationship on the down-low, they can't just text normally. They've invented a hasty, desperate "isomorphic cipher" to communicate.

Atharv writes a "safe" text (string s) that looks like he's debugging code (e.g., "egg"), which *should* secretly map to his "real" romantic text (string t, e.g., "add").

The mapping is simple:

- The first character in s maps to the first in t.
- Every subsequent occurrence of that same character in s *must* map to that *same* character from t.

The catch: If a single character from s tries to map to *two different* characters in t (e.g., `s = "foo"`, `t = "bar"`), the cipher is broken, the message is unreadable, and he's at risk of being exposed. Even worse, if two *different* characters from s map to the *same* character in t (e.g., `s = "baba"`, `t = "Kiki"`), it's also a failure.

He's paranoid and needs you to be his "vibe check." Before he hits send, write a program to verify his "safe" text s and his "real" text t are truly isomorphic.

Formal Rules: Two strings s and t are isomorphic if:

1. They have the same length.
2. There is a one-to-one mapping (a bijection) between the characters of s and the characters of t, a rule that is followed for the entire string.
3. This means:
   - Every character in s must map to *one and only one* character in t.
   - No two different characters in s may map to the *same* character in t.

Input Format

- The first line contains a single integer T — the number of test cases.
- Each test case is described by two separate lines:
  - The first line contains the string s.
  - The second line contains the string t.

Constraints

- $1 \leq T \leq 1000$
- $1 \leq \text{length}(s), \text{length}(t) \leq 10^5$
- s and t consist of any printable ASCII characters.
- It is guaranteed that the total sum of $\text{length}(s)$ over all test cases does not exceed 10^6.

Output Format

- For each test case, output **'YES'** if the strings are isomorphic, and **'NO'** otherwise.

Note: The output is **not case sensitive**. You can print `YES`, `yes`, `Yes`, `NO`, `no`, `nO`, etc. All will be accepted.

Sample Input 0

```
3
egg
add
foo
bar
paper
title
```

Sample Output 0

```
YES
NO
YES
```

Explanation 0

Test Case 1:

- `s = "egg"`, `t = "add"`
- 'e' maps to 'a'.
- 'g' maps to 'd'.
- All mappings are unique and consistent. Output: **YES**.

Test Case 2:

- `s = "foo"`, `t = "bar"`
- First 'o' maps to 'a'.
- Second 'o' maps to 'r'.
- Since 'o' cannot map to two different characters ('a' and 'r'), this is invalid. Output: **NO**.

Test Case 3:

- `s = "paper"`, `t = "title"`
- 'p' -> 't', 'a' -> 'i', 'e' -> 'l', 'r' -> 'e'.
- This is a valid one-to-one mapping. Output: **YES**.