



## Input Format

The first line of the input contains a single integer  $T$  denoting the number of test cases that follow.

For each test case:

- The first line contains the string  $A$ .
- The second line contains the string  $B$ .

## Output Format

For each test case, output a single line containing  $A$  with the  $\#$  replacements.

## Constraints

$$1 \leq T \leq 100$$

$$1 \leq |A| \leq 100$$

$$1 \leq |B| \leq 100$$

It is guaranteed that  $A$  and  $B$  contain only lowercase letters `abcdefghijklmnopqrstuvwxyz` and underscores `_`

It is guaranteed that a solution is always possible with the given inputs.

# Sample Test Cases

## Sample Input

[Download](#)

```
5
ccaalliiccoo
calico
big_bens_bday
bbb
job
jb
include_over_californication
ilovecalico
six_seven_rizzler_sigma_banned_discord_skibidi_toilet
severance
```

## Sample Output

[Download](#)

```
#ca##l#i#c#o
b###b####b###
j#b
i##l####ove##cali#####c###o#
s####eve##r#####a###n#####c#####e#
```

*Note that this is one of many possible correct outputs. If there are multiple solutions, you may output any of them.*

## Sample Explanations

For test case #1, **A** is `ccaalliiccoo` and **B** is `calico`. **A** contains the character of `calico` with each characters repeated twice. We can replace a copy of each character with `#` to get **B**.

For test case #2, we need to replace all the characters of **A** except the three `b`'s.

For test case #4, note that there are three `o`'s that we could choose to keep or replace. However, if we keep the second or third `o`, we will be unable to fully create **B** by joining the remaining characters, as there are no `v`'s later in the string.