U. From S To T

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

You are given three strings s, t and p consisting of lowercase Latin letters. You may perform any number (possibly, zero) operations on these strings.

During each operation you choose any character from p, erase it from p and insert it into string s (you may insert this character anywhere you want: in the beginning of s, in the end or between any two consecutive characters).

For example, if p is `aba`, and s is `de`, then the following outcomes are possible (the character we erase from p and insert into s is highlighted):

- **a**ba → ba, de → **a**de;
- **a**ba → ba, de → d**a**e;
- **a**ba → ba, de → de**a**;
- a**b**a → aa, de → **b**de;
- a**b**a → aa, de → d**b**e;
- a**b**a → aa, de → de**b**;
- ab**a** → ab, de → **a**de;
- ab**a** → ab, de → d**a**e;
- ab**a** → ab, de → de**a**;

Your goal is to perform several (maybe zero) operations so that s becomes equal to t. Please determine whether it is possible.

Note that you have to answer q independent queries.

Input
The first line contains one integer q ($1 \le q \le 100$) — the number of queries. Each query is represented by three consecutive lines.

The first line of each query contains the string s ($1 \le |s| \le 100$) consisting of lowercase Latin letters.

The second line of each query contains the string t ($1 \le |t| \le 100$) consisting of lowercase Latin letters.

The third line of each query contains the string p ($1 \le |p| \le 100$) consisting of lowercase Latin letters.

Output
For each query print `YES` if it is possible to make s equal to t, and `NO` otherwise.

You may print every letter in any case you want (so, for example, the strings `yEs`, `yes`, `Yes` and `YES` will all be recognized as positive answer).

Example

input Copy

```
4
ab
acxb
cax
a
aaaa
aaabbcc
a
aaaa
aabbcc
ab
baaa
aaaaa
```

output Copy

```
YES
YES
NO
NO
```


Assiut University Training - Newcomers

Public

Participant

→ About Group

ICPC Assiut community

Group website

→ Group Contests

- Sheet #10 (General Hard)
- Sheet #9 (General medium)
- Sheet #8 (General easy)
- Sheet #7 (Recursion)
- Sheet #6 (Math - Geometry)
- Sheet #5 (Functions)
- Sheet #4 (Strings)
- Contest #3.1
- Sheet #3 (Arrays)
- Contest #2
- Sheet #2 (Loops)
- Contest #1
- Sheet #1 (Data type - Conditions)

Sheet #10 (General Hard)

Finished

Practice

→ About Time Scaling

This contest uses time limits scaling policy (depending on a programming language). The system automatically adjusts time limits by the following multipliers for some languages. Despite scaling (adjustment), the time limit cannot be more than 30 seconds. Read the details by the link.

Note

In the first test case there is the following sequence of operation:

1. $s = \texttt{ab}, t = \texttt{acxb}, p = \texttt{cax}$;
2. $s = \texttt{acb}, t = \texttt{acxb}, p = \texttt{ax}$;
3. $s = \texttt{acxb}, t = \texttt{acxb}, p = \texttt{a}$.

In the second test case there is the following sequence of operation:

1. $s = \texttt{a}, t = \texttt{aaaa}, p = \texttt{aabbcc}$;
2. $s = \texttt{aa}, t = \texttt{aaaa}, p = \texttt{abbcc}$;
3. $s = \texttt{aaa}, t = \texttt{aaaa}, p = \texttt{bbcc}$;
4. $s = \texttt{aaaa}, t = \texttt{aaaa}, p = \texttt{bbcc}$.